Documentation Flow Analysis

to improve project execution

A. Reumer

December 16, 2020– SDPO.20.023.m

TU Delft

Documentation Flow Analysis

to improve project execution

by

A. Reumer

to obtain the degree of Master of Science
at the Delft University of Technology,

Student number:	4286553	
Project duration:	January 20, 2020 – December 17, 2020	
Thesis committee:	Dr. ir. J.F.J. Pruyn,	TU Delft
	Dr. ir. E.B.H.J. van Hassel,	TU Delft & Universiteit Antwerpen
	Dr. F. Schulte,	TU Delft
	ir. M.R.B. van Schaik	Damen Shipyards

TU Delft

Preface

This study is caried out to fulfull the 'MT54015 - MSc Thesis' course requirements of the master program of Marine Technology. This course marks the end of the program.

A. Reumer
Delft, December 16, 2020

Summary

Current challenges at Damen Technical Coorperation (DTC), regarding an imperfect procedure for documentation sharing, misalignment and quality issues in documentation flow, are the motivation for this master thesis. DTC is a company within the Damen Group that offers a unique service for shipyards; a variety of material and service packages to make local building possible. Project management is one of these services. Although their approach is unique, are the faced challenges not uncommon; similar issues are reported in shipbuilding literature. However, due to their method is documentation flow essential for good performances of DTC-projects.

In shipbuilding projects project information is stored in project documents. Documentation is a term that includes both information storage and retrieval systems. Documentation quality concerns quality of the documents and the retrieval systems. The challenges faced at DTC can be described by documentation quality.

This study aims to improve the documentation sharing strategy, and so project execution, of DTC-projects. In line with this aim the central research question is:

In what manner can project execution at DTC be improved, by providing insight into the effect of the quality of shared project documentation on project success?

From the literature it is found that project success requires process integration and process integration demands process understanding. Documentation flow modelling is seen as an attractive tool to obtain process understanding, that can lead in the direction of process integration. Out of four models, the Design Structure Matrix (DSM) is chosen to model documentation flow. Subsequently, a DSM-based discrete-event Monte Carlo simulation is used to analyse the documentation flow of a DTC-project.

Documentation flow is studied by analysing the effects of flow interruptions. Rework is an example of a flow interruption. Based on the literature, multiple causes of document-flow related rework, document quality attributes, are simulated. Documentation quality attributes are ranked on 7 criteria: mean, median, 25th percentile, 75th percentile, standard deviation, likelihood for unacceptable outcome and schedule risk. These criteria rank the impact of the documentation quality attributes in the following order, where highest impact is ranked highest:

1. **Mistakes in the documentation**
   Accuracy of the documentation
2. **Input changes in the documentation**
   Certainty of the documentation
3. **Poor communication of the documentation**
   Clarity, conformance, coordination and legibility of the documentation
4. **Delayed documentation - Wait for documentation**
   Timeliness of the documentation
5. **Delayed documentation - Use assumptions to start tasks early**
   Completeness of the documentation

A detailed analysis is carried out to identify activities that have a large impact on total project outcome, when rework occurs. From the detailed analysis it is concluded that:

1. Rework on the below-listed activities has the most impact on project duration.
   (a) Detailed engineering tasks, specifically:
      i. Ship construction drawings

   ii. Mechanical diagrams

   (b) Block outfitting, specifically:

      i. Equipment
      ii. Hot work

   (c) Zone outfitting, specifically:

      i. Equipment
      ii. Hot work
      iii. Remaining outfitting tasks

2. A balance exists for waiting on delayed documentation and using assumptions to start activities early.

   (a) Projects that have a forecasted overrun that exceeds 11% should use assumptions to start activities early.

   (b) Projects that have not a forecasted overrun that exceeds 11% should force activities to wait on delayed documentation and prioritise activities that can continue.

To end, this thesis underlines the importance of documentation quality in shipbuilding projects. Conclusions identify areas for improvement for DTC-projects. At the same time, the outcome serves as a guideline for project managers.

Contents

List of Figures

List of Tables

Nomenclature

Vectors

$\boldsymbol{W}$	Work Vector 1 stands for 100% 0 for 0%	%
$\boldsymbol{WN}$	Work Now Vector, Boolean entry for each activity	–
$\boldsymbol{W_D}$	Work done for a given run	%
$\boldsymbol{MLV_d}$	Most Likely Value for initial duration of tasks	days
$\boldsymbol{BCV_d}$	Best Case Value for initial duration of tasks	days
$\boldsymbol{WCV_d}$	Worst Case Value for initial duration of tasks	days
$\boldsymbol{MLV_c}$	Most Likely Value for initial cost of tasks	€
$\boldsymbol{BCV_c}$	Best Case Value for initial cost of tasks	€
$\boldsymbol{WCV_c}$	Worst Case Value for initial cost of tasks	€
$\boldsymbol{V_m}$	Activity Sequence Vector	–

Matices

$\boldsymbol{DSM}$	m x m matrix with task dependencies ’1’ and no dependencies ’0’	
$\boldsymbol{DSM_1}$	mxm matrix with rework probabilities	
$\boldsymbol{DSM_2}$	mxm matrix with rework impact	

Scalars

m	Number of tasks	
n	Number of process runs in a simulation	
dT	Time step, 1 day	
C	Cumulative process cost of a given run	€
S	Cumulative process duration for a given run	days
P_{val}	P-value to scale TV values to probability space	%
α	Required stability of output distributions (% change allowed between batches of runs)	
b	Run batch size between output distribution stability checks	
COR	Correlation between sampled activity duratrion and cost; 0.9	
T_s	Target value process duration	days
T_c	Target value process cost	€
A_{bound}	% of activity that is allowed to be unfinnished for subsequent activities to start	%

1
Introduction

A drastic change is seen in shipbuilding processes of European companies over the past decades. Traditionally one company was producing a vessel, today the number of involved companies has increased significantly (Levering et al., 2013, Mello et al., 2015a). Consequently, a shift is seen in performed activities. European companies moved back to there core business: construction of complex vessels (Holte et al., 2009). Other activities are outsourced to low-labour countries (Mello et al., 2015b). This change resulted in a separation between engineering and production (Mello et al., 2015a). Also, activities are no longer executed sequential (Kjersem and Emblemsvag, 2014). A reduction in delivery time is achieved; however, new challenges are created.

Currently, 60-80% of a vessel's value is outsourced (Held, 2010), leading to an increase in involved companies (Bolton, 2001). Shipbuilding is characterised by the variety and complexity of developed systems to customer requirement (Mello and Strandhagen, 2011). Besides, multidisciplinary teams create and manipulate large amounts of information (Andritos and Perez-Prat, 2000, Pedersen and Hatling, 1997). Hence, the information flow is complex and coordination of all involved is difficult. Subsequently, the effects of changes are hard to foresee, and controlling project execution has increased complexity over the past decades.

The Dutch Shipyard Damen implemented concurrency of activities and outsourcing. At the Business Unit, Damen Technical Cooperation (DTC), a unique strategy is seen; DTC builds not at owned yards. Although an unique approach is applied, the issues that are faced are not unusual. Current challenges at DTC are the motivation for this master thesis. At this moment, DTC has challenges sharing documents with other companies in a structured manner, misalignment of information on content and time and information quality does not always meet expectations.

This introduction presents the research field, problems and objectives of this study. First, Section 1.1 gives an overview of the problem environment Damen Technical Cooperation. Second, activities at Damen Technical Cooperation are placed on a broader picture by Sections 1.2-1.4. Insights from the background lead to a problem statement in Section 1.5 and research objectives in Section 1.6. The relevance of this research is noticed in section 1.7. Boundaries and limitations to the research goals are given by Sections 1.8 and 1.9. The introduction is concluded in Section 1.10 with the research strategy and report structure.

1.1. Damen Technical Cooperation

Damen Technical Cooperation (DTC) is a company within the Damen group since 1977 (DTC, 2020a). DTC offers a unique service for shipyards; a variety of material and service packages, shown in Figure 1.1, to make local building possible. DTC projects are unique within the Damen Group since vessels are produced at non-Damen owned yards. Nowadays, DTC has delivered more than 1500 projects in more than 60 countries and it has 70 employees (DTC, 2020b). A DTC solution comes with several benefits for the shipyard: transfer of knowledge, proven designs, access to the entire project portfolio and supplier network. Within the project portfolio distinction is made between *Standard Designs*; existing designs that need minor adjustments and *One-Off Designs*; new designs for a specific client. Local building can be beneficial in case of restrictions related to (DTC, 2020b):

- Landlocked areas
- Closed market
- Business development

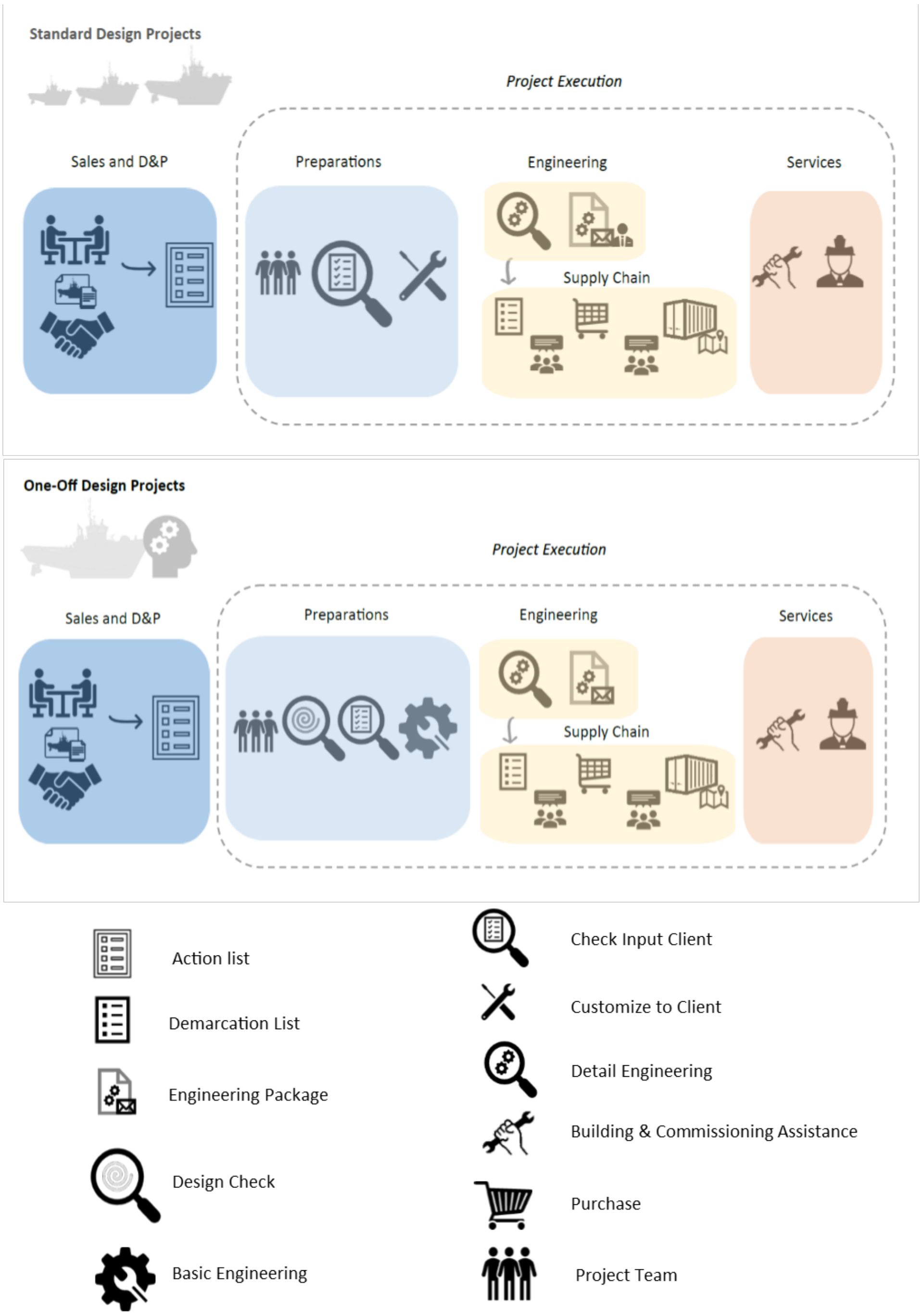


Figure 1.1: Possible DTC services; self-illustration

DTC knows two types of projects *Standard Design* and *One-Off* projects, the building process for both projects is explained. For both *Standard Design* and *One-Off* starts a project at the *Sales* and *Design & Proposal* departments, where the initial design, general arrangement and the contract are created. Once a contract is signed, an action list is made. The action list defines a project's tasks and assigns responsibilities. Two types of this document are known: a restricted and an unrestricted form. The unrestricted list marks the start of the project execution phase. After publication, a project team will be formed out of: engineering, material coordination, procurement, project management and services.

Different from *Standard Design* projects do *One-Off* projects need a *Design Check* and *Basic Engineering*, as shown by Figure 1.1. During the *Design Check* the design is checked on contract specifications, for example the design speed. As *Standard Design* projects are built from proven designs, this is not required. At *Basic Engineering*, also named *Systems Engineering*, all systems are identified and designed. Co-engineering with all co-makers is important to ensure system integration (DAM).

Identical in both working processes are: *Detailed Engineering*, *Material Coordination*, *Procurement* and *Services*. *Detailed Engineering* has a focus on creating required 3D information to generate a smooth production process at the yard. It serves as the link between engineering and production.

After engineering, *Material Coordination* receives requests for materials and transforms these into an order list. Care must be taken here, as not all required material for production are present on the engineering drawings. When the order lists are completed, they are sent to *Procurement*. At procurement suppliers are selected and orders are placed. Whereafter, *Material Coordination* takes care of logistics. Once materials are shipped, *building support* (services) starts at the location. *Project Management* is involved during the entire project and is responsible for the final project outcome.

Background of shipbuilding processes and successes

A background of shipbuilding processes and successes is given to place the shipbuilding process as executed at DTC in a broader context. This background is based on the literature. Section 1.2 discusses shipbuilding processes, followed by project success in Section 1.3. Influences on success are mentioned in Section 1.4. The background serves as the basis for the problem statement in Section 1.5.

1.2. Shipbuilding process

Based on the literature, an overview of shipbuilding processes is given. Studies on the Norwegian shipbuilding industry are most relevant for this overview, as the Norwegian shipbuilding industry has a focus on complex vessels and partly produces in 'low-wage' countries. First, different phases in a shipbuilding project are discussed in Subsection 1.2.1, followed by Subsection 1.2.2 capturing trends in shipbuilding of the past decades. At last challenges in current shipbuilding projects are mentioned by Subsection 1.2.3.

1.2.1. Different phases of a shipbuilding process

Shipbuilding processes exist in general out the phases; tendering, engineering, procurement, production and commissioning (Mello et al., 2015b). In these phases, different parties with a different role are involved. Logistics is not named explicitly, but both procurement and production can include logistics. The phases are explained briefly, to get a better overview of the shipbuilding process.

The process starts with the **tendering** phase, translating the needs of a potential shipowner to design requirements. The ship designer makes an offer based on the requirements.

Subsequently, the **engineering** phase starts, transforming the requirements into detailed technical drawings. Engineering requires technical information from both suppliers and the shipyard.

The engineering phase is followed by the **procurement** phase, here equipment and materials are ordered. The procurement phase includes negotiations with suppliers. Besides contact with suppliers, it is also essential to know when a shipyard needs specific equipment.

Eventually, the **production** phase starts at the shipyard, producing the vessel according to the technical specifications. Production requires the collaboration of multiple parties. These parties are responsible for various deliverables. The ship designer must deliver technical specifications and initial drawings. The timely delivery of equipment is the responsibility of the suppliers. The shipowner is involved in the monitoring process of the project execution.

Finally, the **commissioning** phase can start. In this phase it is assured that the vessel is ready to operate and performs according to contract specifications. Commissioning requires the involvement of suppliers, shipowner and the shipyard.

The collaboration of different parties is required to complete a shipbuilding project. Collaboration can be realised in multiple ways. Chosen strategies depend on the situation. However, a trend is seen in the execution of shipbuilding projects over the past years.

1.2.2. Trend in shipbuilding process

As mentioned before, a trend is seen at the shipbuilding processes of European companies over the past decades (Kjersem and Emblemsvag, 2014). Due to this trend, shipbuilding projects are defined by particular characteristics (Kjersem and Emblemsvag, 2014, Mello and Strandhagen, 2011). Thus the project environment of present shipbuilding projects changed by this trend.

The outsourcing of activities to low-labour countries resulted in a separation between engineering and production (Mello et al., 2015a). Also, activities are no longer executed sequentially but concurrently (Kjersem and Emblemsvag, 2014). Outsourcing and concurrency made it possible to reduce delivery time and overall cost (Kjersem and Emblemsvag, 2014, Mello and Strandhagen, 2011). Furthermore, the vessels became more customised (Andritos and Perez-Prat, 2000). These changes had an impact on the project characteristics.

Today, shipyards are 'assembly' yards (Kjersem and Emblemsvag, 2014) and are part of a global supply chain (Mello and Strandhagen, 2011). Thereby they are dependent on a large number of other parties (Levering et al., 2013, Mello et al., 2015a). Specialists and experts are present within this supply chain. The current process is described as engineer-to-order projects (Andritos and Perez-Prat, 2000, Ludwig et al., 2009), which is also, described by other industries. Characteristics of such projects are (Hicks et al., 2000, Wortmann et al., 1997):

1. Individual products are generally highly customized to meet individual customer requirements
2. There is production of a very low volume of engineered products (one of a kind to small series)
3. The main products have deep and complex product structures with levels of assembly process.
4. There are some components required in very low volume whereas others are required in medium or large volume.
5. Certain components are highly customized while others are standardized.
6. Some systems use advanced control while structural steelwork does not.
7. In general, high levels of customization lead to increased cost, higher risk, and long lead times

1.2.3. Challenges in shipbuilding

With the before mentioned trend in shipbuilding, new challenges arise. Challenges in engineer-to-order shipbuilding projects are unique. An overview of common issues in shipbuilding due to this shift is given. Important work is done by Andritos and Perez-Prat (2000), Held (2010) and Mello and Strandhagen (2011).

First of all, is there a need to involve multiple companies in the designing and building of vessels (Bolton, 2001). Held (2010) argues that 60-80 % of the value of a vessel is outsourced. Next, shipbuilding operations come with the generation and manipulation of large amounts of information (Andritos and Perez-Prat, 2000). Not only the large amount but also the complexity of the information creates challenges. The complexity is a result of both the characteristics of the operation and the number of involved working disciplines (Pedersen and Hatling, 1997). As Andritos and Perez-Prat (2000) said to illustrate the complexity: *'even vessels from the same series differ somewhat from each other'*. The complexity in the combination of both product structure and information flow puts additional pressure on coordination effort among all involved during design, engineering and production (Held, 2010). So it can be stated that information flow in shipbuilding was already complex before the change to 'assembly' yards and that the complexity only increased.

Ensuing, Hicks et al. (2000) identified the interface between product development (tendering, contract, concept, design engineering and procurement) and production as a challenge in the current situation. Mello and Strandhagen (2011) link interface issues to the variety and complexity of developed systems according to customer requirements. Mello and Strandhagen (2011) call these challenges a characteristic for the shipbuilding industry. Also, the wide geographical distribution of operations creates additional challenges (Mello and Strandhagen, 2011).

Moreover, specific challenges are identified for shipbuilding companies producing at various yards. Mello and Strandhagen (2011) conducted a case study at a Norwegian shipbuilding company building vessels at multiple yards in different countries. They observed that building existing designs at other shipyards can create new issues. An example illustrates this statement:

'An engineering package (which contains all the documentation, technical specifications, and drawings to produce a vessel) of a ship already produced in Norway, when sent to be produced in another country, can demand many changes that were not necessary before (i.e. new drawings, more technical information, the inclusion of new components, change of equipment etc.). This evidently incurs unexpected delays and cost that affect the performance of the whole supply chain (Mello and Strandhagen, 2011).'

Lacking integration of information and communications technology (ICT) also creates challenges. According to Sladoljev (1996) makes the absence of a clear information flow among all participants of a project efficient communication across multiple companies difficult. Today, *'evidence confirms that the information flow in shipbuilding is usually confusing and redundant'* (Mello and Strandhagen, 2011). This statement implies that no clear information flow is present.

Mello and Strandhagen (2011) argues that integration of ICT tools is necessary to track the progress of different stages and to share drawings in various formats. Also, they suggest that using business process modelling will help to obtain an integrated and streamlined workflow. Though, it is challenging to track workflow in shipbuilding. Andritos and Perez-Prat (2000) blame the considerable number of activities (around 3000 is usual) with their dependencies for this difficulty.

Despite their high importance ranking in literature, are ICT tools not widely implemented. Both Fleischer et al. (1998) and Andritos and Perez-Prat (2000) state that *there is considerable evidence that the shipbuilding industry lags behind other industries when it comes to the application of new technologies*.

1.3. Definition of project success

Project execution needs a measure before current concerns can be tackled. In this research, project execution is measured by project success. Therefore, a definition of project success is needed to understand the effect of project execution on success. Literature is analysed to find a suitable definition for the success of shipbuilding projects.

Project success and its definition have been studied by many for decades (Bannerman, 2008, Belassi and Tukel, 1996, Clark et al., 2007, Cooke-Davies, 2002, Love et al., 2006, Safapour et al., 2019, Ullah et al., 2018, van Niekerk and Steyn, 2011). Despite the effort, no consensus is found on a definition. Thus a definition of success has to be determined for this research. In many publications, project cost, duration and quality are named. These criteria overlap with Key Performance Indicators (KPIs) used by Damen (DAM). Rework is another frequently called factor concerning project success (Safapour et al., 2019).

For this research, *project success is defined by a time, budget and quality*. It is assumed that additional work, rework, happen if the quality is not sufficient. Rework links cost, duration and quality. Also, insufficient quality can cause issues during the warranty period. Figure 1.2 visualises the relation between cost, time, quality and rework.

1.4. Influences on success

Even though there has been effort to define critical success factors and to implement project management methods, still big issues are seen at execution (Minato, 2003, Ullah et al., 2018, Westin and Päivärinta, 2011). In particular project delays (Love et al., 2006, Ullah et al., 2018), cost overruns (Akampurira and Windapo, 2018, Love et al., 2006) and quality deficiencies (Fatawu, 2017, Slater and Radford, 2012). This section aims to identify influences on the success of engineer-to-order projects to get a better understanding.

At first, research on project execution in engineer-to-order shipbuilding projects is discussed in Subsection 1.4.1. Relevant work in shipbuilding is insufficient to define influences on project success. Thus, Subsection 1.4.2 searches for an industry that represents the shipbuilding industry. The product development and construction industry are used as a representative to identify influences on success in Subsections 1.4.3 and 1.4.4. Rework is the primary influence and is affected by project structure and documentation quality. Definitions of document and documentation are presented in the box to support

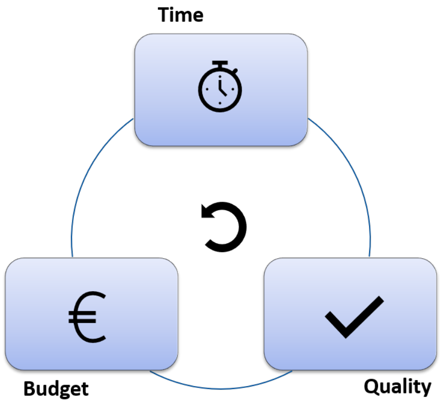


Figure 1.2: Link between project success criteria

this section.

> **Document**: Any source of information, in material form, capable of being used for references or study or as an authority. Examples: manuscripts, printed matter, illustrations, diagrams, etc (Otlet, 1990).
> **Documentation**: Information storage and retrieval systems (Buckland, 1997).

1.4.1. Project execution in engineer-to-order shipbuilding projects

As mentioned before, project execution has become more complicated by the introduction of concurrent engineer-to-order projects in comparison with standardised projects executed sequentially. To get a better understanding of this complexity and to get an overview of existing research, the project execution within engineer-to-order projects in shipbuilding is discussed. Here project execution is defined from contract signing to delivery. The most relevant work is written by researches studying Norwegian shipbuilding companies, research into companies at different locations is limited. Although most work has a focus on the Norwegian market, some research is performed on the Dutch shipbuilding industry.

In The Netherlands project execution of shipbuilding projects was of interest of a few: Coenen (2008), Levering et al. (2013), van der Mooren (2017) and Wesselman (2017). Coenen (2008) highlights the complexity of engineering by modelling the engineering process to understand its behaviour. Levering et al. (2013) studied historical developments of inter-organisational project practices in Dutch shipbuilding projects. To understand to what extend contemporary misfit in project practices is rooted in the past and results from path dependencies and lock-ins. Further worked van der Mooren (2017) on an improvement of the information flow of the secondary steel process using Model-Based Definition. Besides the work of van der Mooren (2017), Wesselman (2017) strives to improved collaboration between engineering and a remote production department by real-time communication introducing virtual reality.

Since only a few studies were found on the Dutch shipbuilding industry, is a representative found in the Norwegian industry. Multiple arguments can argue this comparison. Both Andritos and Perez-Prat (2000) and Ludwig et al. (2009) state that the West European shipbuilding industry can be seen as an engineer-to-order industry as the focus is on more sophisticated vessels. Ludwig et al. (2009) identifies that Norway does not have a large shipyard. The absence of a large yard can be explained to the even higher labour cost in Norway compared to The Netherlands. Furthermore, this implies that Norwegian shipbuilding companies built all large ships outside of Norway. All combined makes this the Norwegian shipbuilding industry a good comparison for the shipbuilding activities within the context of this master thesis.

While work on project execution is limited in The Netherlands, it is of the interest of several researchers in Norway. To start with Johannessen and Hauan (1992), who discussed the effect of the reorganisation of a Norwegian shipyard to improve organisational performance in 1992. Next, he identified that the internal 'carrying capacity' of an organisation depends on the coordination of the information creation processes (Johannessen and Olaisen, 1993). Further, Kolltveit et al. (2004) argues that the profit potential of projects is inversely related to the degree of uncertainty and that the chosen project strategy affects the uncertainty profile. Besides presented statements, Kolltveit et al. (2004) shows potential implications for project management.

The drastic change in the shipbuilding strategy came with new challenges and created new research areas. Supplier selection is one of these. Solesvik and Encheva (2007) constructed a model for branch analysis to support decision making for companies that source business processes abroad.

Another arising research area is supply chain coordination. Solesvik (2011) contributed by a study on inter-firm collaboration, identifying faced issues and directions for further research. Haartveit et al. (2012) presents a typology of different integration alternatives for separated designers and yards. The industry-relevant business factors that are affected by the level of integration are identified (Haartveit et al., 2012). Team dynamics is studied by Halsbog et al. (2015) in cross-cultural collaboration. From a case study on a Norwegian companies interaction with an Asian shipyard was concluded that team members of different nationalities behave almost similar within the same location, although the perception to others differs significantly (Halsbog et al., 2015). Also, defined Ahn (2015) key elements regulating the efficiency of complex engineering projects consisting of Norwegian operators, South Korean EPC contractor and subcontractors of various countries.

Despite other work, most valuable work is done by Mello and Strandhagen (2011), Mello et al. (2015a), Mello et al. (2015b) and Mello et al. (2017). Challenges and perspectives of the application of supply chain management are identified. Results showed that the successful application of supply chain management depends on improving the relationship with suppliers and adopting suitable information and communication technology (Mello and Strandhagen, 2011). Next, the role of coordination in delays is addressed by (Mello et al., 2015a). Mello et al. (2015a) asserts that coordination is a crucial factor for success. Moreover, factors affecting coordination are noted (Mello et al., 2015b):

1. **Integration of engineering and production**
   Comprehend several aspects related to seamless information flow, such as co-located engineering and production, integrated IT, aligned procedures, direct communication and similar organisational culture

2. **Size of the project**
   The size of the project is measured by the total scope of the project (size of the vessel, amount of equipment and number of project activities)

3. **Overlapping engineering and production**
   Engineering and production activities are executed at the same time, where production starts with incomplete design information.

4. **Maturity of the design or technology**
   Represents the level of knowledge available to minimise risks associated with the adoption of a new solution (design or technology)

5. **Collaboration between project partners**
   The commitment established between project partners to improve the overall performance rather than obtaining individual advantages through opportunistic behaviour

6. **Customer order changes**
   Represents the volume of change orders that need to be handled throughout the project. Such changes can have a cost or schedule impact that is difficult to estimate accurately.

7. **Production capabilities**
   The capability of a production team, to identify and solve potential problems on-site, is affected by skills, knowledge, experience and autonomy.

A soft systems methodology is used to expose areas to improve coordination Mello et al. (2017). However, it is hard to generalise results due to context-specific factors of this research.

To summarise, work on project execution in the Dutch shipbuilding industry is limited, but a representative is found in the Norwegian. Although more research is available in the Norwegian industry, research on influences on project success is still limited. Interviews and questionnaires are used to identify the importance of coordination and factors affecting coordination. Furthermore are suggestions presented for improvements. However, no implementation of these is found. Nor a proven solution to improve coordination in engineer-to-order projects.

From this can be concluded that previous work from shipbuilding is not sufficient to determine influences on project success.

The limited amount of previous work implies that work from other fields must be used to determine these influences.

1.4.2. Other industries

While engineer-to-order projects are common in shipbuilding, research on the success of such projects is limited. Whereas the success of engineer-to-order projects in construction is a hot topic. Various studies are present to understand influences on project outcome: delays among contractors (Hasmori et al., 2018, Sambasivan and Soon, 2007, Toor and Ogunlana, 2008), ranking the effects of delays (in Malaysian projects, (Ullah et al., 2018), factors affecting delays (Arditi et al., 2017, Doloi et al., 2012, Toor and Ogunlana, 2008), causes and impacts of poor communication (Hussain et al., 2018, Leje et al., 2019), root cause analysis of variations in construction tasks (Tiware and Kulkarni, 2013), impact of design and contract documentation (Fatawu, 2017, Macariola and Silva, 2019, Malinda, 2017, Minato, 2003, Slater and Radford, 2012), quality enhancement of engineering and design deliverables (O'Connor and Woo, 2017), causes and deficiencies in design documents (Assaf et al., 2018) and interface problems (Arain and Assaf, 2007).

Given the limited research in shipbuilding, a comparison to the construction industry is desired. As the comparison stays to project execution, construction is seen as a good comparison for shipbuilding (Mello and Strandhagen, 2011). Initially, the comparison is made by others too in literature. Furthermore are both industries project-based (Mello and Strandhagen, 2011). To continue, Asbjornslett (2002) points at the vast amounts of people, information, equipment and material to manage in both industries. With other words, there is a separation between engineering and production in both industries.

Furthermore, argues Gronau and Kern (2004) that both industries demand coordination of contributions of different parties; collaborative engineering. Another similarity is the interdependency between specific tasks of main suppliers and the shipbuilding company or design company in construction (Gronau and Kern, 2004). Moreover, both industries execute unique projects.

In comparison to all similarities, differences are identified. Mello and Strandhagen (2011) names the use of different sites in construction as a difference compared to shipbuilding. However, this does not apply to DTC projects (DTC, 2020a). Nevertheless, a difference is seen in duration; Mello and Strandhagen (2011) mentions that duration of shipbuilding projects is usually higher than construction projects. Another difference is seen in building material: steel versus concrete. Due to differences in material properties, it is harder to make changes in concrete than in steel. This difference makes the construction industry more conservative than shipbuilding for design changes.

Interest in the performance of complex projects is not only seen in construction but also in product development. Companies are under increasing pressure to sustain their competitive advantage by reducing product development time and cost while keeping high quality. The development process of complex products involves a large number of tasks and involved parties. This large number makes it hard to manage the interactions among tasks and people and to predict the effect of a single change (Eppinger et al., 1994). Rework is seen as the critical factor affecting project success in product development (Eppinger and Browning, 2012). To improve this situation, so to reduce rework, multiple solutions are proposed in terms of models to get insight into this complex structure (Hameri et al., 1999). So created Cho (2001) and Browning and Eppinger (2005) a model for managing complex engineering projects; is the effect of process architecture studied by Browning and Eppinger (2002); are models used to identify project risks at an early stage (Browning, 2000) and is a link made between the process and customer value (Browning, 2003, Browning and Heath, 2009).

Besides the link with the construction industry, research on product development processes has potential. Also, here it covers complex, unique projects, involving multiple disciplines. It is expected that modelling techniques can enrich applicable techniques to study the effects on project success.

1.4.3. Rework

As before mentioned rework is a critical factor in project success. Research in product development indicates multiple causes of rework. Studies in the shipbuilding industry are limited, so a link is made to product development.

Concurrent projects and coordination difficulties are usual in product development too. Concurrency increased complexity to manage iterations among tasks and people; *it may be impossible even to predict the impact of a single design decision throughout the development process*. Coordination difficulties are noticed by others too. Product development projects are described as a complex web of interactions (Browning and Eppinger, 2002). To ensure efficiency *data that activities need to do their work effectively must be available in the right place, at the right time, and in the right format* (Browning and Eppinger, 2002). Thus, improving efficiency comes with streamlining information flow. Flow disruptions cause rework and affect efficiency. In the context of process information flows, rework is caused by Levardy and Browning (2009):

1. **Inherent coupling**
   Activities are structurally interdependent and cannot be executed without assuming, exchanging, checking and iteratively updating information.

2. **Poor activity sequencing**
   Information is created at the wrong time (often too late), which forces other activities to wait or make assumptions.

3. **Incomplete Activities**
   Information needed by later activities is not fully available, even though the earlier activities have started.

4. **Poor communication**
   Information is not transmitted clearly, promptly, or appropriately.

5. **Input changes**
   External information (or proxy assumptions) used by activities to do their work is subsequently chained, necessitating rework of those activities and potentially many others that have followed.

6. **Mistakes**
   Incomplete information is inadvertently created and later discovered to be erroneous, causing rework of portions of the process. According to Cooper, a more significant time lag until this discovery amplifies the effect.

Table1.1 links cause of rework as defined by Levardy and Browning (2009) to characteristics of engineer-to-order projects and document quality. Engineer-to-order projects were already discussed in 1.4.1. Research on the effect of project document quality on success is mainly found in the construction industry. Subsection 1.4.4 gives and overview of this work.

Table 1.1: Causes of rework linked to the characteristics of engineer-to-order projects and documentation quality

Causes of Rework	Engineer-to-order	Documentation Quality
Inherent coupling	x	
Poor activity sequencing		x
Incomplete Activities		x
Poor communication		x
Input changes		x
Mistakes		x

1.4.4. Project documentation quality

Information should be timely, complete, good communicated, stable and free of errors to execute a project smoothly. In projects project information is stored on project documentation. The quality of this documentation affects success, as it can generate rework. Table 1.1 illustrates the relationship between documentation quality and causes of rework. ISO (2005) defines quality as the degree to which a set of characteristics meets requirements. Project documentation is used to store and share project information among the various parties involved. Thus project documentation quality is defined as the required characteristics to full fill the requirements of the involved parties. This subsection gives a brief overview of the influence of documentation quality for different locations.

Documentation quality affects the success of construction and engineering projects at European companies (Berard, 2012, Westin and Päivärinta, 2011). Poor documentation quality has a significant influence on delays, rework and cost overruns in South Africa (Toor and Ogunlana, 2008). Malinda (2017) exposes factors affecting documentation quality in South Africa. Poor documentation quality causes most quality issues in construction projects in Ghana (Fatawu, 2017). Slater and Radford (2012) register documentation quality issues in the Australian building industry. Eloranta et al. (2001) accent that *'many problems that occur in projects can be solved (directly or indirectly) by an improvement in documentation distribution among project partners'*. Thus it is concluded that insufficient project documentation quality harms project success.

1.5. Problem statement

The motive for this master thesis is the current challenges at DTC regarding an imperfect procedure for documentation sharing, misalignment in documentation flow and documentation quality issues. The occurrence of these issues is not unique—however, the matters described below to a broader problem in the execution of projects.

Rework has a significant effect on success. Causes of rework are linked to the characteristics of engineer-to-order projects and documentation quality by Table 1.1. Research on engineer-to-order characteristics and documentation quality gave insight into the problem. Here, interference is essential. Engineer-to-order characteristics nor documentation quality create the issues seen in itself. A combination of engineer-to-order characteristics and documentation quality define the problem.

The characteristics of engineer-to-order projects affect coordination, while coordination becomes more critical in concurrent engineer-to-order projects. Besides coordination difficulties, engineer-to-order projects are characterised by a complex network of activities. Activities with a large number of dependencies increase the risk of rework.

'A combination of engineer-to-order characteristics and insufficient documentation quality causes quality issues, leading to rework in shipbuilding. Subsequently, rework harms project success in terms of duration and cost.'

1.6. Research goal and questions

Research goal and questions are formulated to give this project direction.

1.6.1. Research goal

The main goal of this project is to **propose** a project documentation sharing **strategy** that **improves** project execution. Where the proposed strategy is a result of insights of the effect of project documentation on success, by analysing the consequences of not getting the right information, in the right form, to the right people at the right time.
Besides the main goal, the project has the following sub-goals:

1. Define requirements for project documentation flow
2. Prioritize areas for improvement (for project documentation)
3. Determine impact on project success
4. Verify documentation sharing strategy
5. Propose improvements to documentation sharing strategy

1.6.2. Research questions

The mean research question is formulated to achieve project goals:

In what manner can project execution at DTC be improved, by providing insight into the effect of the quality of shared project documentation on project success?

Answers to the sub-questions will formulate an answer to the main research question:

1. How does project documentation quality affect project success?
2. How can insight in documentation flow be obtained for engineer-to-order projects?
   (a) Which current solutions strive to reduce the negative effect of documentation flow on success?
   (b) Which models can be used to describe the project environment of engineer-to-order projects?
   (c) Which methods/tools are used by others to analyse documentation flow?
3. How can a useful model be built for documentation flow analysis of engineer-to-order projects?
   (a) How to develop a model for documentation flow analysis for engineer-to-order projects?
   (b) How to evaluate a model for documentation flow analysis for engineer-to-order projects?
4. How can documentation flow analysis be used to improve project execution of a DTC-project?
   (a) How can the impact of quality attributes be prioritized for a DTC-project?
   (b) How can documentation sharing strategy be used to improve project execution?
   (c) What potential improvement can be found by the proposed documentation sharing strategy?

1.7. Research relevance

Time and cost overruns caused by disturbances in the information flow are a significant problem in shipbuilding projects (Mello and Strandhagen, 2011). Currently, it is a difficult time for the maritime industry, which makes the issues more severe.

Problems are not limited to the shipbuilding industry. In general, the issues are relevant for project-orientated sectors with a significant number of parties. Examples are construction and product development industry.

By the proposed solution, it becomes possible to analyse the effect of documentation flow on project execution. Areas with an enormous negative impact can be highlighted and serve as input for decision making. Next potential changes to the process can be tested before implementation. Further makes the proposed solution it possible to define an alternative strategy once changes are required. Also, the proposed solution supports process understanding. Thus the proposed solution can help to improve a companies market position.

Besides benefits for industry, comes the proposed solution also with benefits for society. Although benefits are somehow related, so benefits community from a performing industry in terms of employment opportunities. Next, is performing business in a niche market important for society to keep specific know-how relevant. Besides, are well-performing business beneficial in the current uncertain times due to the COVID-19. Several companies are supported with community money, so it is useful for society if this money is spent well.

Further, the proposed solution has scientific relevance. Issues with documentation flow are not unique for the shipbuilding industry. The problem is also identified in the construction and product development industry. Different industries use different approaches to coop with these issues. A gap is identified in strategies.

The proposed solution contributes to science as it combines approaches of the construction and product development industry. Combining multiple methods can lead to new insights into the problem. Next is research on this topic limit in shipbuilding. Thus this master thesis does not only contributes to the research on documentation flow in shipbuilding, but it is also relevant for other applications in a project environment.

1.8. Scope

Limitations are set to the research objectives, to keep research feasible for the available time.

First of all, the research is limited to documentation flow at DTC and towards a DTC-yard. Other documentation flows are not included, such as flows to other business units or suppliers. Further, project execution is defined as the period starting at release of the unrestricted action list and ending at delivery. Improvement of execution is an increase in project success. Next, the research is limited to DTC solutions for one-off design projects including engineering packages, material packages and building support.

Documentation flow for engineer-to-order projects is modelled as the basis for a future adjustable documentation sharing strategy. Collection of required data is part of the scope, including the identification of yard needs. Existing data is used to build the model and conduct analysis. Furthermore, the content of contract documentation and general arrangement will not be questioned. However, the documentation flow between activities will be questioned.

Documentation flow is analysed on the effects on project success and to identify directions for improvement. A potential solution does not have to lead to an optimal solution. Optimisation of yard processes and activities performed at DTC in Gorinchem is not included. Suppliers and subcontractors are excluded from the documentation flow. Implementation, data security, continuous improvement and case studies at multiple DTC-yards are not included. A case study is included.

1.9. Assumptions

In this research, assumptions are made to capture the problem, to find solutions and to answer the research questions. These assumptions create limitations to which extend the findings of this project can be used in other circumstances—starting with the link between shipbuilding- and construction industry—followed by separation of engineering and production departments. Next, assumptions regarding simulation and client input will be discussed. The impact of the assumptions on different DTC solutions is discussed last.

The essential assumption of this research is the isolation of the impact of documentation flow on success. Other factors affecting success, like poor scheduling or external disturbances, are not taken into account. Thus only the effect of documentation flow is studied, while interference with other influencing factors is eliminated.

Next, it is assumed that lessons can be learned from the construction and Norwegian shipbuilding industry. This comparison is mainly argued by similarity in occurring problems—namely, project delays and cost overruns as a result of insufficient project documentation quality and coordination. Additionally, the similarity is confirmed by internal sources. Thus reported issues in the literature are relevant for DTC projects. However, the construction industry is more conservative regarding design changes. The difference relates to used building material; it is harder to make changes in concrete than in steel due to different material properties. As a result, it is expected to find more conservative results than when similar studies were present for shipbuilding.

This study is bounded to projects with separation between engineering and production where engineering is responsible for coordination. Coordination is a crucial factor for success with separation. Results are restricted to projects with similar characteristics. Thus care is needed when outcomes are used in other industries.

As a model is used to model and analyse documentation flow in this project; it is limited to characteristics of engineer-to-order shipbuilding projects. Assumptions can restrict the use of this model for other characterised engineer-to-order projects.

Client input is required to deliver project documentation according to needs. Since clients are located all over the world, it is difficult to reach all. Hence site managers are used as client representatives. Site managers often have years of experience at one or multiple DTC-yards and are working for Damen (which makes contact easier). This approach can affect results by perception differences.

Differences between DTC and other shipbuilding companies have to be noted. Most companies build on owned yards, DTC does not. Also at DTC yards can differ from project to project. Not only yards can differ, but also responsibilities. Thus building location and terms are additional parameters. It is hard to foresee if one model can capture a wide variety of options. It is not feasible concerning the available time to build multiple models. So, a complete option is modelled first. It is assumed that model outcomes can be transferred to less complete options.

Further is assumed that it is harder to transfer from less complete to completer. However, priorities for improvements differ. Thus the application of the model on less complete options can result in less promising results in comparison to a complete option.

1.10. Report structure

The Information Systems research framework (Hevner, 2007, Hevner et al., 2004) structures this report. The framework is developed for understanding, executing and evaluating research on information systems. A study on the information flow within shipbuilding projects can be categorised as a study on information systems. Behavioural-science and design-science are combined in the framework (Hevner et al., 2004). Three domains are present: environment, design science research and knowledge; an overview is provided in Figure 1.3. Chapters are linked to different fields, and research questions are related to chapters, as shown by Table 1.2.

The second chapter contains the knowledge base of the framework. Different research questions are answered based on the literature. An in-depth analysis of the effect of documentation quality on success is presented. Multiple databases are used to search for relevant research such as Google Scholar and the online databases of TU Delft Library. Current solutions are discussed, and an overview and selection are made of methods to model information flow.

The output of the knowledge domain is used in chapter three for the Design Science Research Domain. Chapter three discusses the development of a DSM-based discrete-event Monte Carlo simulation. Verification and validation are also included in this chapter.

The developed model is applied in the DTC-project environment in chapter four. The outcome of the Design Science Research domain is tested to business needs. The application includes model parameter scaling, the ranking of quality attributes and the proposal of an alternative documentation sharing strategy.

Conclusions and recommendations are discussed in chapter five, and serves as a feedback loop from the information system research to the knowledge base. This feedback loop captures contributions to the literature.

Table 1.2: Structure of chapters and research questions

Domain	Chapter	Research Questions
Knowledge base	2	1, 2a-c
Develop	3	3a
Evaluate	3	3b
Apply to environment	4	4a-c
Additions to the knowledge base	5	main research question

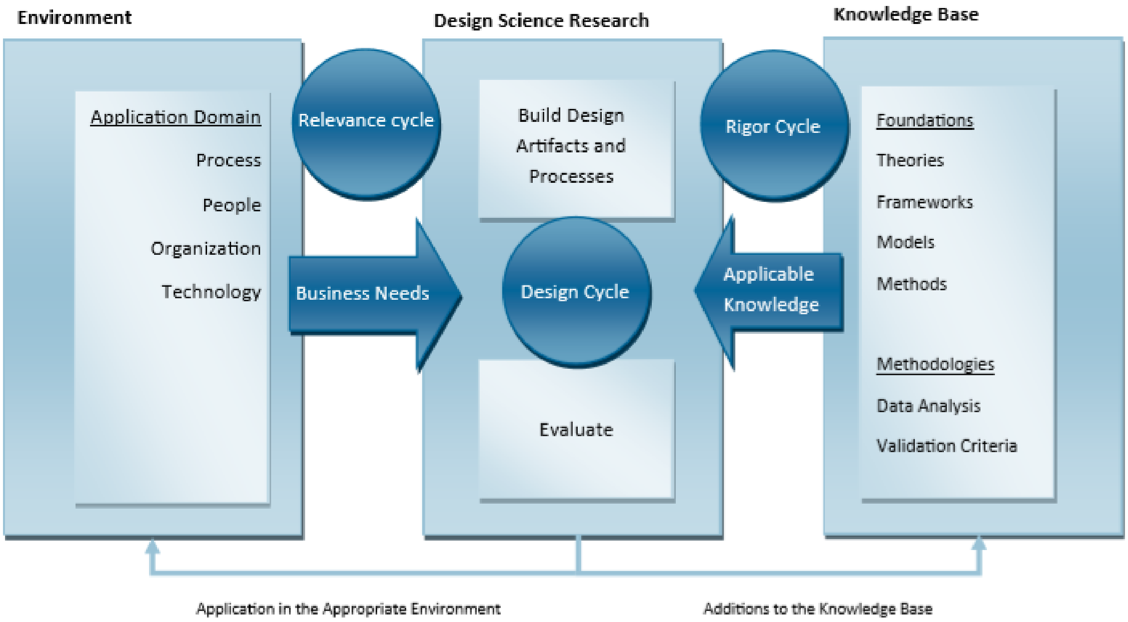


Figure 1.3: Information systems research framework (Hevner, 2007, Hevner et al., 2004)

2
Literature Study

In the previous chapter, an introduction is given to the problem area, research questions and limits of this project. This chapter contains a literature study on project success affected by documentation flow. This chapter aims to answer multiple research questions: 1) 'How does project documentation quality affect project success?', 2a) 'Which current solutions strive to reduce the negative effect of documentation flow on success?', 2b) 'How can a model describe the project environment of engineer-to-order projects?' and 2c) 'Which methods/tools are used by others to analyse documentation flow?'. Different viewpoints are used to answer these questions. Results of this chapter lead to a research methodology for the remaining part of the project.

First, the viewpoint of different industries is used to analyse the effect of documentation quality and directions for improvement in Sections 2.1, 2.2 and 2.3. A gap in approaches is identified among different industries in Section 2.4. Then is moved to another viewpoint, models for documentation flow modelling. Different models and their applications are discussed in Section 2.5. A Design Structure Matrix (DSM) model is selected out of four models in Section 2.6. An overview of analysis techniques is given in Section 2.7. A plan for the remaining research is described in Section 2.8.

2.1. The influence of documentation quality on projects

This section aims to reveal the influence of documentation quality on project execution. Westin (2013) states that quality attributes measure documentation quality. Additionally, quality attributes make it possible to describe documentation quality in more detail. Moreover, quality attributes can be linked to causes of rework as mentioned in Subsection 1.4.3. Subsection 2.1.1 explains quality attributes. These quality attributes reoccur in research on the importance of documentation quality. Subsection 2.1.2 handles, based on literature, the relative importance of the quality attributes.

2.1.1. Documentation quality attributes

Since quality attributes are a measure for documentation quality, a basic understanding is needed to place previous work in perspective. Many quality attributes are seen in the literature. However, only the essential attributes to understand the results of prior research are discussed in this subsection.

Based on work by Akampurira and Windapo (2018), Akampurira and Windapo (2019) and Westin and Päivärinta (2011) a list of relevant quality attributes is determined:

1. **Accuracy**
   Drawings and other documents are free of errors and omissions.

2. **Certainty**
   Drawings and other documents do not require changes or amendments.

3. **Clarity**
   Drawings and other documents are legible and easily interpreted.

4. **Completeness**
   Drawings and other documents provide all the required information for construction.
5. **Conformance**
   Drawings and other documents meet the requirements of performance standards and statutory regulations.
6. **Consistency**
   Information provided on the drawings and other documents is consistent in meaning.
7. **Coordination**
   Information on the drawings and other documents are thoroughly coordinated between. design disciplines
8. **Final Checking**
   Drawings and other documents are adequately checked before release to the contractor.
9. **Legibility**
   Drawings and other documentation are physically easy to read.
10. **Relevance**
    Trade specifications and details are specific, relevant and appropriate to the project.
11. **Representation**
    Drawings and other documents correctly represent the geological (subsurface), topography (Surface) conditions, including existing utilities and structures.
12. **Standardisation**
    Drawings and other documentation make use of standard details and specifications in drawing and other documents.
13. **Timeliness**
    Drawings and other documentation are supplied when required to avoid delays.

Additionally, quality attributes create the possibility to describe documentation quality in more detail. In the introduction linked Table 1.1 causes of rework and documentation quality. Using quality attributes to describe documentation quality makes it possible to connect the causes of rework to specific quality attributes. Table 2.1 presents this relationship in more detail. Except for *Inherent Coupling*, all causes are affected by documentation quality. Not all listed quality attributes are named in Table 2.1. Still, a complete list is used as other quality attributes come back in research on documentation quality in Subsection 2.1.2.

Table 2.1: Link causes of rework to documentation quality attributes

Causes of Rework	Documentation Quality
Inherent coupling	-
Poor activity sequencing	Timeliness
Incomplete Activities	Completeness
Poor communication	Clarity, Conformance, Coordination, Legibility
Input changes	Certainty
Mistakes	Accuracy

2.1.2. Relative importance of quality attributes

Exceeding deadlines and budgets motivated researches to study the effect of documentation quality on the execution of engineer-to-order projects in construction. Despite the broad interest, is research into the relative importance of quality attributes limited. Questionnaire and interview-based studies are conducted at different locations by: Akampurira and Windapo (2019), Tilley (1998), Tilley and Mcfallan (2000), Minato (2003), Chen et al. (2013) and Westin and Päivärinta (2011). Outcome differences are present, possibly caused due to local context or the used research methodology. However, a consensus is found on the importance of documentation quality. The importance is underlined by work of Malinda (2017), where *a correlation is found between project documentation quality and achievements in project objectives in terms of quality, cost and time*. This subsection gives an overview by a discussion on previous work.

Tilley and Barton (1997), Tilley (1998), Tilley and Mcfallan (2000) and Phililps-Ryder et al. (2013) studied the importance of quality attributes in Australia. Among these studies outcome differences are identified. Tilley and Barton (1997) started research in the Austrian context with a survey-based study to identify the most significant effects of poor documentation quality. Both contractors and designers were invited to workshops to identify changes in documentation quality over the past 10-15 years. Contractors identified *accuracy* as affecting process efficiency most. Although the designers acknowledged a lack in *accuracy*, argue designers for a different ranking. Later, Tilley (1998) extended this research by incorporating the steel industry perspective. In contrast to the noticed perspective difference by Tilley and Barton (1997), showed results of Tilley (1998) consensus among contractors and designers. Accuracy, completeness, final checking and certainty declined most significant (Tilley, 1998). This result is in line with earlier research (Tilley and Barton, 1997). Another survey-based study among designers by Tilley and Mcfallan (2000) ranked *accuracy*, *clarity* and *final checking* as the most important quality attributes. Phililps-Ryder et al. (2013) show different results for a similar study among subcontractors. In this research *accuracy*, *conformity* and *completeness* were named as most important. So again, a perception difference was identified.

Akampurira and Windapo (2019) ranked relative importance of quality attributes for the South African construction industry. The research was based on a cross-sectional web-based questionnaire survey among civil engineering design consultants and constructors. The questionnaire was developed from insights from the literature. Results showed that *legibility*, *coordination* and *consistency* are the most important quality attributes for design documentation. *Standardisation*, *relevancy* and *certainty* are ranked lowest. Other interesting findings are a perception difference in the relative importance of quality attributes among designers and contractors. Akampurira and Windapo (2019) concludes from the perception difference that *it depends on your role in a project what you rank as important*. Also, a difference was found in the perception of the current state; contractors ranked the current state quality lower than designers. Akampurira and Windapo (2019) explains this as: *'Contractors are more critical to quality of the design documentation'*.

Questionnaire-based surveys and interviews are also used to study the problem in Japan (Minato, 2003). Minato (2003) discusses design documentation related problems and their impact on the Japanese construction process. A slightly different approach is used as seen in work of Tilley and Mcfallan (2000) and Akampurira and Windapo (2019), as Minato (2003) identifies least incorporated quality attributes: *certainty*, *coordination*, *accuracy*. By this approach the least incorporated are seen as the problem cause and ranked most important. However, incorporated attributes can also have a high importance. Thus this approach can lead to different results, in comparison to other work.

Others also use the approach of Minato (2003); e.g. Westin and Päivärinta (2011) and Chen et al. (2013). Westin and Päivärinta (2011) conducted a case study on a large European multi-discipline construction and engineering company to study information quality problems. Results showed that most problems related to: *timeliness*, *accuracy* and *consistency*. Later identified Chen et al. (2013) key problems in documentation quality in Malaysia: *accuracy*, *clarity* and *timeliness*.

To conclude, two approaches to rank the relative importance of quality attributes are present in literature: studies identifying essential attributes and studies identifying attributes that cause most problems or are least incorporated. Furthermore, perception differences are seen among different studies. Results of Minato (2003) are interpreted as an exception, which could relate to the context of the study (Japan). An overview of named quality attributes is given by Table 2.2. From this can be seen that *accuracy* is named most (4 times). From this, it is concluded that the importance of *accuracy* is not affected by the different approaches nor due context-specific factors. *Coordination*, *consistency* and *Completeness* were named by studies using a different approach. So it is not expected that the different approach influences these attributes.

On the other hand, is *timeliness* only named in studies focussing on factors that are not incorporated in the current situation. Not naming *timeliness* indicates that the factor is hard to incorporate. Attributes that are only named once (Certainty, Conformity, Final Checking and Legibility) are expected to be influenced by factors of the local context. However, some conclusions can be made at this point; not enough information is available to determine a relevant quality attribute ranking for shipbuilding.

Table 2.2: Important quality attributes in the literature

Attribute	(Akampurira and Windapo, 2019)	(Tilley and Mcfallan, 2000)	(Chen et al., 2013)	(Westin and Päivärinta, 2011)	(Minato, 2003)	Number
Accuracy		x	x	x	x	4
Consistency	x			x		2
Coordination	x				x	2
Completeness		x		x		2
Clarity		x	x			2
Certainty					x	1
Conformity		x				1
Final Checking		x				1
Legibility	x					1
Timeliness			x	x		2

An overview of named quality attributes is given to place Table 2.2 in perspective.

- Westin and Päivärinta (2011)
  Currency, volatility, uniqueness, appropriate amount of data, accessibility, credibility, interpretability, usability, derivation integrity, conciseness, maintainability, applicability, convenience, timeliness, comprehensiveness, clarity, traceability, security, accuracy, objectivity, relevancy, reputation, ease of operation, consistency, completeness and interactivity
- Akampurira and Windapo (2019)
  Accuracy, Completeness, Timeliness, Coordination, Clarity, Conformance, Certainty, Relevance, Consistency, Standardization, Legibility and Representation
- Tilley and Mcfallan (2000)
  Accuracy, Completeness, Timeliness, Coordination, Clarity, Conformance, Certainty, Relevance, Standardization, Final Checking
- Chen et al. (2013)
  Accuracy, Clarity, Timeliness, Conformance, Consistency, Coordination
- Minato (2003)
  Completeness, Clarity, Consistency, Accuracy, Standardization, Relevance, Timeliness, Coordination, Certainty, Conformity, Representation

2.2. Link results from construction to shipbuilding

Ranking quality attributes for shipbuilding is hard without previous work in this context. Factors affecting quality attributes are compared to determine in what extend findings in construction can be applied at shipbuilding. It is assumed that results are relevant for shipbuilding if factors affecting documentation quality most in construction are present in shipbuilding too.

Industry-related factors affect quality most in the construction industry (Akampurira and Windapo, 2018). Also, design professional related aspects score high (Akampurira and Windapo, 2018). Besides, the design firm and the client influence documentation quality in construction (Akampurira and Windapo, 2018). Some of these factors are more bounded to the construction industry than others. Highest ranked factors are discussed and are compared to shipbuilding.

To start with industry-related factors, which are ranked highest in construction. According to Gronau and Kern (2004) are price-competitive procurement approaches and rigid contracts seen in shipbuilding. This statement implies that suppliers are selected based on price. Whether low design fees are an issue in shipbuilding is not clear from the literature, it is not mentioned. However, a price-competitive procurement approach can result in low-fees for suppliers. Poor communication among multi-disciplinary teams is seen as a design professional related factor in the construction industry (Love et al., 2006, Minato, 2003, Phililps-Ryder et al., 2013, Tilley et al., 1999). In shipbuilding, poor communication among multi-disciplinary teams is a factor that relates to multiple categories (Mello et al., 2017, Mello and Strandhagen, 2011, Mello et al., 2015b). Mello and Strandhagen (2011) argues that inter-firm collaboration, which includes communication, is challenging in shipbuilding. Since it is expected that suppliers also do business with competitors of shipbuilding companies, as this is related to characteristics of the shipbuilding industry, it is seen as an industry-related factor.

On the other hand, poor communication among multi-disciplinary teams is also seen as a design firm related factor. Especially communication with different firms is a problem in the shipbuilding industry (Solesvik, 2011). Lacking ICT systems are blamed for this problems in a number of publications: Mello and Strandhagen (2011), Andritos and Perez-Prat (2000), Fleischer et al. (1998), Sladoljev (1996) and Ying-Han and Heiu-Jou (2004). Although the problem already was addressed in 1996; it is still present. Some companies developed in-house systems for data sharing. Briggs et al. (2005) conclude that the biggest problem for shipbuilding is increasing data and systems that do not communicate with each other.

Moreover, if they do, it is a specific solution that only works for particular systems (Mello and Strandhagen, 2011). Furthermore, Tann et al. (2005) argues that stand-alone systems cause problems for integration. Several standard systems for data exchange are proposed to overcome the issue: STEP (standard for the exchange of product model data) (Benthall et al., 2003) and XML (extensible markup language) (Ying-Han and Heiu-Jou, 2004). Nevertheless, these systems are not widely implemented.

Coordination between different design disciplines is also an issue for shipbuilding (Mello et al., 2015a). Due to the shift to engineer-to-order projects over the past years; vessels are offered with shorter delivery times (Kjersem and Emblemsvag, 2014). Shorter delivery times increase pressure on design professionals (Kjersem and Emblemsvag, 2014). It is expected that when pressure increases, documentation quality will drop. Also, communication becomes more important with the shift to a global supply chain (Mello et al., 2015b).

Changing client requirements are present in shipbuilding too (Iakymenko et al., 2018, 2019, Norbye, 2012), which implies that the factor is relevant. Although, studies on unrealistic client expectations or insufficient documentation quality of client provided documentation are not found.

To conclude, many factors identified as 'affecting documentation quality' in construction are also seen in shipbuilding. However, time for checking and coordination is not mentioned in studies. Also, unrealistic client expectations are not reported in shipbuilding literature. Although some factors are not named in shipbuilding, construction results ought to be relevant for shipbuilding. Nevertheless, the ranking could differ. Different rankings could relate to the lack in implementation of ICT systems in shipbuilding.

2.3. The importance of information flow modelling

Cost and time overruns are not unique for shipbuilding, construction or product development. Similar issues get attention in other industries from many in the literature. Many process owners strive for a more customer-orientated, value-stream based, end-to-end process orientation (Browning, 2014b). Process integration is essential to achieve this (Browning, 2002, Hunt, 1996). However, process integration necessitates process understanding (Baldwin et al., 1999, Browning, 2014b, Hunt, 1996).

Baldwin et al. (1999) claims that only a better understanding of the information flow among project participants can improve management. Information flow modelling increases process understanding and its importance (Ambler, 2004, Browning, 2014b, Hunt, 1996). Austin et al. (1996) discovers that a lack of information flow and dependencies knowledge lead to disturbed process flows. According to Browning (2014b) many fail in process integration due to poor process understanding and motivation. A role-play is developed by Browning (2014b) to open eyes and to stress the importance of process understanding.

Motivation issues are not the only cause of process integration failure, inappropriate models for information flow modelling affect integration (Browning, 2002, 2014b, Cho, 2001). An appropriate model is required for information flow modelling (Steward, 1981). Browning (2014b) states that extensive project processes are hard to manage due to size, complexity, and uniqueness. Different models are introduced to deal with this difficulty; e.g. process flow charts, Gantt charts, CPM, PERT, work breakdown structures (WBS) and formal procedures (Browning, 1998, Cho and Eppinger, 2001, Eppinger et al., 1994, Yassine et al., 2001). However, various models are implemented; many claims that such models are not suitable for project processes (Browning, 2014b, Cho and Eppinger, 2001, Eppinger et al., 1994, Oloufa et al., 2004, Yassine and Braha, 2003). Often multiple models are needed for one project, which creates issues with synchronisation (Browning, 2014b). Out sync models create a situation where some participants will ignore models (Browning, 2014b). Besides such models only include part of the information (Browning, 2014b, Yassine and Braha, 2003). Thus no clear overview is given.

A model must represent processes well, to get a representative overview. All models are wrong, but some models are useful. Thus a useful model is required for process understanding. Current methods can not deal with interdependent tasks and feedback loops and will not lead to a helpful model (Browning, 1998, Cho, 2001, Cho and Eppinger, 2005, Weske, 2007, Yassine et al., 2001, Yassine and Braha, 2003). Without process understanding, process improvement is almost impossible (Browning, 2002, 2014b, Cho, 2001). Thus a suitable information flow model is crucial for process improvements.

2.4. Eliminate negative effects of insufficient documentation quality

Low documentation quality harms project success in multiple industries. Section 2.3 emphasised the importance of problem understand. However, problem understanding is not the aim of this research. The objective is to propose a solution to reduce this adverse effect.

Currently used approaches to limit adverse effects are discussed for construction, shipbuilding and product development in Subsections 2.4.1, 2.4.2 and 2.4.3. A gap is identified in the different approaches among the industries in Subsection 2.4.4.

2.4.1. Solutions in the construction industry

Building Information Modelling (BIM) is an example of an ICT solution. Bryde et al. (2013) use secondary data from 35 construction projects utilising BIM, to analyse the benefits of BIM for construction project management. Significant cost savings were found (Bryde et al., 2013). Besides, benefits of BIM applications were found for: time reduction, improved communication and coordination (Bryde et al., 2013). Nevertheless, BIM also has disadvantages; it is not user friendly, which makes it less likely to implement (Bryde et al., 2013). Furthermore, BIM is a construction industry-specific solution. A system like BIM can only work when all involved companies cooperate in the system. In other words, a solution like BIM must come from the industry.

Besides BIM, other solutions are present. So used Hiremath and Skibniewski (2004) object-oriented modelling to capture information flows in construction projects. This method is used to evaluate and select critical information flows that are present for all construction projects. Thus Hiremath and Skibniewski (2004) introduced a non-project specific solution for complex information flows. The primary outcome of the study is an on-site project documentation tracking system. The solution has several benefits: improve processing cycle time of critical information flows, speedy retrieval of archived information, track all information flows precisely and creates a central source for critical documentation (Hiremath and Skibniewski, 2004).

Further, Akcay et al. (2017) did an attempt to streamline the information flow between multiple parties in a structural steel components supply chain by the Unified Modeling Language (UML) model. In this study, the information generation process is formalised. Face-to-face interviews, ethnographic observation and examination of frequently used documents are used to collect data at the different companies. This research contributed in terms of 1) formulation of an information flow process for structural steel, 2) identification of information generated at different stages, and 3) analysis of source and destination of information (Akcay et al., 2017). Besides UML models, are also other methods used for information flow modelling. So showed Bastos and Ruiz (2001) that C-Wf models could be applied. Also, Data Flow Diagrams can be used to capture data flow in a system (Ambler, 2004).

On the other hand, internet-based solutions are found. Weippert et al. (2003) compared different online ICT systems on remote construction projects by a survey. For this survey, four remote construction projects were used. This study aimed to identify critical success factors for successful implementation of such solutions. It resulted in 'best practice guidelines' as a direction of further research.

Besides innovative solutions for documentation sharing, other solutions are proposed. These solutions try to seek for a process improvement that increases documentation quality. So argues Ergen and Akinci (2008) that formalisation of information flow simplifies complexities in tracking information. It is noted that advanced technologies to streamline information flow are available. However, a formal way to identify the requirements and appropriate systems is lacking. The study is conducted on a precast concrete supply chain in the United States and results in an information flow framework for formalising information flow patterns in a supply chain.

Leje et al. (2019) provide insight in support of practical communication tools to increase project performance. In this survey-based study, 160 surveys are returned by Nigerian contractors. Further, Dogan et al. (2015) propose using social network analysis on e-mail communication to measure coordination performance. Other work by Arditi et al. (2017) identified that organisations with a 'clan' culture have fewer delays than organisations with a 'market' organisational culture. The study is based on a questionnaire survey among construction companies located in the United States and India.

2.4.2. Solutions in the shipbuilding industry

Solutions for documentation sharing gets less attention in shipbuilding than in construction. The limited research is in contrast with the frequency 'lack of proper ICT solutions' is named as one of the significant issues in shipbuilding. In-house developed systems for data sharing give enormous problems with integration. These solutions are not discussed here, as they are not seen as a real solution to the problem.

Benthall et al. (2003) proposed a standardised solution for data sharing: STEP (standard for exchange of product model data). This solution is based on ISO 1030, an international standard. The primary function of the model is to facilitate the exchange of product models between different computer-aided design (CAD) software. According to Benthall et al. (2003) is STEP able to transfer information regarding ship structures, piping, heating, ventilation and air-conditioning. It should be noted that CAD-related data is only a share of all project data. Thus by using STEP, the exchange of CAD data improves, but the problems remain for all other data types.

The problem was also acknowledged by Ying-Han and Heiu-Jou (2004); seeing that it is broader than issues with different CAD software. A knowledge management system for the full life cycle was developed in XML define by W3C (the world wide web consortium). With this system, collaborative management is combined with the advantage of the internet. By this combination, the speed of knowledge sharing increases, enabling it to get the right information at the right time. Although the by Ying-Han and Heiu-Jou (2004) proposed system sounds promising; no literature is found about the implementation of this system nor its success.

Although implementation is not described; it was a basis for further work of Cheng and Shaw (2012). XML systems are not widely implemented, but the use of the internet gained the attention of Cheng and Shaw (2012). Cheng and Shaw (2012) proposes cloud-based services for documentation sharing at different stages of shipbuilding. The study included ship specification files (Cheng and Shaw, 2012). Difficulties are identified in the integration of files as they have different formats, are stored in different databases or are stored at different departments. Another benefit of the cloud-based system is that it does not require high capital investments. Ongoing research acknowledges these difficulties and strives to eliminate them by a study on the implementation of an object-based approach instead of a document-based approach (NAVAIS, 2020). XML systems are valuable for object-based approaches (Ambler, 2004). Multiple partners are involved, among others, Damen Schelde Naval Shipbuilding, Damen Shipyards and Damen Shipyards Galati Romania participate in this research (NAVAIS, 2020).

Although different methods are proposed in the literature, documentation sharing problems are not solved. Proposed methods only handle part of the problem. STEP encounters issues related to CAD files, and both versions of the XML solutions deal with ship specification files. However, more documentation is required to build a vessel. Thus solutions for documentation sharing are not sufficient to solve the problem.

2.4.3. Solution in product development

The product development industry strives to limit the adverse effects of rework by the introduction of the Design Structure Matrix (DSM). The DSM is the basis of modelling, analysing and adjusting information flow among activities. Different DSM types exist Product DSM, Organisational DSM and Process DSM. Besides, various categories are identified: Static Architectures, Temporal Flow and Multi-Domain. Product and Organisation DSMs belong to the category Static Architectures. Static means that dependencies are physical the entire time. Process DSMs belongs to the Temporal Flow category, where the presence of elements changes over time. The last category, Multi-Domain, represents DSMs that consists of a combination of Static Architecture and Temporal Flow. This study focusses on process DSMs.

Work of Steward (1981) is seen as the beginning of substantial research on process DSMs. Steward (1981) captured a process by square-matrix-based models. Eppinger and Browning (2012) define a process as: *'a system of activities and their interactions comprising a project or business function'*. Another strength of the DSM is the ability to include different types of dependencies sequential, parallel, coupled and conditional. A process DSM is mainly used to model input-output relationships among the activities. Though, it can also be used to model the hierarchical decomposition of the process into activities (Browning and Eppinger, 2005). Browning (2016) asserts that *'a DSM view is especially advantageous when seeking to highlight cycles (iterations or rework loops), which are both prominent and problematic in project success'*.

Applications of process DSM are widely seen. Most of them apply the process DSM in order to improve processes, for which multiple approaches are used. DSM applications for process improvement can be grouped into four categories: 1) applications that gain insight in a process, 2) applications that focus on improve scheduling, 3) applications that make it possible to control a process and 4) applications that highlight at areas for improvement. Before applications in these categories are discussed, it should be mentioned that most work on process DSM has a focus on structuring and estimating a given project process (Browning, 2016). Besides this, other applications are seen as well.

The first category deals with the application of process DSM to gain insight to improve a process. A notable contribution is seen, in work predicting effects of changes in design or requirements. Effects of changes are hard to foresee in complex projects. Mello et al. (2015a) named this as one of the reasons coordination in engineer-to-order projects is so complicated. By change prediction, it became possible to improve efficiency in the development process of the reduction of aero-acoustic noise effects in commercial planes (Eppinger and Browning, 2012). Others used the process DSM to describe organisations, instead of the organisational DSM. This work made it possible to include organisational structures in a process and perform analysis on organisation network design. Chen and Huang (2007) use this approach to identify and qualify interactions among process elements. In addition, shows Chen and Huang (2007) a manner to decompose large interdependent groups into smaller manageable groups, to improve overall process performance. Similar to Chen and Huang (2007), Gomes and Dahab (2010) used the process DSM to describe organisational structures to find an approach to improve process performance. Including organisational structures made it possible to identify organisational requirements for process improvement, which also had a contribution to the improvement of collaboration between departments. Work of both Chen and Huang (2007) and Gomes and Dahab (2010) is done in the context of supply chains.

The second category focusses on DSM applications to improve scheduling (Gaertner et al., 2015, Zhao et al., 2008). This research includes different scheduling levels. Hence resource dependencies among various projects within a portfolio are studied (Cho and Eppinger, 2005). Understanding resource dependencies support companies to determine schedules for multiple projects with demand on identical resources. Others used the DSM to define appropriate project deadlines and budgets. Cost overruns and delays can be reduced by incorporating project complexity in set targets. Some put the application of the DSM for scheduling to another level; Levardy and Browning (2009) introduced a complex adaptive system based on a DSM structure, containing self-organising tasks according to simple rules.

Applications to control processes better belong to the third category. Procedures to monitor project progress are researched. Rogers and Salas (1999) proposed a web-based system for optimisation and control of a design project. This system connects the sequencing of activities with monitoring project status (Rogers and Salas, 1999). On the other hand, the DSM is used to distinguish the outcome of diverse strategies (Arcade et al., 1999). By this work, it became possible to control the outcome by strategy. Furthermore, the DSM is used to identify information flow disconnects (Browning and Eppinger, 2002). Disconnects disturb the process; by this application, disconnects are made visible. The awareness of disconnects makes it possible to act timely and to keep control of the process.

The last category deals with DSM applications to recognise areas for improvement. Browning and Eppinger (2002) and Browning (2002) used the DSM to model the impact of process architecture on process outcome, by a discrete-event Monte Carlo simulation. A process architecture is defined as structure of dependencies among activities. Browning and Eppinger (2002) *shows that a process architecture has impact on project outcomes and recommends a method to generate alternative architectures*. Ensuing, Tang et al. (2008) and Zhao and Cui (2009) applied the DSM to map information flows in multiple design tools, to capture all information in one 'meta-tool'.

Browning (2016) states that although a wide variety of utilisation is described at this moment, more applications of the DSM are possible. Today, most applications are on project processes. Nevertheless, it might be possible to apply DSM to repetitive business and production processes. Furthermore, names Browning and Eppinger (2002) that use of the process DSM should be integrated with risk management as it is a powerful tool for the prediction of risk mitigation steps. Additionally, he points at the absence of process DSMs in project management standards, in contrast to the attention process DSM gets in project management journals. He marks this point by *it seems that further visibility of process DSM methods and applications is needed in project management*.

2.4.4. Gap analysis - Comparison of solutions in construction, shipbuilding and product development

In shipbuilding, construction and product development data sharing can affect project success. Different industries use different approaches in attempts to eliminate adverse effects.

In shipbuilding research on both the effect of documentation flow on success and proposed solutions to eliminate negative influences is limited. Suggested solutions have a focus on documentation sharing with different systems. However, only a part of the problem is tackled by these solutions. Streamlining the information flow does not get attention. Also, documentation quality is not the focus.

Streamlining the information flow and documentation quality get attention in the construction industry. Different methods are used to model information flow in construction. However, most solutions have a separate focus. Although previous work addresses the importance of documentation quality, no suitable solution to secure the required quality is found.

In product development, streamlining information flow gets special attention. A design structure matrix structure is used to capture, analyse and adjust information flow in processes. However, by this approach, it is assumed that process architecture affects the project outcome. Thus, documentation quality is not included in developed DSM-based solutions. Notwithstanding successful applications are known, which underlines the importance of process understanding. A gap is identified in the different approaches to the problem by the construction and product development industry.

It is proposed to combine knowledge of the construction and product development industry to fill this gap. Modelling the information flow could facilitate further analysis of the effect of documentation flow in construction projects. In the end, also a contribution is made to shipbuilding literature, as the construction industry serves as a representative.

Research questions 1 and 2a are answered by Sections 2.1-2.4. Furthermore, a gap is identified in different approaches. However, all strive to process integration. Disconnects in information flow harms process integration and can be detected by information flow modelling. In shipbuilding projects project information is stored on project documentation. Thus, to model information flow, the documentation has to be modelled. This conclusion marks the end of the first part of the literature study; the following part Sections 2.5, 2.6 and 2.7 focusses on methods and techniques for documentation flow modelling. The second parts aims to answer research questions 2b and 2c.

2.5. Methods for documentation flow modelling

Documentation flow modelling is required to model information flow in shipbuilding projects. Multiple models are named in Subsections 2.4.1 and 2.4.3. This section gives an overview of promising methods.

First Data Flow Diagrams are explained in Subsection 2.5.1. Followed by a description of C-Wf Models in Subsection 2.5.2. Then the Design Structure Matrix is discussed in Subsection 2.5.3. Lastly the Unified Modelling Language - Activity Diagrams are mentioned in Subsection 2.5.4.

2.5.1. Data flow diagrams

Data Flow Diagrams (DFDs) are a simple technique to visualise data flow within a system. It includes data from external entities into a system, data track and data storage. Gane and Sarson (1979) introduced the method in the late 1970s, since then the approach is used by many. This subsection describes the lay-out of Data Flow Diagrams and their application.

To start with the layout of DFDs, they are built from only four symbols Ambler (2004).

1. **Squares** are used for external entities.
2. **Rounded rectangles** represent processes.
3. **Arrows** visualise data flows.
4. **Open-ended rectangles** are used for data stores.

According to Ambler (2004) consists DFD modelling procedure traditionally of five diagrams:

1. **Context DFD**
   Used to describe the context in which the system exists.
2. **Current physical DFDs**
   Used to capture the varying levels of detail modelled in the current state.
3. **Current logical DFDs**
   Used to depict the business process an independent technology point of view.
4. **Proposed logical DFDs**
   Used to depict the new design for the business process in a technology-independent manner.
5. **Proposed physical DFDs**
   Used to depict one or more potential detailed designs for the system, including technology and architecture decisions.

All five diagrams are related to each other, which makes constructing DFDs a very time-consuming process. Ambler (2004) argues that only a few project teams executed the traditional process. Besides time issues, occur model dependency issues. If one item is changed, all four other models are affected too. Ambler (2004) implies that this lead to a change in the use of DFD for business process modelling. Many contributed to research on DFDs; an overview is given of relevant DFDs applications.

Ward (1986) exposed that the interaction between timing and control aspects of a system was not taken into account. A transformation schema is proposed to solve this. This schema made it possible to *'provide a qualitative prediction rather than a quantitative one, describing the acceptance of inputs and the production of outputs by the transformations but not input and output values'* (Ward, 1986).

On the other hand, Adler (1988) states that processes for softare engineering analysis are decomposed in an ad hoc manner. Adler (1988) points at analysts using heuristics, experience and knowledge to decompose the process. Therefore, Adler (1988) proposes algebra for the formalization of the decomposition process, using a De Marco representation scheme. Disjoint input and output sets are related to specific elements of an input/output connectivity matrix (Adler, 1988).

Chang and Jimming (1991) used DFDs to address the specifications of a total service quality management model. According to Chang and Jimming (1991), the model could be used for further development of computer integrated systems. In the same year Wang and Raz (1991) combined DFDs with Analytic Hierarchy Process (AHP) for system configuration selection. In this application DFDs are used for system analysis and systems design. AHP is a support tool to determine the best system configuration for a given application.

The applicability of DFDs is improved by work of Ibrahim and Yen (2010) resulting in a tool for consistency checking. A formalisation of drawing rules and diagram definitions examined the diagrams. Besides a consistency check between the different diagrams, it is possible to draw diagrams by this tool (Ibrahim and Yen, 2010). This work contributed to the literature on DFDs as it made it possible to automate the manual process for consistency checking.

Macariola and Silva (2019) showed that *DFDs could be of great value in the improvement of data management systems for construction projects*. Extending DFDs with a simulation made it possible to reduce project duration significantly.

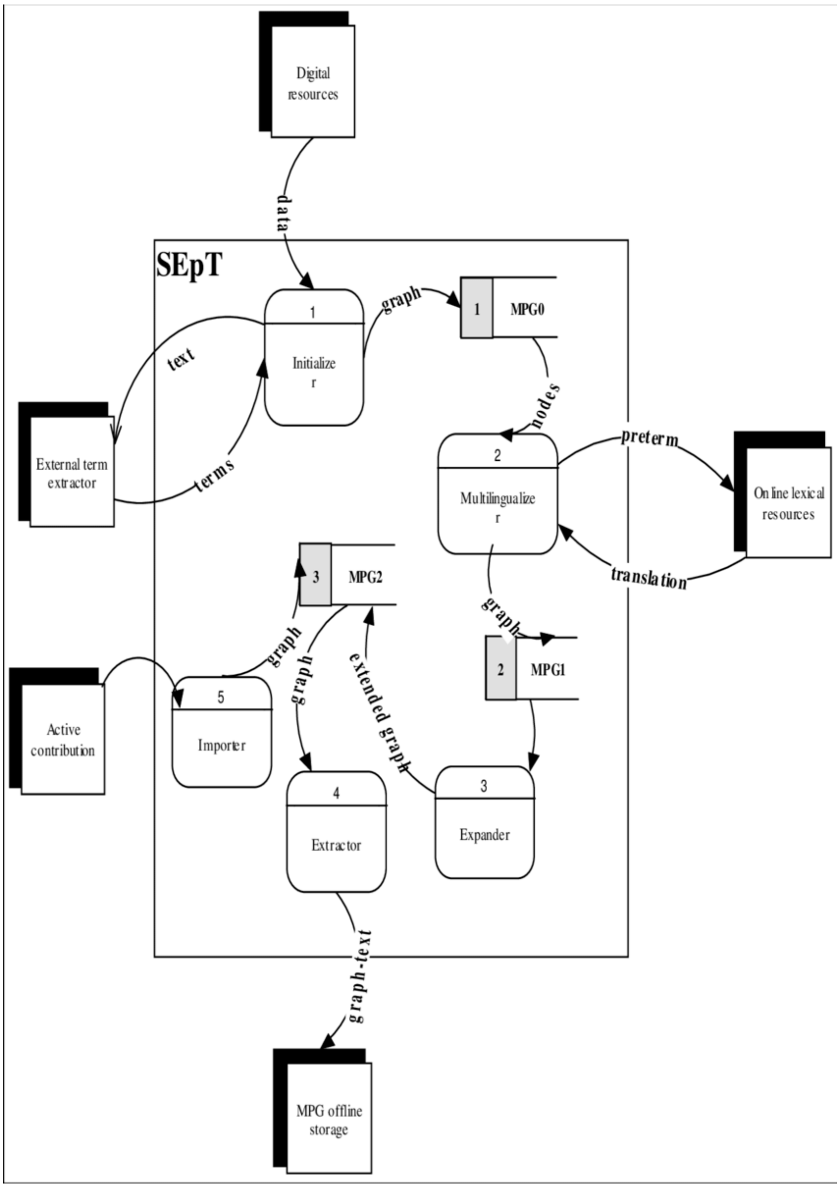


Figure 2.1: Example of the lay-out of a Data Flow Diagram (Gane and Sarson, 1979)

This reduction is found in a study developing a data flow diagram-based knowledge management system for mitigating delays in construction projects. In this research, six ongoing projects at three different companies in the Philippines are used. Macariola and Silva (2019) concludes that DFDs and other emerging technologies should be applied to construction data management to support construction delay mitigation.

To conclude, DFDs are a straightforward approach to visualise data flows in a system as the layout is easy to understand. However, due to the multiple required diagrams, it is a time-consuming approach. Over the past decades DFDs applications are studied by many. In these studies it is even called as an approach for improved construction data management.

2.5.2. C-Wf model

Bastos and Ruiz (2001) created the C-Wf model as an extension to available models in the construction industry: CIMOSA and WfMC. By introducing the C-Wf model, it becomes possible to extend CIMOSA with workflow modelling. An object-oriented approach is used to model business processes. By this approach, documentation flow can be mapped between activities. Several predefined classes are used to construct the model. The model allows different dependency types: sequential and parallel. It is also

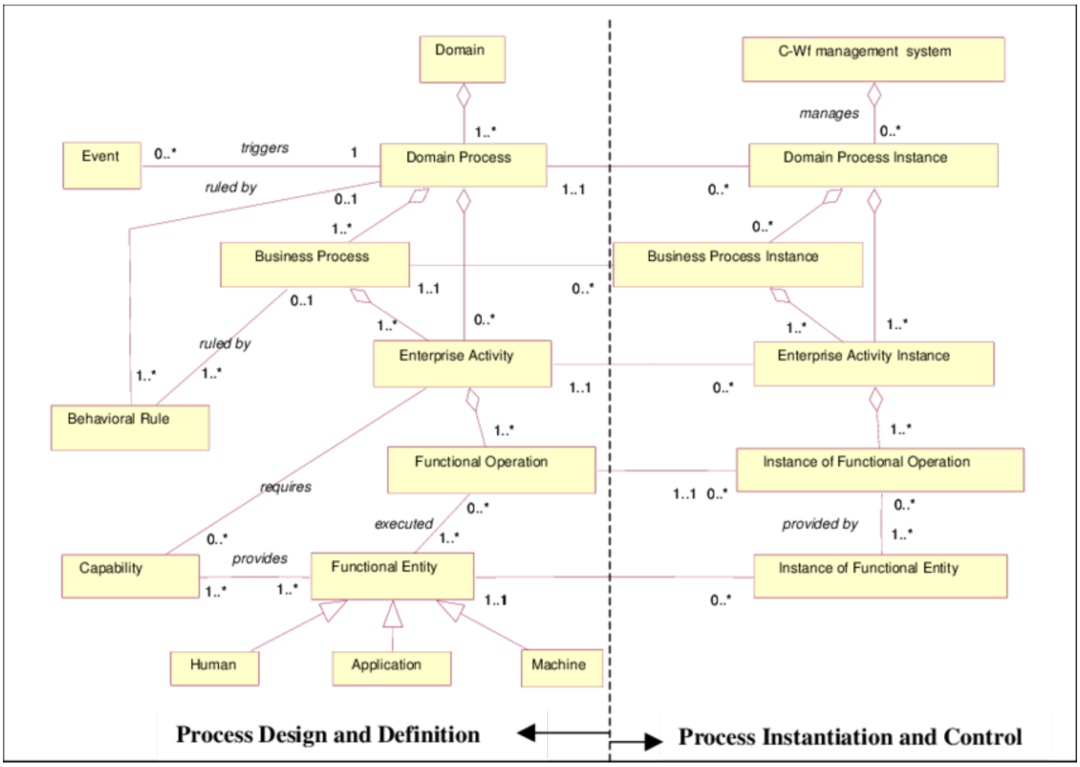


Figure 2.2: C-Wf model as defined by Bastos and Ruiz (2001)

possible to represent feedback loops in a process.

Bastos and Ruiz (2001) put particular focus on resource allocation, the C-Wf model is used to model business processes, including the planning of a production system. Constraints are required to find the optimal path, which can alternate. It is possible to use a set of constraints.

Dependencies between tasks are captured in behaviour rules of the system and are visualised in a diagram. Applications of C-Wf models are seen in process planning or multiple process planning of production systems (Bastos and Ruiz, 2002). Once processes change, the model has to change as well. A wide variety of input data is required to understand the process: resource constraints, deadlines, task durations, tasks sequence, feedback loops, triggers for activities, behaviour rules of activities.

Besides work of Bastos and Ruiz (2001) and Bastos and Ruiz (2002), research on C-Wf models is rare.

2.5.3. Design structure matrix

No difficult modelling techniques are required for the Design Structure Matrix. It is a simple combination of tasks and their dependencies an example is given by Figure 2.3. Nevertheless, it requires a notable effort to construct such a matrix. Although research on the improvement of the model build process resulted in varied techniques, still significant effort is required. A basic approach is available for the set-up of a DSM model and to conduct analysis (Eppinger and Browning, 2012). Existing out of five steps:

1. **Decompose**
   Break the system down into its constituent elements perhaps through several hierarchical levels.

2. **Identify**
   Document the relationships among the system's elements

3. **Analyse**
   Rearrange the elements and relationships to understand structural patterns and their implications for system behaviour.

4. **Display**
   Create an useful representation of the DSM model, highlighting features of particular importance or of special interest

5. **Improve**
   Most DSM applications do not only result in better understanding of the system but also improvement of the system through actions taken as a result of the DSM analysis and interpretation of its display.

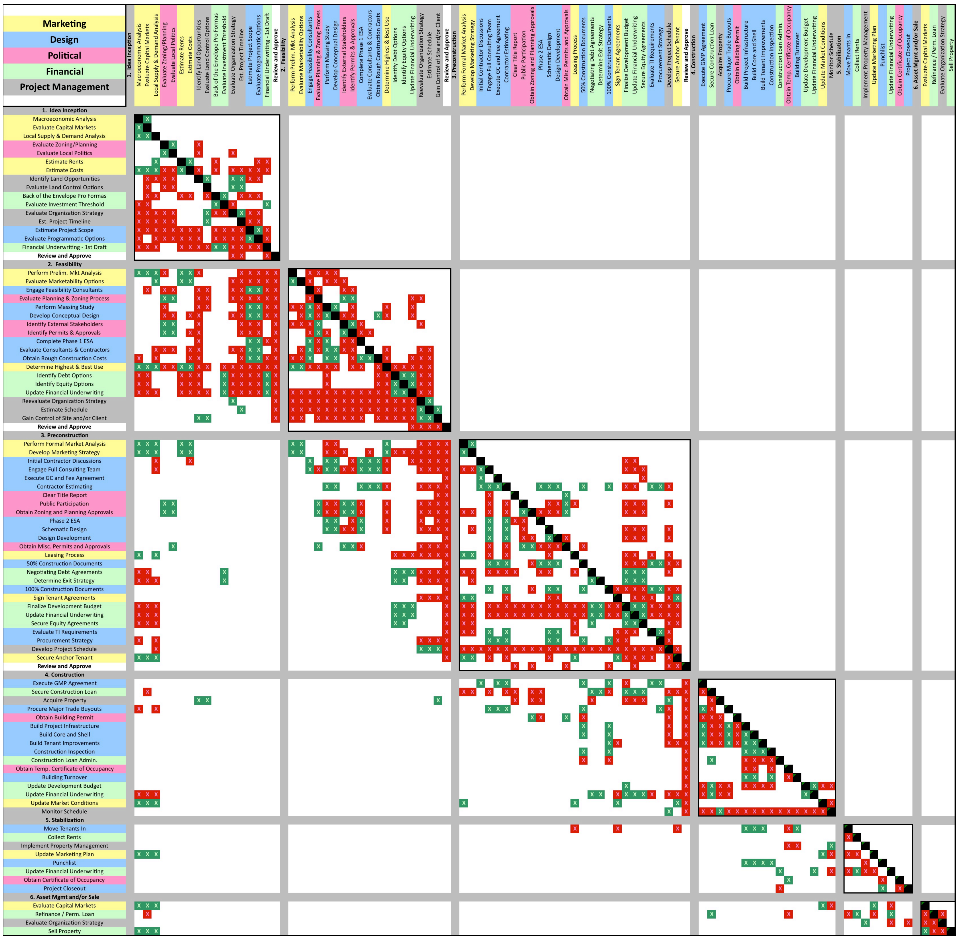


Figure 2.3: Example of a DSM structure (Eppinger and Browning, 2012). This figure serves only as an example of the model structure. Terms in rows and columns are not relevant.

According Moon et al. (2015) and Browning (2014a) it eases to construct two DSM models; one built row-by-row and one built column-by-column. With this approach, input and output perspectives are gathered separately and can be compared (Browning, 2014a, Moon et al., 2015). Hence show Chen and Huang (2007) and (Sharon et al., 2013) that it is feasible to map dependencies and activities from a product DSM to a process DSM. Hameri et al. (1999) used documentation flow data to build a process DSM. Lan et al. (2015) argued that e-mail logs are of great guidance when identifying hidden tasks and process structure. Sabbaghian (1999) developed a web-based tool to capture and visualise tasks and their interactions in large projects. Furthermore, Browning (2002) showed that it is achievable to construct a large DSM, by the integration of multiple smaller DSMs.

Before above-mentioned methods can be used, a task decomposition of the process is required (Eppinger and Browning, 2012). Subsequently, dependencies among those tasks must be identified. The task decomposition relates to the 'Decompose' step, identification of dependencies relates to the 'Identify' step. Mixed methods are used for data collection: using existing process data or models in another format, interviews and surveys (Browning, 2016). Browning (2002) argues that the use of existing process data often creates difficulties as it only involves a minimal number of activities and their dependencies. Browning (2016) relates the effectiveness of interview and surveys to the asked questions.

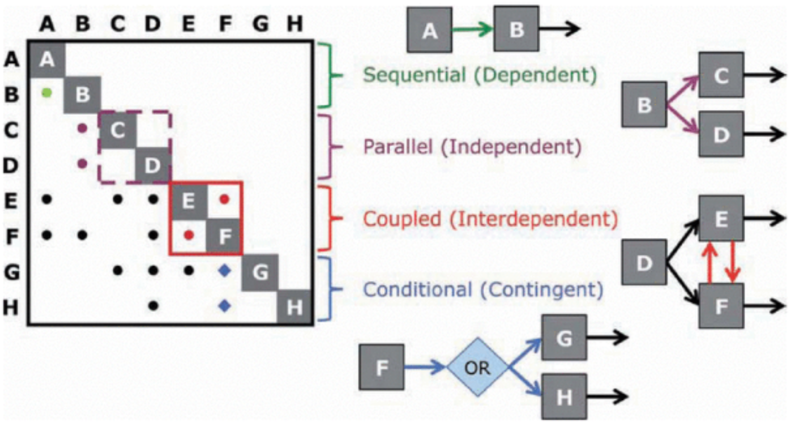


Figure 2.4: Different types of dependencies in DSM (Eppinger and Browning, 2012)

The ability to include different dependency types makes the DSM a suitable modelling method for complex projects. Figure 2.4 shows the incorporated dependency types: sequential, independent, interdependent and contingent (Browning, 2016, Browning and Eppinger, 2002, Browning, 2001, Eppinger and Browning, 2012). Another attractive feature is the possibility to add feedback loops in the documentation flow. Also, the model size is not dependent on the number of dependencies but the number of activities. This feature makes *the DSM a proper method to include extensive information flows among activities*. Browning (2016) argues that this head to a richer model of information and workflow then when flowcharts are used; *for a flowchart often 1-2 in and outputs are common, whether 3-5 is common for DSM models*. Additionally, different software tools and applications are used for: model builds up, analysis and improvement.

2.5.4. Unified modelling language - Activity diagrams

Unified Modelling Language (UML) is a widely used modelling language. Different extensions make various applications possible. In this overview UML is limited to UML Activity Diagrams (UML AD) an example is given by Figure 2.5. Ambler (2004) states that UML AD is the object-orientated equivalent of DFDs. Ozkaya and Erata (2020) observed from a survey-based study among 109 practitioners on the practical use of UML classes that UML AD is used most in data flow modelling. These results are in line with other research. Hence argues Rábová (2011) that UML AD is a suitable manner to show how workflows in business are managed. *Notably, the possibility to see how documentation moves through a business is named as the success of the UML AD* (Rábová, 2011). Dumas and ter Hofstede (2001) argue that UML AD becomes the standard in organisational process modelling. Although UML AD gains positive feedback in literature, there are still some obstacles to overcome.

Wirtz et al. (2000) identified that Unified Modelling Language and workflow management become more prevalent in practice. However, typical modelling languages for both operations can not be integrated, leading to separate modelling languages. Wirtz et al. (2000) proposed an extension to UML that makes it possible to include workflow management in the UML language to improve this situation. Integration of UML and workflow management made UML ADs a suitable tool for data flow modelling.

Dumas and ter Hofstede (2001) see potential in UML ADs but addressed that it fails to capture some useful situations. The failure is noticed in a study on expressiveness and the adequacy of activity diagrams for workflow specification.

Eshuis and Wieringa (2004) identified problems regarding verification of UML ADs. A supporting tool for verification is proposed to overcome these difficulties. This tool translates an activity diagram into an input format, which is checked by a model checker on mathematical semantics (Eshuis and Wieringa, 2004). Verification issues are the subject to more research. So pointed George and Samuel (2016) at issues regarding consistency. Inconsistency causes serious problems in later stages, especially for complex systems. George and Samuel (2016) proposed a particle swarm optimisation method to

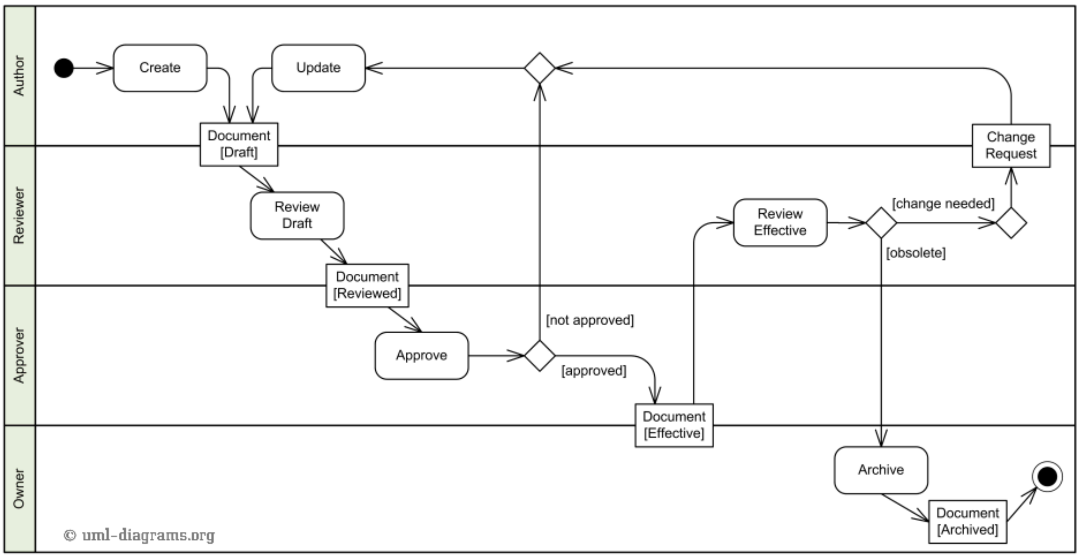


Figure 2.5: Example of an UML AD of a document management process [https://www.uml-diagrams.org/document-management-uml-activity-diagram-example.html]

detect inconsistencies and to optimise the consistency value. Application of this method will lead to valid code.

On the other hand, Beckmann et al. (2017) found that most existing models are mainly used for documentation. Current models give an improved overview of communication within a business process, compared to the situation before the model was present. However, this is often not the initial purpose of an documentation flow model. Next, Beckmann et al. (2017) identified a large number of redundant entities, which make it challenging to use the model for its initial purpose; modelling and analysing documentation flow. A tool proposed by Beckmann et al. (2017) identifies and eliminates redundant entities. Additionally, Le (2019) shows a procedure to model and verify activity diagrams by use of SPIN.

Besides research on the improvement of UML ADs, much is written about UML tools. Ozkaya (2019) conducted a literature review on publications regarding UML tools. From this review, nine tools were identified to support the simulation of UML ADs. On the other hand, is the use of UML diagrams compared to WF Petri nets by Rábová (2011). This comparison leads to valuable findings regarding UML ADs:

1. It is possible to model the entire process
2. It is possible to model responsibility of workers
3. Advanced modelling knowledge is not required
4. It is not possible to target a document
5. More then one diagram is needed
6. It is not possible to simulate or optimise processes

The last point is in contradiction to the results of Ozkaya (2019). This can be explained by the time difference between the two publications: nine years. In nine years software developments can lead to the difference.

2.6. Model selection

'All models are wrong but some are useful.'
-George E. P. Box-

Different models are explained in Section 2.5. The next step is to compare these models to criteria. Only some models are useful; thus, the aim is to find a useful model. A useful model is a model that represents reality at an appropriate level and can be used for analysis. Selection criteria for a useful model are defined by Subsection 2.6.1. Subsequently, one model is chosen from the comparison in Subsection 2.6.2.

2.6.1. Selection criteria

Reality needs to be captured in a simple but not too simple way to develop a useful model. Multiple methods are known to represent this reality. However, the purpose of the model must be clear to select a suitable model. Brooks and Tobia (1996) defines eleven elements which should be included when selecting a model. These elements can be grouped into four categories:

1. Model results;
2. Future use of the model;
3. Verification and validation; and
4. Required resources.

To start with the first category: model results. This category includes the following elements:

1. The extent to which the model output describes the behaviour of interest
2. The possibilities for analysis
3. The accuracy of the model's result
4. The ease which the model and its results can be understood

Table 2.3: Criteria for documentation flow model selection

Selection Criteria		
1. Model Results	*1.1 Representation of reality*	1.1.1 Flexibility level customisation
		1.1.2 Different dependency types
		1.1.3 Uncertainty in task duration
		1.1.4 Large amount of information
		1.1.5 Feedback loops in information flow
		1.1.6 Different production methods of a yard
	1.2 possibilities for analysis	1.2.1 Identification of redundant information flows
		1.2.2. Identify the affect success
		1.2.3 Address directions for improvements
	1.3 Accuracy of model results	
	1.4 Understandability of model and its results	
2. Future use	2.1 Ease to combine model with other models / programs	
3. Verification and Validation	3.1 Successful applications in literature	
4. Resource Requirements	4.1 Time and Cost to collect data	
	4.2 Time and Cost to construct the model	
	4.3 Time and Cost to run the model	
	4.4 Time and Cost to analyse results	
	4.5 Hardware requirements	

Within the context of this project, the behaviour of interest is project execution in engineer-to-order shipbuilding projects. The purpose of the model is to describe documentation flow, so this can be analysed in a later stage. The characteristics and challenges of engineer-to-order shipbuilding projects were described in Subsections 1.2.2 and 1.2.3 and will not be handled in detail here. From Subsections 1.2.2 and 1.2.3 is concluded that a simulation model for documentation flow in such projects should be able to deal with:

1. The extent to which the model output describes the behaviour of interest

   (a) Flexibility in the level of customization of different parts
   (b) Different types of dependencies between activities, including overlapping
   (c) Uncertainty in task duration
   (d) Large amount of information generation and manipulation by different working disciplines
   (e) Feedback loops in the information flows
   (f) Different production methods of a yard

   Besides the representation of the project environment, it must be possible to analyse the documentation flow. Therefore, an additional element is added to the first category: possible analysis. An appropriate model is able to analyse the following aspects of the documentation flow:

2. The possibilities for analysis

   (a) Identification of redundant documentation flows
   (b) Identify the affect of documentation flow on project success
   (c) Address directions for documentation flow improvements

Brooks and Tobia (1996) claims that also future use of a model must be included in selection. The ease to combine a model with existing models and portability are seen as essential elements. For this project; it must be able to use the model in combination with other programs. Next, a strategy for verification and validation should be available for the selected model. Previous work on the successful application of a method helps to determine such a strategy.

Furthermore argues Brooks and Tobia (1996) to include required resources in the selection process. To do so, Brooks and Tobia (1996) proposed the following elements:

1. The time and cost to build the model (including data collection, verification and validation)

2. The time and cost to run the model

3. The time and cost to analyse the results of the model

4. The hardware requirements of running the model.

For the selection process of this project, it is desired to make a distinction between resource requirements for data collection and model generation, including verification and validation. As it is expected, that available time for data collection and model creation is limited by different constraints.

To conclude the different models as described by 2.5 are compared to the criteria as displayed by Table 2.3.

2.6.2. Model comparison

The models mentioned in Section 2.5 are compared to the criteria displayed in Table 2.3 by decision tables. The models get a score based on the criteria; this results in an overall ranking. All criteria have the same weight in this comparison. All scores are summed; the method with the highest score performs best according to the criteria. Selection criteria are split-up in two groups: 1) criteria on the representation of the documentation flow and possible analyses and 2) criteria on the use of a model. By this manner, separate scores are established for the representation of reality and use of the model. In this comparison,

- -2 stands for very poor;
- -1 stands for poor;
- 0 stands for fair;
- 1 stands for good;
- 2 stands for excellent.

Table 2.4: Decision table ability to represent project environment and available analysis

Selection Criteria			**DFDs**	**C-Wf**	**DSM**	**UML AD**
1	1.1	1.1.1	-1	0	0	0
		1.1.2	0	0	2	0
		1.1.3	0	1	1	0
		1.1.4	0	1	2	1
		1.1.5	1	1	2	1
		1.1.6	-2	0	0	0
	1.2	1.2.1	2	0	2	2
		1.2.2	2	0	2	1
		1.2.3	1	-1	1	1
	1.3		0	1	1	1
	1.4		0	0	1	1
3	3.1		1	-2	2	2
Total Sum			5	1	16	10
Rank			3	4	1	2

Scores are based on findings in the literature. Score determination is explained per criteria.

To start with the criteria on the representation and analysis, as given by Table 2.4. It can be seen that all have some problems with representing different levels of customisation (criteria 1.1.1). DFDs received the lowest score (poor), scores for C-Wf, DSM and UML AD are equal (fair). The same is seen for the ability of the models to capture different production methods (criteria 1.1.6). Again, DFDs received the lowest score (poor), scores for C-Wf, DSM and UML AD are equal (fair). The difference between DFDs and other methods is caused by the fact that DFDs consist out of multiple diagrams that require consistency. All models face difficulties with incorporating flexibility since it is included as a decision. Decision options are not flexible but fixed. It means that flexibility is limited to the options captured by the model.

Concurrency requires a variety in dependency types (criteria 1.1.2). DFDs, C-Wf and UML AD models get the same score, fair. Because these models use in basis only parallel and non-parallel tasks, however, it should be possible to manipulate this slightly by changes in behaviour rules of the models. So argues Russell et al. (2006) that this should be possible for UML AD models, although he points at potential issues for highly concurrent business processes. The DSM model gets the highest score, excellent. This is based on the wide variety of dependencies as named by Browning and Eppinger (2002), Eppinger and Browning (2012) and Cho (2001).

An uncertain initial task duration marks the complexity of engineer-to-order project (criteria 1.1.3). Different models are able to include uncertainty, although different scores are given. The ability to cover uncertainty in DFDs is ranked fair. From literature is found that DFDs can manage uncertainties in data, but uncertainties in executing strategy are not explicit named (Durugbo et al., 2010). Nevertheless, it is expected that this should be possible to incorporate as it is possible to include uncertainty in input data. Similar applies to UML AD models, Ma et al. (2011) introduced fuzzy information modelling to UML AD models. Additionally, uncertainty is involved in the data flow and not in the execution of different processes. On the other side, Bastos and Ruiz (2001) describes uncertainty in task duration by an average, minimum and maximum value for the duration. Using three values is comparable to the approach used by Browning and Eppinger (2002) for analyses of a DSM structure.

Not all models score good on the ability to handle a large amount of information (criteria 1.1.4). Browning (2016) argues that the DSM model is especially suitable for large amounts of information. Therefore, gets the DSM model an excellent score. C-Wf and UML AD score good, they can handle a large number of information flows but not as good as a DSM model. The fact that C-Wf and UML AD models come with modelling of the business process by business rules for creation and manipulation of information flow affects this (Ambler, 2004, Bastos and Ruiz, 2001). It should be noted that work to improve consistency for UML AD models is included, for example, the possibility to use particle swarm optimization to check for consistency among different diagrams (George and Samuel, 2016). DFDs get a fair score as a result of less research on consistency models and the need to model all actions for data creation and manipulation in business rules. This score is supported by Jilani et al. (2008), who argues that DFDs lack formalization, which leads to inconsistency problems. Not only the amount of information is essential, the model should also be suitable to deal with feedbacks in information flow (criteria 1.1.5). The DSM method gets the highest score as it can deal with both feedback and feed-forward loops (Eppinger and Browning, 2012). Other methods are bounded to feedback loops alone (Ambler, 2004, Bastos and Ruiz, 2001).

From the criteria for possible analysis (criteria 1.2.1-1.2.3) is made clear that the models are developed for various applications. So scores C-Wf worst on these criteria, which is related to the purpose of this model: workflow management with a focus on resource allocation (Bastos and Ruiz, 2001). UML AD scores higher then C-Wf, but does not get the highest score. It is concluded from the work of Russell et al. (2006) that UML AD models are less suitable for the analysis of the effect of the data flow on project success. DFDs and DSM get the highest score among all. Durugbo et al. (2010) and Macariola and Silva (2019) showed that DFDs can be used to analyse the efficiency of data flow and to identify areas for improvement. Additionally, Gelbard et al. (2002) claims that DFDs can be used for project management purposes. On the other hand alike conclusions are made for DSM models by Moon et al. (2015), Browning (2014b), Eppinger and Browning (2012) and Browning and Eppinger (2005).

Accuracy of the different models (criteria 1.3) is related to the quality of input data (Yassine et al., 2001). However, not all methods receive the same score. DFDs score lower as it is expected that models are less accurate when formal structure is lacking (Ibrahim and Yen, 2010, Jilani et al., 2008, Sun et al., 2006).

Understandability (criteria 1.4) of almost all methods receives the score good. Only the C-Wf model scores fair, as some knowledge is needed to understand this model (Bastos and Ruiz, 2001). DFDs and UML AD are ranked good, as they are based on a flow chart layout which is familiar to many. However, it makes it challenging to visualise dependencies among activities in a structured manner. DSM is ranked good, due it abilities to present high-level dependencies in a structured manner (Browning, 2016). Though are DSM models less familiar to people, so it may take a more time before the model is understood. It should be noted that a DSM model was previously used at Damen (van Boekel et al., 2020). However, this model had a focus on design instead of processes. Thus no notable challenges are seen in the use of a DSM model at Damen.

Quite a difference is seen in successful applications of the models in the literature (criteria 3.1). Where literature about C-Wf models is rare, DSM and UML AD are frequently named. DFDs is also named often, but applications on documentation flow analysis are limited.

From the criteria named above is concluded that both DSM and UML AD are good candidates for representing the documentation flow of engineer-to-order shipbuilding projects. Difficulties with highly concurrent processes are a notable drawback of the UML AD model. Selection is not done on representation and analysis scores alone, resource requirements play also an important role. For now, the DSM looks like the best alternative. However, all models are compared on resource requirements to ensure that the DSM is a good pick. This judgment is also done by a multi-criteria analysis, as shown in Table 2.5.

Table 2.5: Decision table resource requirements

Selection Criteria		**DFDs**	**C-Wf**	**DSM**	**UML AD**
2	2.1	1	2	1	2
4	4.1	-1	-2	-2	-2
	4.2	-2	-2	-1	-2
	4.3	2	2	2	2
	4.4	-1	0	1	0
	4.5	2	2	2	2
Total Sum		1	2	3	2
Rank		4	2	1	2

The first criteria (2.1) is linked to the ease models can be joined to other models. Such as used models at the company. C-Wf and UML AD are ranked highest since UML language can be combined with a large number of applications. Russell et al. (2006) stresses this by *'UML is posited as the 'Swiss army knife' for systems modelling and design activities'*. As C-Wf is based on UML language, this gets the same score. Jilani et al. (2011), Tran et al. (2004) and Meng et al. (2010) showed that DFDs can be transferred to UML language, resulting in a good score for DFDs. DSM method also gets the score good, as no complex programs are used for model creation. It is also possible to transfer outcomes to other formats.

Almost all models score very poor on data collection (criteria 4.1). This score is mainly caused due to the required time. For all methods, data collection is a time-consuming procedure (Akcay et al., 2017, Ambler, 2004, Bastos and Ruiz, 2002, Browning, 1998). Global procedures are similar; in extensive interviews, surveys, present process documentation and observations are used to collect data (Akcay et al., 2017, Ambler, 2004, Browning, 2016). It should be noted that the duration of the data collection process is affected by more than a chosen model. Akkermans and Bosker (1994) shared learned lessons after an unsuccessful Case Study at a large multinational; one of the failure causes was the unwillingness of people to collaborate during interviews and workshops. Thus, people must be willing to participate in the research.

Scores differ for model construction (criteria 4.2), as the global process is not similar. DFDs scores very poorly due to to the five models and their consistency requirement (Ambler, 2004). DSM scores poor due to the timely process of entering a large number of tasks and defining different types of dependencies between them (Browning, 2016). C-Wf and UML AD score poor since the process has to be modelled in terms of behaviour rules; a time-consuming process (Ambler, 2004, Bastos and Ruiz, 2001). The construction of C-Wf and UML AD models requires less time, as multiple tools are available as support during the determination of behaviour rules (Bastos and Ruiz, 2001, Beckmann et al., 2017, George and Samuel, 2016, Le, 2019). The value of support tools is recognised in DSM literature but comes with expensive software (Browning, 2016).

No risks are seen in the required time or cost to run the different models (criteria 4.3). On the other hand, it is expected to see differences in the required time for analysis (criteria 4.4). Based on applications of different models in the literature is expected that DSM comes with the most suitable analyses techniques taking time into account (Browning, 1998, Browning and Eppinger, 2002, Cho, 2001, Meier, 2011, Meier et al., 2007). To emphasis that DSM models do not grow with increasing dependencies but with increasing tasks (Steward, 1981). Techniques for analysis use process architecture in a matrix form as an input (Browning, 1998, Browning and Eppinger, 2002, Yassine et al., 2001). This manner implies that the number of dependencies does not affect the required time to analyse the system (Eppinger and Browning, 2012). UML AD and C-Wf score fair as they do depend on the number of dependencies (Bastos and Ruiz, 2001, Obaid et al., 2014). DFDs score poor due to the multiple diagrams (Ambler, 2004). To end are no risks seen on hardware resources for the different methods (criteria 4.5) as no large computational power is needed (Ambler, 2004, Eppinger and Browning, 2012).

To conclude, the DSM model is selected as a suitable method to model documentation flow in engineer-to-order projects. This is mainly based on its ability to include different types of dependencies, deal with large amounts of information and opportunities for analysis. Also, the DSM model is broadly applied in industrial settings and described in many publications. Furthermore, no sophisticated software is required, and the DSM method is used in earlier research at Damen (van Boekel et al., 2020). Challenges are identified regarding the required time for data collection and model construction. Still, all other methods scored also poor on these criteria. Risks with data collection and model construction are acknowledged, but it is expected that this will not make it impossible to use DSM for this application.

2.7. Model application

Multiple simulations are proposed over the past years to forecast the effects of rework on duration and cost. Various analytical models are developed to predict project duration, assuming sequential activities (Smith and Eppinger, 1997a), parallel activities (Smith and Eppinger, 1997b) or a combination (Smith and Eppinger, 1998). The before mentioned models can not include general cases. Browning (1998) introduced a DSM-based discrete-event Monte Carlo simulation by Latin Hypercube sampling to predict project duration, cost, risk and their variations. The simulation analyses the effect of alternative process architectures on project outcome (Browning, 1998). It includes rework probability, rework risk, learning curves and work policy. Yassine et al. (2001) created a framework to determine rework probabilities. The framework is applied on a study streamlining an automotive hood system development process (Zambito, 2000). Yassine et al. (2001) uses *'in total 15 personal and/or phone interviews with 10 stakeholders with a duration varied from 45 to 90 minutes'* for data collection.

After the introduction of a DSM-based discrete-event Monte Carlo simulation (Browning, 1998),

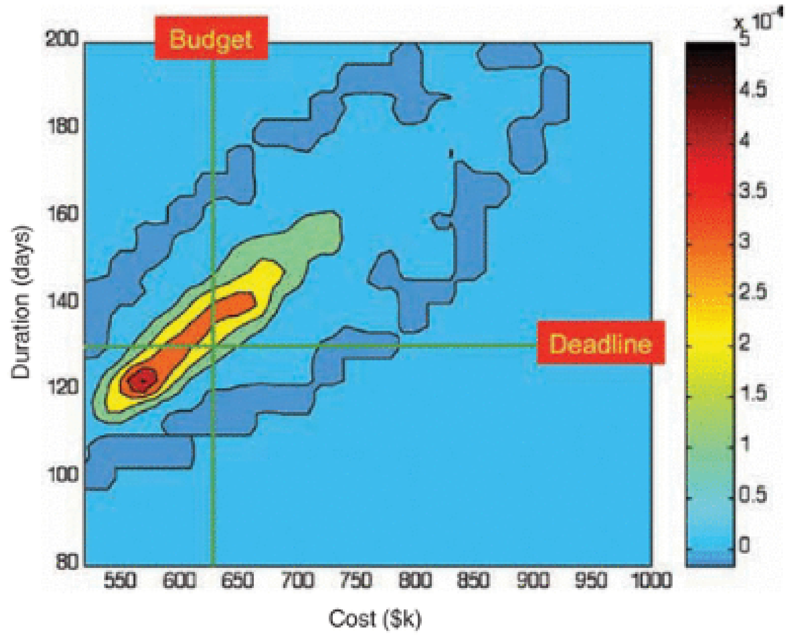


Figure 2.6: Example of simulation outcome (Browning and Eppinger, 2002)

more research is carried out. Others proposed extensions in terms of resource constraints (Browning and Eppinger, 2005), technical performance characteristics (Levardy and Browning, 2009) or by introducing other objectives like process robustness (Yassine, 2007). A discrete-event Monte Carlo simulation is used by many (Adler et al., 1995, Baldwin et al., 1999, Cho and Eppinger, 2001, Galvez and Ordieres, 2015, Lukas et al., 2007, Zhao et al., 2008). Some (Browning, 1998, Browning and Eppinger, 2002, Cho and Eppinger, 2001, Lukas et al., 2007) use Latin Hypercube sampling techniques to implement the Monte Carlo simulation. Length and computational costs are a limiting factor for Monte Carlo simulations (Janssen, 2013). For convergence, a massive amount of runs is required Seaholm et al. (1988). Janssen (2013) claims that Latin Hypercube sampling is more efficient than Monte Carlo sampling. This outcome is in line with earlier findings of Saliby and Pacheco (2002). Li et al. (2018) also states that Latin Hypercube sampling has better efficiency than Monte Carlo sampling. Seaholm et al. (1988) identifies a significant reduction in required runs by Latin Hypercube sampling. Janssen (2013) observes that Latin Hypercube sampling reach the same accuracy as general Monte Carlo sampling. Thus Latin Hypercube sampling can be used to reduce required simulation runs (Browning, 1998).

Nevertheless, some explore the use of process DSM with other models then Monte Carlo simulations; Markov Chains (Carrascosa et al., 1998) and Petri-nets (Yan et al., 2003). Browning (2016) place this in perspective by *'none of these modelling views alone is sufficient to capture a rich process model in its entirely'*. Despite some use of other simulation models, a consensus is found on the use of discrete-event simulations for this application.

Others used the eigenstructure of the process DSM matrix to analyse a particular case of parallel iteration (Bashir et al., 2009, Smith and Eppinger, 1997a, Yassine and Braha, 2003). By this method, coupled blocks are executed simultaneously, whereafter information is exchanged. This approach leads to a rework model based on the DSM interaction values (Smith and Eppinger, 1997a). By this model, it becomes possible to analyse the DSM as a work transformation matrix. Yassine and Braha (2003) used this method to introduce phenomena as design convergence and design churn, where iterations may continue when they add value to the entire process. Bashir et al. (2009) use the eigenstructure for the assessment of decomposability of interdependent design project tasks.

On the other hand, meta-heuristics are developed for the analysis of process DSMs. So are enhanced genetic algorithms used by Meier et al. (2007) to sequence binary DSMs. Some researches use Evolutionary Algorithms to find some kind of optimisation (Meier, 2011, Wang et al., 2013, Whitfield et al., 2003). Meier (2011) and Wang et al. (2013) use multi-objective genetic algorithms to analyse time-cost optimisations and trade-offs in processes. Others used simulated annealing or genetic algorithms to manipulate process architecture in order to minimise project duration (Lancaster and Cheng, 2008, Sered and Reich, 2006, Whitfield et al., 2003).

For short advanced analysis techniques make it possible to analyse a DSM structure in detail. Many features are included in the analysis, and applications are convenient. So are time-cost trade-offs made (Meier, 2011), the probabilistic outcome of a project calculated (Browning and Eppinger, 2002) and is it possible to display results on a Ghantt Chart (Browning, 1998). Besides, Table 2.6 demonstrates that developed methods are used by others.

Table 2.6: Numbers of citations as defined by Google Scholar

Publication	**Cited**
Browning and Eppinger (2002)	609
Browning (1998)	188
Browning and Eppinger (2005)	389
Cho and Eppinger (2001)	150
Yassine et al. (2001)	69
Yassine (2007)	67

2.8. Chapter summary

This chapter summary gives an overview of the achieved aims of this chapter as mentioned at the beginning. Research questions are briefly answered, and a plan for the next chapters is proposed.

1. *How does project documentation quality affect project success?*
   Documentation quality is related to rework, which harms success. A link between the causes of rework and quality attributes for documentation quality is given by Table 2.1. The relative importance of quality attributes is partly related to the context. Table 2.2 provides quality attributes affecting success in different studies. Accuracy is mentioned most frequently. Influencing quality attribute factors are context specific.

2. *Which current solutions strive to reduce the negative effect of documentation flow on success?*
   Multiple solutions are present to deal with adverse effects on success. However, the problem is still present. Different industries used different approaches, which creates valuable insight. In the end, all strive to process integration, which requires process understanding. Information flow modelling has a crucial role in process understanding. Different studies showed that process improvements failed, through lack of process understanding. Thus, documentation flow modelling is crucial for process improvement.

3. *How can a model describe the project environment of engineer-to-order projects?*
   Four different models are discussed for documentation flow modelling:

   (a) Data Flow Diagram
   (b) C-Wf Model
   (c) Design Structure Matrix
   (d) Unified Modelling Language - Activity Diagrams

   The models are compared to several criteria, as is shown in Tables 2.4 and 2.5, and the **Design Structure Matrix** performed best according to the selected criteria.

4. *Which methods/tools are used by others to analyse documentation flow?*

   - **Advanced techniques for DSM analysis**
     Discrete-event Monte Carlo simulations are often used. Latin Hypercube Sampling is used to limit computational effort. Also, matrix characteristics as eigenvalues are used for analysis. Meta-heuristics are also used to find some sort of optimisation.

5. *Plan for coming chapters*
   The literature study forms the basis of this master thesis. Results give direction for the following chapters. The content is described for each chapter.

   (a) **Chapter 3**
   Based on the literature, a DSM-based discrete-event Monte Carlo simulation is developed. Existing documents and survey-based interviews are used to collect data. This approach is line with the approaches of Yassine et al. (2001) and Browning (1998).

   (b) **Chapter 4**
   The DSM-based discrete-event Monte Carlo simulation is used to determine process improvement by an alternative documentation sharing strategy. First parameter scaling is required. The simulation of different scenarios ranks quality attributes. The results are the basis for the proposed alternative strategy.

   (c) **Chapter 5**
   Conclusions are made, and recommendations for further research are stated.

3

A DSM-Based Documentation Flow Simulation

In the previous chapter, the Design Structure Matrix was chosen to model documentation flow of engineer-to-order shipbuilding projects. Furthermore, techniques for analysis were discussed. Potential is seen in the use of a discrete-event Monte Carlo simulation. This chapter aims to set-up a non-project-specific Design Structure Matrix for DTC projects and to develop a DSM-based discrete-event Monte Carlo simulation in MATLAB. The simulation is used in the next chapter to rank quality attributes and to test alternative documentation sharing strategies.

Two main tasks must be completed to construct a model to find answers to the research questions. Defining a non-project-specific DSM model and the development of a DSM-based discrete-event Monte Carlo simulation. These tasks are executed in parallel. Model requirements are needed to model the project environment. Thus first requirements are presented in Section 3.1. Then simulation input and data collection are addressed in Sections 3.2 and 3.3 respectively. Once input data is collected, the development of the simulation is discussed in Section 3.4. The simulation outcome is explained in Section 3.5, followed by verification and validation in Section 3.6. The chapter ends with a chapter summary in Section 3.7, stating relevant results used in the next chapters.

3.1. Simulation requirements

Based on the literature study it is decided to use a DSM-based discrete-event Monte Carlo simulation to find answers to the research questions. However, the DSM-based discrete-event Monte Carlo simulation is not an answer to itself. The simulation must represent the environment of interest to be useful. To ensure the development of a valuable model, requirements on process behaviour are defined.

The DSM-based discrete-event Monte Carlo simulation:

1. Must be useable for different engineer-to-order shipbuilding projects at DTC.

   This includes:

   (a) Different designs
   (b) Different building yards

2. Must represent documentation flow (in the form of documentation) between Engineering, Material Coordination, Purchase and a building yard.

   This documentation flow is must include:

   (a) Feedback and feed forward loops.

   (b) Uncertainty in exact information flow

3. Must be able to represent the characteristics of a engineering-to-order project.

   In particular:

   (a) Uncertainty in initial task duration and cost.
   (b) Large (inter)dependency among tasks
   (c) Overlapping activities (concurrent)

4. Must be able to give insight in the effect of rework caused by the documentation sharing strategy.

   (a) Highlight areas that have large impact on project outcome.
   (b) Include different causes of rework through documentation quality attributes.
   (c) Quantify the impact of areas on project outcome.
   (d) Display results in an easy to interpreted manner.

5. Must be able to adjust documentation sharing strategy based on gained insight in current situation.

3.2. Simulation input

To assure a simulation model according to the requirements defined in Section 3.1, specific input data is required. The dependency structure is captured in a Binary DSM and explained in Subsection 3.2.1. The Binary DSM is a crucial input for the simulation, but not the only essential input. Rework probabilities and rework impact explained in Subsections 3.2.2 and 3.2.3 respectively are also essential. Further, activity sequence and initial duration and cost are required. Subsection 3.2.4 deals with activity sequence and Subsection 3.2.5 with the initial data.

3.2.1. Binary DSM

The binary DSM is an elemental DSM model; the strength of dependencies is not covered. Just the presence of dependencies is presented. A binary DSM model gives the opportunity to include sequential (dependant), parallel (independent) and coupled (interdependent) tasks (Eppinger and Browning, 2012).

The model is constructed using the IF/FAD notation, since this notation is used most frequently (Browning, 2016). This notation puts activity input in rows and output in columns. Another notation that could have been used is IC/FAD, placing input values in columns and output values in rows. According Eppinger and Browning (2012) is no absolute standard present.

Figure 3.1 outlines the potential layout of a binary DSM for a DTC-project. The difference in size between the *DTC DSM* and *YARD DSM* is not related to the number of tasks. However, it shows the possibility of combining two non-squared matrices. The green and grey block represents sharing information from DTC to the yard or the other way around. Dependencies are represented by a ’1’ when no dependencies are present a ’0’ is used (not displayed in Figure 3.1).

3.2.2. Rework probabilities

Rework probability, *the likelihood an activity has rework due to a typical change in its input*, is defined by Equation 3.1 (Browning and Eppinger, 2002). Further is rework affected by the likelihood for input changes and the probability that such changes affect the activity. Equation 3.2 describes this relationship (Browning and Eppinger, 2002).

$$\text{P(rework for activity i caused by a typical change in its input from activity j)} = \boldsymbol{DSM}_{ij1} \tag{3.1}$$

$$\begin{aligned}\text{P(rework for an activity caused by change in one of its inputs)} = \\ \text{P(change in the input)} \cdot \text{P(the change affecting the activity)}\end{aligned} \tag{3.2}$$

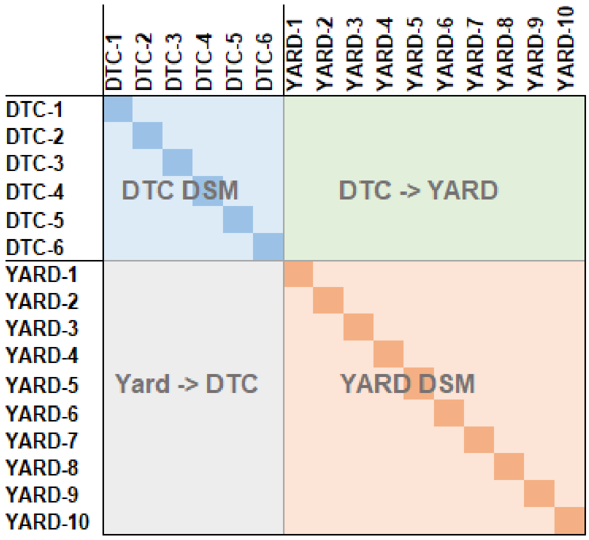


Figure 3.1: Global structure of the Design Structure Matrix, self-illustration

3.2.3. Rework impact

The consequence of rework can differ from task to task. Robust activities experience less consequences of changes, then sensitive activities. Rework impact is defined as *'the percentage of the activity that must be reworked'*, Equation 3.3, (Browning and Eppinger, 2002). Each dependency link among activities has its rework impact value.

$$\%(\text{rework for activity i caused by a typical change in input from activity j}) = \boldsymbol{DSM_{ij2}} \quad (3.3)$$

3.2.4. Activity sequence

The binary DSM, rework probability and rework impact matrices capture the dependency structure and relation to rework. The activity sequence vector, **V**, defines the *executing order of activities*. Changes to the activity sequence vector adjust process architecture.

3.2.5. Activity duration and cost

Initial duration and cost for individual tasks are needed to predict total project duration and cost. Three values are required to include uncertainty: *Best Case Value (BCV)*, *Most Likely Value (MLV)*, and *Worst-Case Value (WCV)* (Browning and Eppinger, 2002). Uncertainty in initial duration and cost characterises engineer-to-order projects (Mello et al., 2017, Mello and Strandhagen, 2011, Mello et al., 2015a,b). Thus including this is crucial to create a useful simulation.

Activity duration and cost are project-specific, so this is discussed in more detail in chapter 4. However, the determination of **MLV**, **BCV** and **WCV** is independent of the project. Planned duration and cost are seen as **MLVs**. Thus *it is assumed that project schedules and budgets give reasonable values*. Also, it is assumed that **BCV** and **WCV** can be approximated by Equations 3.4 and 3.5. The impact of this assumption is tested for the case study in Chapter 4. Definitions are equal for duration and cost. A larger overrun then underrun is used as it is more likely that an activity has a delay.

$$\boldsymbol{BCV} = 0.9 * \boldsymbol{MLV} \quad (3.4)$$

$$\boldsymbol{WCV} = 1.2 * \boldsymbol{MLV} \quad (3.5)$$

Duration and cost are sampled from a triangular distribution based on the three values. Figure 3.2 visualizes this probability distribution. It should be noted that there is more chance to have a value between MLV and WCV then between BCV and MLV. By including this in the DSM, *the attempt to relax when work is ahead of schedule is taken into account* (Browning, 1998, Browning and Eppinger, 2002).

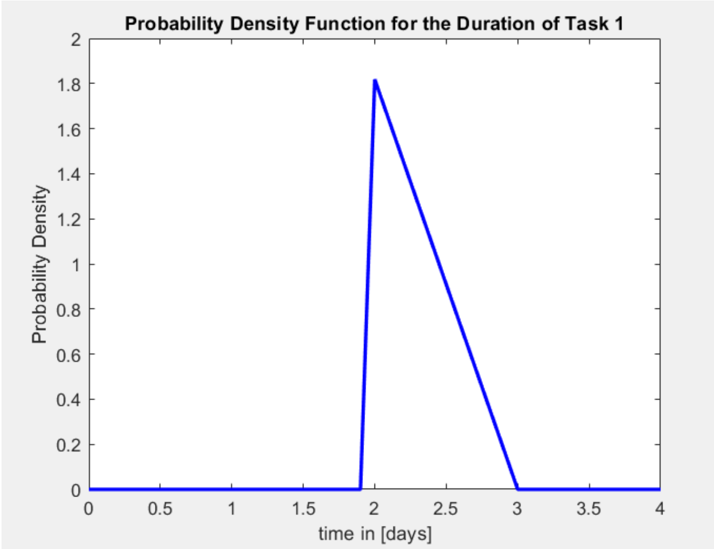


Figure 3.2: Probability distribution activity duration, self-illustration

3.3. General data collection

Company documentation, interviews, and surveys are used to collect simulation input data. Browning (2016) argues that a combination of company documentation and interviews can increase effectiveness. However, effectiveness is also affected by the way questions are asked and what participants are selected. Akkermans and Bosker (1994) demonstrate the effect of selecting uncooperative participants on the data collection process. This outcome underlines the importance of participant selection.

Participants are selected based on their knowledge of the subprocesses and willingness to participate in this research. It resulted in 10 participants; an overview of the participants can be found in Table 3.1.

Table 3.1: Participants for interviews for each department

Department	**Function**
Project Management	Design and Proposal Engineer DTC, former Project Manager DTC
Engineering	Manager Engineering DTC, Project Manager Engineering
Material Coordination	Logistics Process Manager, Team Leader Material Coordination, Material Coordination Support
Procurement	Coordinator Project Procurement
Production	Site Managers Vietnam, Israel and South Africa

The data collection process of the non-project-specific model input is discussed in this section. First, the data collection process for the Binary DSM matrix is described in Subsection 3.3.1. Followed by the data collection processes to construct the rework probability matrix in Subsection 3.3.2 and the rework impact matrix in Subsection 3.3.2. Activity durations and cost are project-specific and not included in this section. The collection of project-specific data is not included in this section, but is described as part of the case study presented Chapter 4.

3.3.1. Binary DSM

A task decomposition and identified dependencies are required to construct the Binary DSM. As DTC executes different projects at multiple yards, a general decomposition is desired. Although organisational differences are present at different yards, all yards build vessels. Thus it is expected to find a general decomposition for projects within the scope of this master thesis. The development of the Binary DSM for DTC-projects is a time-consuming and iterative process. On one side, it is a challenging task due to perspective differences. On the other side, it is challenging because a balance between available data and adequate detail to gain valuable results must be obtained.

First, an initial high-level task decomposition is made from building schedules, project plans and other process documentation. In this high-level decomposition, the process is decomposed into departments, shown by Figure 3.3.

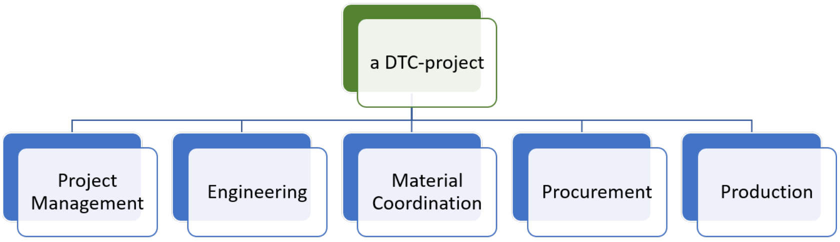


Figure 3.3: Decomposition based on departments

The next step is to define tasks and dependencies at a more detailed level through interviews. Interviews are based on shared question forms, enclosed in Appendix A.1, to keep the conversations structured and efficient. Questions are divided into three groups:

1. General questions;
2. Decomposition questions; and
3. Dependency questions.

Participants are asked to answer and return the general questions before the interview. By this, participants get an impression of the type of questions. Decomposition and dependency questions are related. Process activities must be clear to identify dependencies.

Interviews are performed using Microsoft Teams and have an approximated duration of 90 minutes. Sometimes more time is needed, and a new moment is scheduled. *Experience learned that long (>90 min) Microsoft Teams meetings are less effective than two shorter (<90 min) meetings*. During the interviews answers to general questions are discussed, remaining questions are handled, and notes are taken of relevant relating issues.

After each interview, results are used to adjust the initial decomposition. In case of minor adjustments, decomposition and dependencies are discussed in the same interview. If major adjustments are needed, dependencies are discussed in another interview. This protocol gives time to adjust the decomposition carefully and to discuss the alternative decomposition. Finally, a DSM structure for each department is created. Links with other departments or the yard are identified.

Eventually, all department DSMs are collected, and a final binary DSM structure is created. Tasks of different departments and the yard are combined. The concluding DSM structure does not make a difference between batch sizes, amount of sections, blocks or zones. Batches are defined as groups of engineering drawings that are shared with different parties. Zones are the smallest space in a vessel, often used for outfitting. Sections are larger and multiple sections for a block. Block are used for the hull assembly. In this way, the DSM structure is modular. An algorithm in Subsection 3.4.1 transforms the modular DSM matrix into a project-specific matrix. *It should be noted that the general binary DSM is modular*. Figure 3.4 gives an overview of the task decomposition at a high level. Tables 3.2, 3.3 and 3.4 give more information to the high-level decomposition. The table headers are the building blocks of the modular DSM matrix.

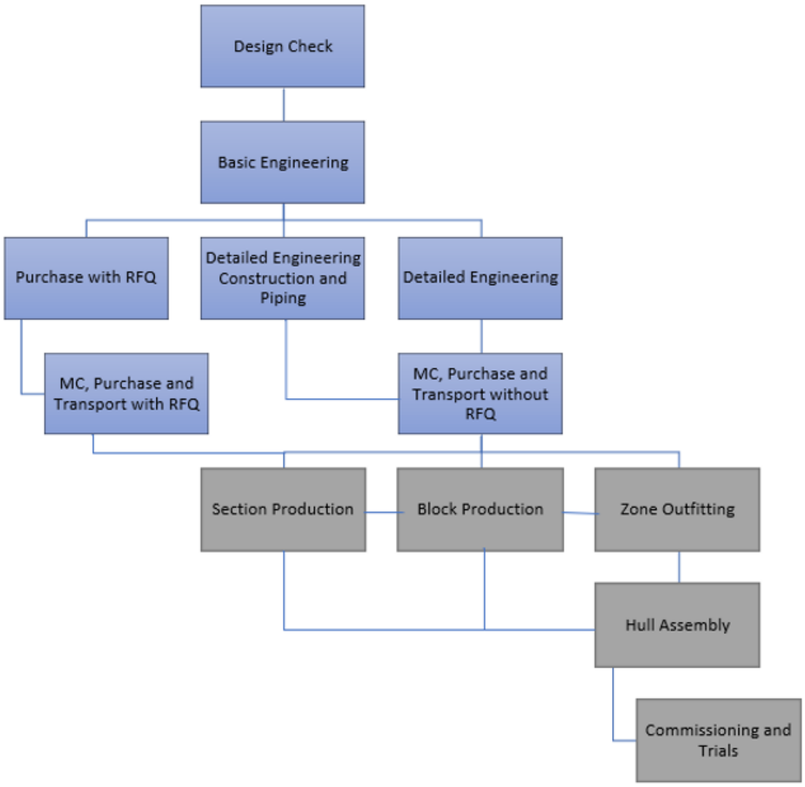


Figure 3.4: High-level task decomposition of a DTC-project, grey blocks represent yard activities. RFQ stands for request for quotation.

Table 3.2: Tasks of the design check, basic engineering and detailed engineering

Design check	Basic Engineering	Detailed Engineering
1. Shipbuilding 2. Carpentry 3. Mechanical 4. Nautical and electrical 5. Electrical and automation 6. Nautical and communication 7. Stakeholder review	**1. Shipbuilding** 1. Arrangements 2. Construction 3. Rules and regulations 4. Carpentry 5. Stability **2. Mechanical** 1. Arrangements 2. Diagrams 3. System design descriptions 4. Preliminary 3D-routing **3. Nautical and electrical** 1. Electrical and automation 2. Nautical and communication	**1. Shipbuilding** 1. Arrangement 2. Construction 3. Carpentry 4. Stability 5. Outfitting Hull sections 6. Outfitting superstructure section **2. Mechanical** 1. Diagrams 2. Arrangements **3. Batches** 1. Construction batch 2. Piping batch

3.3.2. Rework probabilities

For each dependency link in the binary DSM, a rework probability value exist. These values are crucial simulation input, but assigning rework probabilities is rather difficult. Yassine et al. (2001) define two reasons why it is challenging to define accurate values for rework probabilities:

1. engineers and designers are generally uncomfortable providing direct probabilities; and
2. provided estimates vary widely.

Input quality defines simulation output quality. Thus a reliable method to determine rework probabilities is essential. A confident method is found in the *Rework Probability Assessment* advised by Yassine

Table 3.3: Tasks of purchase and logistics, section production and block production

Purchase and logistics	Section production	Block production
1. Document control 2. Completion bill of materials 3. Purchase process 4. Inbound logistics 5. Outbound logistics 6. Transport coordination	1. Job preparation 2. Parts fabrication 3. Quality control parts fabrication 4. Section assembly 5. Quality control section assembly 6. Workshop for section outfitting 7. Quality control workshop for section outfitting 8. Section outfitting 9. Quality control section outfitting 10. Section painting	1. Job preparation 2. Workshop for block outfitting 3. Quality control workshop for block outfitting 4. Block assembly and weld 5. Quality control block assembly and weld 6. Block outfitting: equipment 7. Quality control block outfitting equipment 8. Block outfitting: hot works 9. Quality control block outfitting: hot works 10. Painting block assembly

Table 3.4: Tasks of zone outfitting, hull assembly and commissioning

Zone outfitting	Hull assembly	Commissioning
1. Job preparation 2. Workshop for outfitting 3. Quality control workshop for outfitting 4. Zone outfitting: hot works 5. Zone outfitting: equipment 6. Zone outfitting: finalise 7. Quality control zone outfitting	1. Hull assembly block x 2. Quality control hull assembly block x	1. Launch 2. Commissioning and yard trials 3. Sea trails 4. Technical acceptance and outstanding items

et al. (2001).

With the assessment, direct rework probability values are no longer required. Rework probability is described by *Information Variability (IV)* and *Task Sensitivity (TS)*. Where *IV stands for the likelihood that information will change and TS stands for the sensitivity of a task for such change*. IV and TS values are assigned from a predefined scale presented by Tables 3.5 and 3.6. Multiplication of IV and TS results in *Task Volatility (TV)* values, Equation 3.6. *TV is not a probability; it represents relative probability to other tasks on a scale from 1 to 9*, Table 3.7.

$$TV = IV \times TS \tag{3.6}$$

Table 3.5: Levels of Information Variability (IV) as stated by Yassine et al. (2001)

Value	Description	Likelihood of change
1	Stable	Low
2	Unknown	Medium to high
3	Unstable	Very high

A DSM-based discrete-event Monte Carlo simulation is required to convert TV values to the probability domain. This simulation is similar to the simulation discussed in Subsection 3.4.2. A probability (P-value) is assigned to the largest TV value. P-value for other TV values are calculated from a linear relation. Other relationships are possible, but result in less conservative probability values. The P-value is increased after every simulation run until the simulation outcome matches the mean process duration. In this study mean process duration is defined by the scheduled duration. TV values are not converted to probabilities here as scaling is project-specific. More information on the mapping and calibration method is available in Appendix A.2. Results of the scaling procedure for a DTC-project are described in Section 4.3.

Table 3.6: Levels of Task Sensitivity (TS) as stated by Yassine et al. (2001)

Value	Description	Dependent tasks is:
1	Low	Insensitive to most information changes
2	Medium	Sensitive to major information changes
3	High	Sensitive to most information changes

Table 3.7: Levels of Task Volatilities (TV) as stated by Yassine et al. (2001)

Value	Description	Strategies
1 or 2	Dependency is weak, low risk of rework	Feedback loops in documentation can be used, especially if it promotes process robustness
3 or 4	Dependency is moderate, moderate risk of rework	Avoid feedback loops in documentation were possible
6 or 9	Highly sensitive to change, high risk of rework	Do not use feedback loops in documentation

3.3.3. Rework impact

Rework impact represents the consequence of rework, which can differ from task to task. Each link in the binary DSM has its rework impact. Thus rework impact on a specific task depends on the task that causes rework. *It is assumed that rework impact is a fixed percentage*. So it does not change with the number of iterations. For shipbuilding projects, it is hard to assign a rework impact value. Rework consequences are influenced by time and the severity of rework. Compared to literature, rework impact values for DTC-project tasks are small. Rework on a section will for example not lead to an increase of 30%-50% of the initial work but rather, as concluded from interviews, values around 1%-5%.

A data collection survey asking for values between 0% and 100% is not suitable for the required accuracy. It is not precise due to small values and the fact that a percentage is a relative value. Nonetheless, rework impact is an essential input. Misplaced rework impact values affect process duration and cost. So assigning rework impact values demands attention. These characteristics make the data collection process for rework impact difficult and timely.

Similar activities are grouped to reduce time. For the tasks in each group, a rework impact value in hours is proposed based on activity duration. For example, one value is assigned to all activities of Basic Engineering. Sometimes not enough information is available to argue a proposed value and the field is kept empty. Trough interviews proposed impact is discussed. Remaining values are identified by the (in general) required hours for typical rework on an activity. Also, this method does not result in exact numbers suitable for all DTC-projects. However, *it is essential that order of magnitude matches*. Minor deviations have a low impact on simulation outcome due to the scaling process of rework probabilities.

3.4. Develop a DSM-based documentation flow simulation

The development of the DSM documentation flow simulation consists out of two parts. The first part is the development of an algorithm to generate project-specific DSM matrices. The second part consists of the implementation of the DSM-based discrete-event Monte Carlo simulation as described by Browning and Eppinger (2002) in MATLAB. Subsection 3.4.1 handles the first part. The second part is described in Subsection 3.4.2.

3.4.1. Algorithm for project-specific DSM

Essential input for the DSM-based discrete-event Monte Carlo simulation are three matrices:

1. *Binary DSM* (DSM);
2. *Matrix with Rework Probabilities* (DSM_1); and
3. *Matrix with Rework Impact* (DSM_2).

These matrices differ from project to project. A MATLAB script is used to prepare project-specific DSM matrices. The algorithm is explained step-by-step, Figures 3.5 and 3.6 support the explanation. The full MATLAB script is enclosed in Appendix B.1.

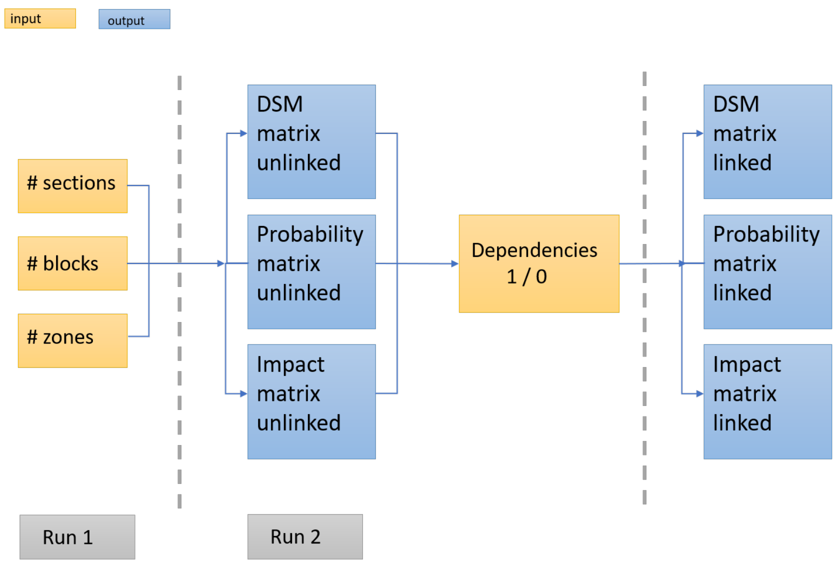


Figure 3.5: Flowchart to the algorithm to generate project-specific DSM matrices

The algorithm executes the following steps with the input defined in Figure 3.5:

1. **Load Standard DSM Matrices**
   Standard matrices are defined for the binary DSM, rework probabilities and rework impact in Subsections 3.3.1, 3.3.2 and 3.3.3. The standard matrices includes all modelled tasks of a DTC-project. Tasks are not grouped yet. It should be noted that all tasks are only named once. This forms the building block that can be tailored to a case study.

2. **Group tasks into blocks**
   Tasks from the Standard matrices are divided into groups as *Basic Engineering, Detailed Engineering* and *Section Production*. These grouped tasks form the building blocks of the modular DSM matrices. The tasks in Tables 3.2, 3.3 and 3.4 represent the different DSM building blocks. The matrices with the DSM building blocks are called $BLOCK - DSM$, $BLOCK - DSM_1$ and $BLOCK - DSM_2$ in the rest of this description.

3. **Define number of DSM building blocks**
   The amount of DSM building blocks has to be defined to determine matrix size. All matrices are equal in size.

4. **Create new DSM matrices with tasks based on the defined number of DSM building blocks** BLOCK-DSM matrices groups are printed multiple times based on defined building blocks. No dependencies among groups are included at this stage.

5. **Add dependencies**
   Based on the building strategy dependencies are identified. Dependencies among simulation building blocks are defined in a separate matrix.

6. **Print Project Specific DSM**
   Finally the project-specific DSM structures are printed as numerical arrays. [In previous steps the matrices are cell arrays]

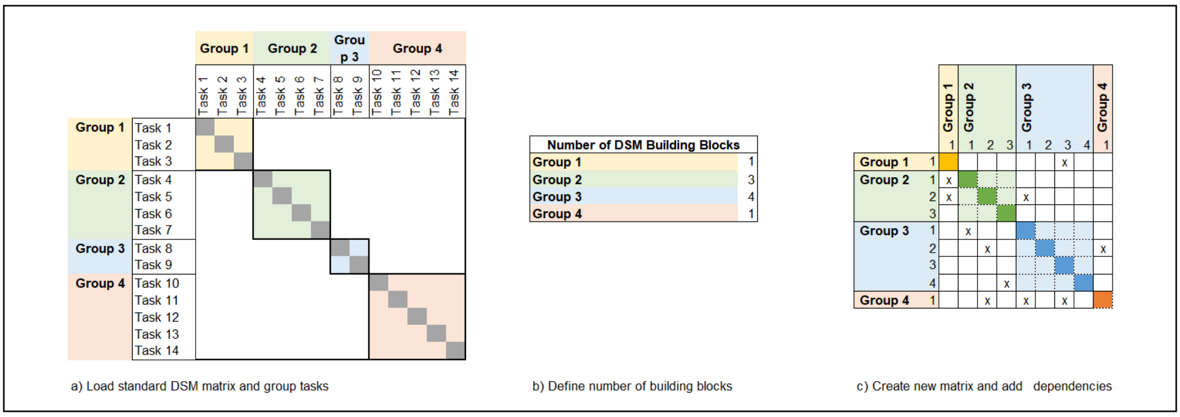


Figure 3.6: Procedure to create project-specific DSM matrices

3.4.2. Algorithm for DSM-based discrete-event Monte Carlo simulation

The algorithm of the DSM-based discrete-event Monte Carlo simulation is based on work of Browning (1998) and Browning and Eppinger (2002). This subsection describes the algorithm step-by-step. For each step formulas and assumptions are stated. Explanation of model variables is found in the Nomenclature. The full MATLAB script is given in Appendix B.2.

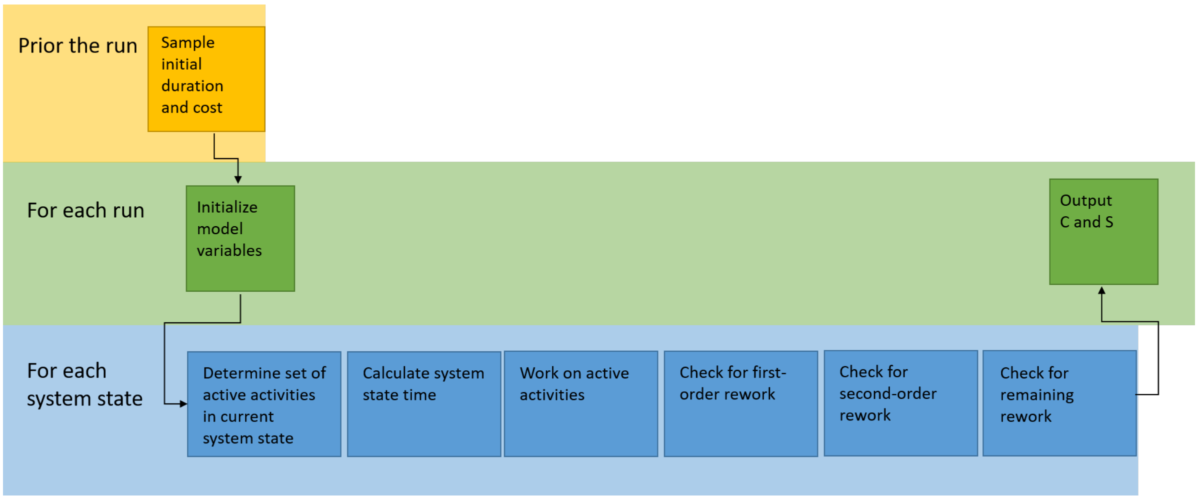


Figure 3.7: Flowchart of the DSM-based discrete-event Monte Carlo simulation

The below-listed tasks are executed by the algorithm to simulate a shipbuilding project by a DSM-based discrete-event Monte Carlo simulation. The DSM matrices constructed by the algorithm of Subsection 3.4 and information on activity cost and duration form the input for this algorithm.

1. **Randomly sample duration and cost of each activity**
   Before the simulation starts, initial tasks duration data is prepared. Because initial task durations are uncertain in engineer-to-order projects activity duration and cost are randomly sampled from a Triangular distribution. The Triangular distribution is used to include the attempt to relax when work is ahead of schedule. Duration and cost are correlated, a correlation similar to Browning and Eppinger (2002) of 0.9 is used. In the literature study was concluded that Latin Hypercube Sampling (LHS) is an effective technique to create a correlated sample from a triangular distribution. Because no MATLAB built in function for LHS existis, the lhsgeneral function from the MATLAB File exchange is used to sample correlated values from the distribution (Iman, 2013). Sampled values are stored in two separate n x m matrices. Where n stands for number of runs and m for the number of activities.

2. **Initialize model variables**
   At the start of each run, work vector, loop counting, cumulative cost and duration are reset to initial values. The reset ensures that at each run, the entire process is simulated. One run represents the execution of one project.

   - **Work vector**: a vector with the remaining work for all activities.
   - **Loop counting**: a value that counts number of system states before the entire project is completed.
   - **Cumulative cost and duration**: cost and duration of each system state are summed.

3. **Determine set of active activities**
   Many system states, time-steps, are needed to complete a run. For each system state, a set of active activities is determined. Active activities are activities where is worked on in the current system state. The selection of active activities is based on the following work policy:

   (a) Reset the work now vector (**WN**), the vector that represents all active activities in the current state. Active activities of a previous state are not relevant for the current state.

   (b) Find the most upstream activity with unfinished work (**W** > 0) and add this activity to the set of active activities.

   (c) Check for subsequent activities that have unfinished work AND are independent on unfinished upstream activities, add those activities to the set of active activities. Otherwise, the active set is found.

4. **Calculate system state time**
   The system state time is required to calculate a system states cost, duration and done work. The activity with the shortest duration determines the duration of the current system state. Thus *at least one activity is completed during each system state*. Equation 3.7 demonstrates how the minimum duration is determined. Work done in a previous system state is taken into account.

$$t = min(\boldsymbol{W} \cdot \boldsymbol{WN} \cdot \boldsymbol{S_D}) \tag{3.7}$$

5. **Work on active activities**
   Work is done on all activities in the active set. Done work results in a decrease in required work stored in the work vector (**W**) and an increase in cumulative process cost (C) and time (S).

   (a) Decrement **W**
   Equations 3.8 and 3.9 show the procedure for decrement **W** for the current system state. In this equations, $\boldsymbol{S_d}$ stands for sampled duration, $\boldsymbol{W_d}$ for work done and t is the current system state time.

$$\boldsymbol{W_d} = \frac{\boldsymbol{WN} \cdot t}{\boldsymbol{S_D}} \tag{3.8}$$

$$\boldsymbol{W} = \boldsymbol{W} - \boldsymbol{W_d} \tag{3.9}$$

(b) Increment S and C
The cumulative duration S is calculated by Equation 3.10 for every system state. Almost similar procedure exists for cumulative cost C in Equation 3.11. In this equation, $\boldsymbol{S_C}$ stands for sampled cost and m for the number of tasks.

$$S = S + t \tag{3.10}$$

$$C = C + \sum_{1}^{m} (\boldsymbol{W_{d,m}} \cdot \boldsymbol{S_{C,m}}) \tag{3.11}$$

6. **Check for first order rework by the completed activity**
Once an activity is completed it can generate first-order rework on previous activities. Due to the uncertain characteristic of rework is it generated by a random number. Since current system state duration is dominated by activity duration, at least one activity is finished at every state. *A finished activity can cause rework on upstream activities*; this is called *first-order rework*. The following procedure is used to check whether activity j causes *first-order rework* on activity i:

   (a) Evaluate all entities in $\boldsymbol{DSM_1}$ in column j above activity j against a random number. *Rework occurs if the random number is smaller than the rework probability*.

   Activity j causes first-order rework on activity i = $\boldsymbol{DSM_1}(1 : j-1, j) > RAND(j-1, 1)$ (3.12)

   (b) If rework occurs, add additional work to the work vector by rework impact.

$$\boldsymbol{W_{add}} = \boldsymbol{DSM_2}(i, j) \cdot \boldsymbol{W} \tag{3.13}$$

   (c) When additional work results in work vector entities exceeding 1, entities are adjusted to 0.9. This procedure prevents work from expanding beyond the original scope.

7. **Check for second order rework**
As a consequence of first-order rework, following activities could also required rework (second-order rework). Again, rework is uncertain. Thus also here a random number is used to generate rework.

   (a) Evaluate all entities in $\boldsymbol{DSM_1}$ in column i (activity with first-order rework) below activity i against a random number. *Second-order rework occurs if the random number is smaller then the rework probability, and there is already work done on the activity*. It is not possible to have rework on an activity where it is not worked on yet.

   (b) If rework occurs, add additional work to the work vector by rework impact.

   (c) When additional work results in work vector entities exceeding 1, entities are adjusted to 0.9. This procedure prevents work from expanding beyond the original scope.

8. **Check if there is remaining work**
If work is remaining (**W** > 0), go back to step 3 and repeat. System state is incremented.

9. **Output C and S**
When no work is remaining (**W** = 0) store C and S values in a vector, the number of system runs determines the length of this vector. This vectors are used to generate simulation outcome plots.

10. **Check Output Stability for Number of Runs**
A Monte Carlo simulation requires sufficient runs to generate stable outcome distributions. Browning and Eppinger (2002) use Equations 3.14 and 3.15 to check outcome distribution stability. Thus no fixed number of required runs is determined yet. However, required runs are determined for each situation. Number of runs is increased by batches (b) until the requirements in Equations 3.14 and 3.15 are satisfied.

$$D_{mean}(n) = \frac{|E[S_n] - E[S_{n-b}]|}{E[S_{n-b}]} < \alpha \tag{3.14}$$

$$D_{var}(n) = \frac{|\sigma^2_{s,n} - \sigma^2_{s,n-b}|}{\sigma^2_{s,n-b}} < \alpha \tag{3.15}$$

$$\alpha = 0.01 \quad [-]$$
$$b = 100 \quad [runs]$$

3.5. Simulation outcome

Multiple plots are the result of the simulation. The different plots give insight in the project environment. Interface criticality plots, as shown in Subsection 3.5.1 show assumptions that can result in high project cost or duration when rework occurs. The probability distribution plots handled by Subsection 3.5.2 give insight in the probability distribution of project cost and duration. The joint distribution plots in Subsection 3.5.3 visualize the probability that a project is completed within time and budget. The rework contribution plots illustrate the effect of rework on total cost and duration in Subsection 3.5.5.

3.5.1. Interface criticality plots

Interface criticality can be visualised by two three-dimensional bar plots: one for the duration and one for cost. A MATLAB script is created based on work by Browning and Eppinger (2002) and Browning (2000) on the Product Development Process of an Uninhabited Combat Aerial Vehicle (UCAV). These publications give required data for the construction of interface criticality plots of the product development process of a UCAV, which consist of 14 (inter)dependent tasks. Figures 3.8 and 3.9 show the interface criticality plots created by this script. The generated plots are similar to the plots in the publications.

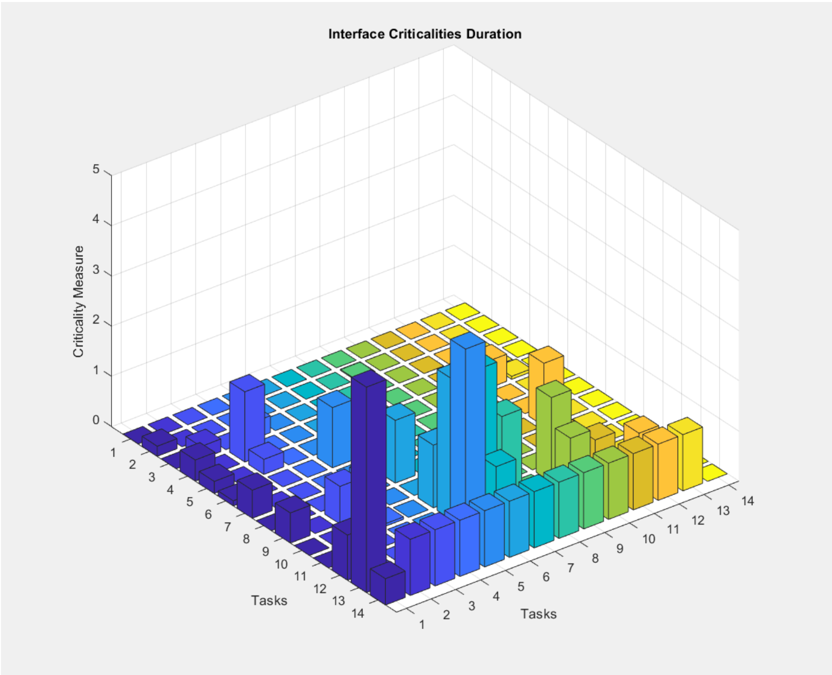


Figure 3.8: Example of interface criticality plot for duration created by the author using MATLAB

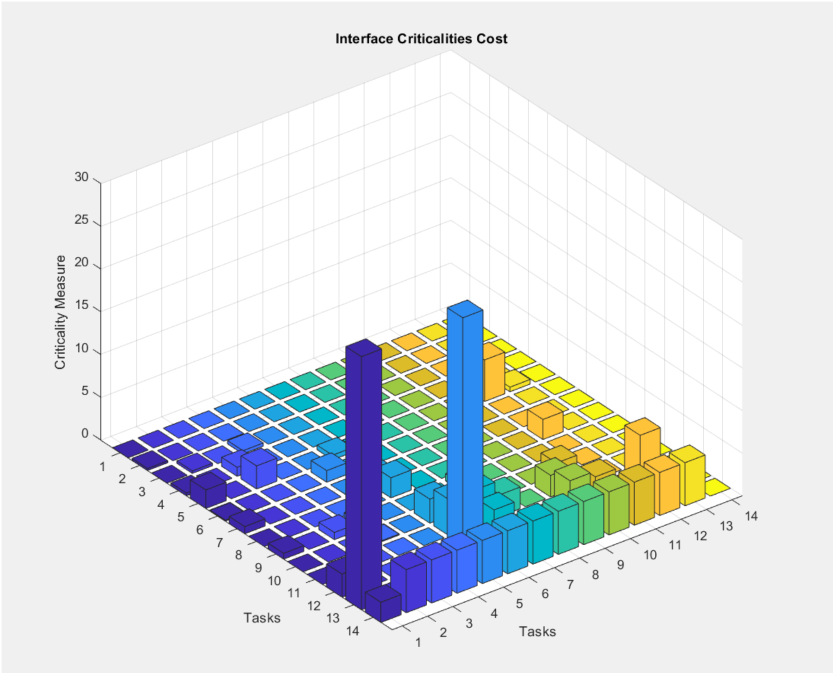


Figure 3.9: Example of interface criticality plot for cost created by the author using MATLAB

3.5.2. Probability distribution plots

The Probability Mass Functions (PMF) and Cumulative Distribution Functions (CDF) for process duration and costs are plotted on a histogram with a target value, displayed in Figures 3.10 and 3.11.

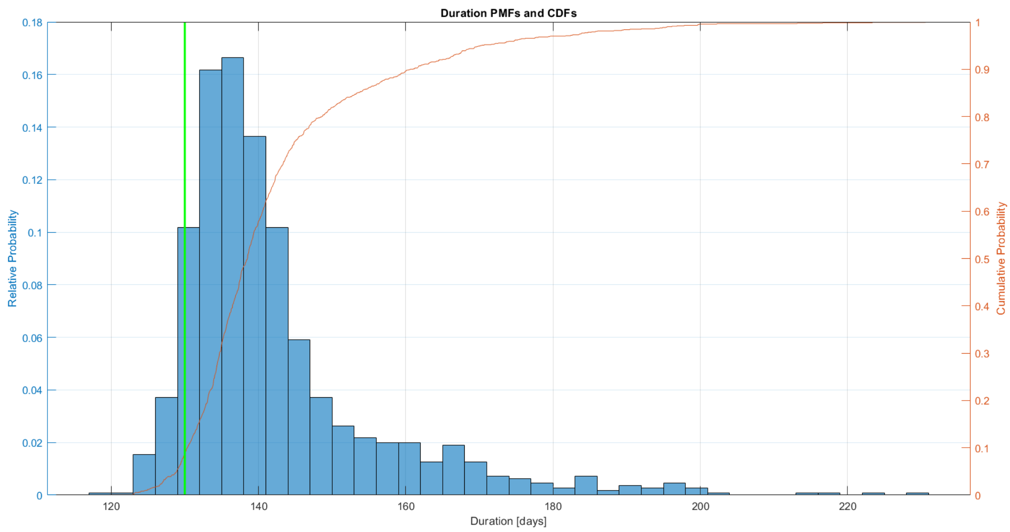


Figure 3.10: PMF, CDF and target value (130 days) for process duration

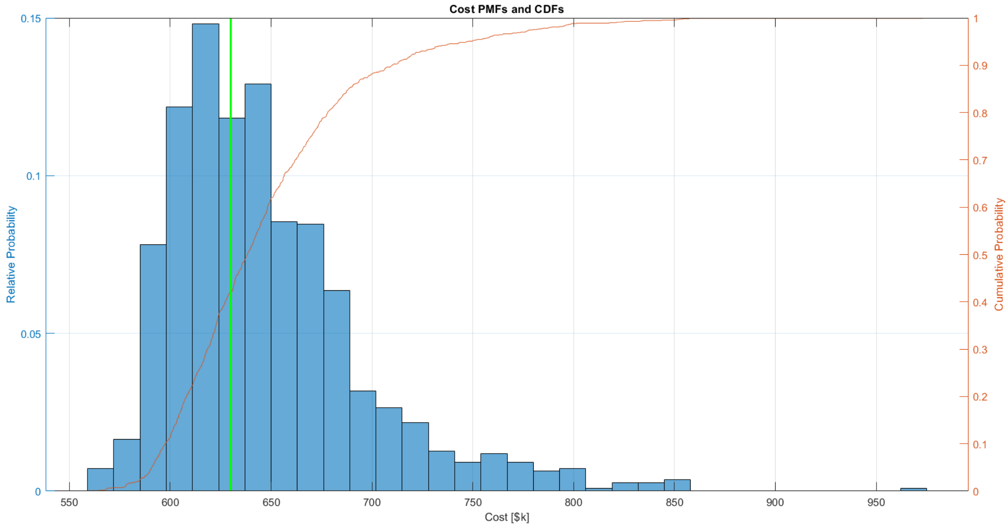


Figure 3.11: PMF, CDF and target value (630 k$) for process cost

3.5.3. Joint distribution plots

The joint distribution plots provides insight in the likelihood a project is finished in time and within budget. Moreover, it makes it possible to see what is more critical for the project, cost or duration. Figure 3.12 gives an example of such a joint distribution plot for the product development process of an Uninhabited Combat Aerial Vehicle.

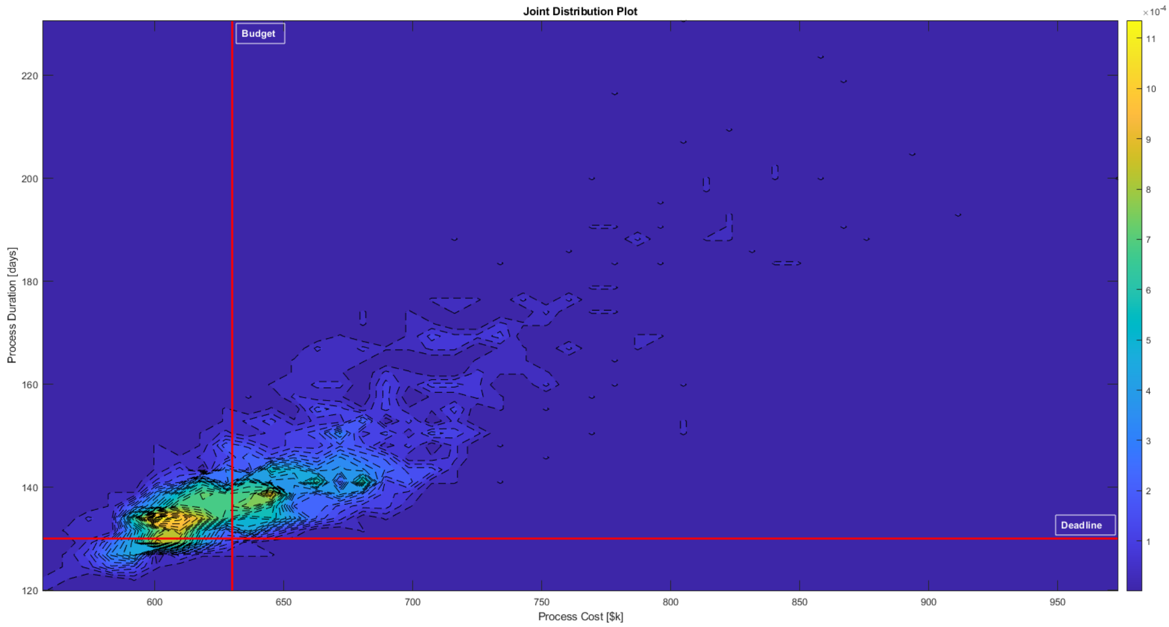


Figure 3.12: Example of joint distribution plot for the UCAV product development process created in MATLAB

3.5.4. Schedule and cost risk

Process architectures, dependency structures among tasks, are characterised by schedule and cost risk (Browning and Eppinger, 2002). Schedule risk is the risk to exceed a deadline for a certain process architecture. Cost riks is the risk to exceed a budget for a certain process architecture. Thus, a different process architecture can result in different schedule and cost risk values. A change in activity sequence, changes the process architecture. Risk is the product of probability for a particular outcome and the consequence of that outcome, also presented by Equation 3.16. In this master thesis, process outcomes that exceed project deadline or budget are of interest. Incentives for early finish or finish below budget are not included.

> **Process Architecture**: The structure of a process - embodied in its activities and their interactions with each other and the process environment - and the principles guiding its design and evolution (Eppinger and Browning, 2012).

$$\text{Risk} = \text{Probability} \cdot \text{Impact} \tag{3.16}$$

Schedule and cost risk are determined using the same approach. First, the general probability of an unacceptable outcome is identified, which is defined as an outcome that exceeds the target value (T_s). Equations 3.17 and 3.18 demonstrate the used method to calculate the likelihood of an unacceptable outcome. ECDF is the Empirical Cumulative Distribution Function. Since no MATLAB built-in function is available for the allocation of intersections between lines, a function from MATLAB's file exchange is used (Schwarz, 2009).

$$P_{acceptable} = intersections(ECDF, T_s) \tag{3.17}$$

$$P_{unaccepatble} = 1 - P_{acceptable} \tag{3.18}$$

Since a Monte Carlo simulation is used, has each run the same probability. Thus individual outcomes have similar probability. Meaning that the likelihood is defined by Equation 3.19.

Outcome impact depends on the outcome. Outcomes below the target value have no impact. Equation 3.20 is used to determine which run exceeds the target value and their impact is calculated using Equation 3.22. The risk values are computed by taking the product of probability and impact in Equation 3.22. Finally, schedule risk is obtained by the sum of all risk values in Equation 3.23. A similar approach is applied for cost risk.

$$\text{Probability for the outcome of 1 run} = \frac{1}{n} \tag{3.19}$$

$$\boldsymbol{S_{impact}} = \boldsymbol{S_{TOT}} > T_s \tag{3.20}$$

$$\boldsymbol{S_{IMPACT}} = \boldsymbol{S_{impact}} \cdot (\boldsymbol{S_{TOT}} - T_s)^2 \tag{3.21}$$

$$\textbf{Risk Values} = \frac{\boldsymbol{S_{IMPACT}}}{n} \tag{3.22}$$

$$\text{Schedule Risk} = sum(\textbf{Risk Values}) \tag{3.23}$$

3.5.5. Rework contribution plots

Violin plots in Figure 3.13 are used to show rework and its distribution for all tasks. A violin plot is similar to a box plot, however it provides more information. No MATLAB built-in function is present to create such plots. Thus, a function from the MATLAB File Exchange is used to generate violin plots (Dorn, 2017).

In Figure 3.13 median is given with a red cross, quartile data is visualised with a green box. The ability to present the distribution together with this statistical data makes the violin plot unique. These characteristics make it an attractive tool to display rework data in processes. Box plots are less suitable for this application, because rework for all tasks starts from zero. Using a box plot creates less insight into the lower part of the data.

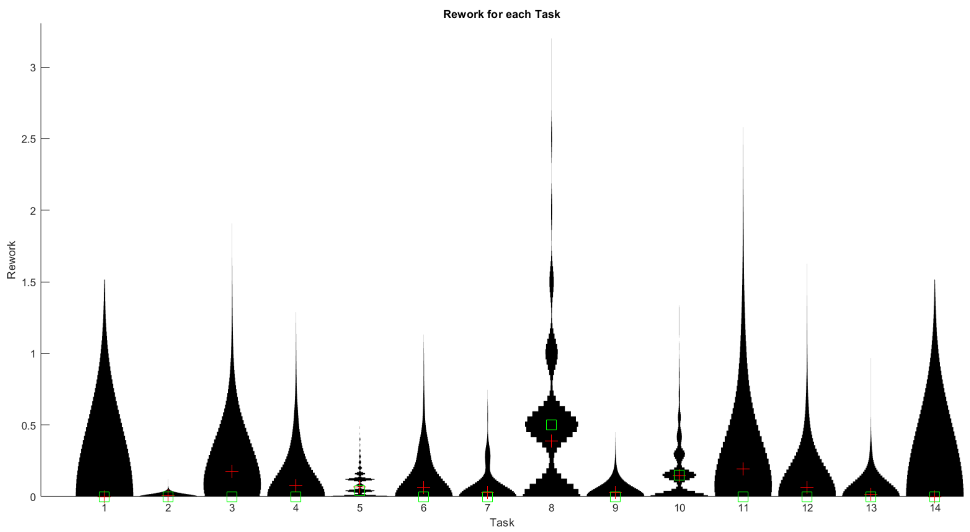


Figure 3.13: Violin plot with rework on tasks | red cros: median, green box: percentile data

3.6. Simulation verification and validation

For verification and validation of the simulation model, the code has been verified using the paper that the simulation model is based on (Browning and Eppinger, 2002). The design process of an uninhabited combat aerial vehicle (UCAV) is simulated by 14 tasks (Browning and Eppinger, 2002). Because the models are equal, the simulation outcome can be compared to the results from literature. Also, it is easier to verify and validate the simulation with a small number of tasks. Binary DSM, rework probability and rework impact matrices are only model inputs. Thus using different Binary DSM, rework probability and rework impact matrices does not affect verification and validation of the DSM-based discrete-event Monte Carlo simulation algorithm.

It should be noted that the project-specific DSM matrices: the binary DSM, rework probability DSM and rework impact DSM are not included in this verification and validation. Project-specific data is discussed in the case study in Chapter 4.

3.6.1. Verification

Multiple methods are used for verification of the simulation model. First, one-step-analysis is carried out. Next random generators with different seed numbers are used. Besides, the model is run with other random generators. Finally, the simulation is verified with extreme input values.

As the script for the simulation consists out of many different loops (if-else, while and for loops), it is hard to follow each sequence individually when the model is run in its entirety. Therefore, loops are disconnected for verification to keep track of actions inside the loop.

Sampled initial duration and cost and the occurrence of rework depend on random variables. Although random variables are built-in MATLAB functions, they are not entirely random (Mat). Random generator outcome depends on seed number and random generator type. The simulation needs to be independent of seed numbers and random generator types. Three seed numbers, namely 0 (MATLAB default), 1 and 5, and two random generator types, Twister (MATLAB default and Phillox, are tested. The outcome is presented in Tables 3.8 and 3.9. Independence is checked with a two sample T-test for equal means under the assumption of unequal variances.

T-tests are conducted with the data analysis add-in for Microsoft Office 365 Excel. Under the null hypothesis (Equation 3.24) are the mean values for both sets statistically equal. The null hypothesis is rejected if the two-tailed P-value is smaller than the critical value, $\alpha = 0.05$. Table 3.10 provides P-values for different combinations of duration outcome. Similar data is provided for cost in Table 3.11. All values are greater than 0.05. Thus *outcome is independent to seed number and random generator type*.

Table 3.8: Results for different seed numbers using random generator 'Twister'

	Seed 0	Seed 1	Seed 5
Mean Duration	142.32	141.83	142.22
Standard deviation	14.01	13.01	14.09
Mean Cost	647.65	645.99	648.19
Standard Deviation	49.01	48.37	50.91
N	1100	1100	1100

Table 3.9: Results for different seed numbers using random generator 'Philox'

	Seed 0	Seed 1	Seed 5
Mean Duration	142.39	142.39	141.72
Standard deviation	14.40	14.40	13.83
Mean Cost	648.64	648.88	645.96
Standard Deviation	53.62	53.19	50.68
N	1100	1100	1100

$$H_0 : \mu_1 = \mu_2 \quad (3.24)$$

$$H_1 : \mu_1 \neq \mu_2 \quad (3.25)$$

Table 3.10: Two-tailed P-values T-test for duration outcome

	Twister 0	Twister 1	Twister 5	Phillox 0	Phillox 1	Phillox 5
Twister 0	X	0.39	0.86	0.72	0.91	0.31
Twister 1	0.39	X	0.51	0.23	0.34	0.85
Twister 5	0.86	0.51	X	0.59	0.77	0.41
Phillox 0	0.72	0.23	0.59	X	0.95	0.96
Phillox 1	0.91	0.34	0.77	0.95	X	0.27
Phillox 5	0.31	0.85	0.41	0.96	0.27	X

Table 3.11: Two-tailed P-values T-test for cost outcome

	Twister 0	Twister 1	Twister 5	Phillox 0	Phillox 1	Phillox 5
Twister 0	X	0.42	0.80	0.65	0.57	0.43
Twister 1	0.42	X	0.30	0.22	0.18	0.99
Twister 5	0.80	0.30	X	0.84	0.76	0.30
Phillox 0	0.65	0.22	0.84	X	0.92	0.23
Phillox 1	0.57	0.18	0.76	0.92	X	0.19
Phillox 5	0.43	0.99	0.30	0.23	0.19	X

Another way to verify the simulation model is by using extreme input values. For extreme values, it is easier to reason the expected behaviour of the model, then when more normal values are used. First initial task duration for all tasks is increased. With an extreme increase in initial duration, it is expected to see a significantly longer process duration and an increasing effect of schedule overruns on project outcome. $\boldsymbol{MLV_d}$, $\boldsymbol{BCV_d}$ and $\boldsymbol{WCV_d}$ values are multiplied by 100 to realise an extreme increase. It results in a process duration of 14286 days, significantly more notable than the initial duration (142.35 days). No significant change is expected for process cost, as initial cost, rework probability and rework impact are not changed. Process cost are dependent on active tasks; change in activity duration can cause a small difference in the set of active tasks. It results in a slightly higher value for process costs (649.19

compared to 647.65). Results and expectations match; thus, this extreme input does not harm the simulation.

Reducing rework impact with factor 10, is expected to result in less rework. As rework has effect on process cost and duration, lower values are expected. Simulation outcome matches expectations by a mean duration of 134.23 days, mean cost of 616.57 and rework on tasks range 0-0.4. Again this input does not harm simulation output. Thus, *the simulation is not affected by the tested extreme input values*.

It is expected that changes in process architecture affect rework and thereby, also process cost and duration. Equation 3.26 represents the initial sequence vector, the adjusted sequence vector is given by Equation 3.27.

$$V_{int} = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] \quad (3.26)$$

$$V_{adj} = [1, 2, 3, 4, 5, 13, 6, 7, 8, 9, 10, 11, 12, 14] \quad (3.27)$$

A significant difference is seen in results of the initial and adjusted sequence for both duration (142.3 days compared to 110.3 days) and cost (647.7 $k compared to 664.4 $k). Thus *the simulation model is able to represent different process architectures and their influences on project outcome*.

3.6.2. Validation

The aim of simulation validation is to ensure that the model is useful. A useful model represents the process behaviour of interest. However, a useful model does not have to represent the entire process accurately. Model validation is closely related to the use of the model. Validation comes in three levels: 1) assumptions, 2) input parameter and distribution and, 3) output value and conclusion (Hil). In reality, it is hard to validate a model on all levels. In this project, the focus of the validation will be on output values and conclusions. Validation at other levels is considered when validation at output level gives reasons to do so.

Comparing simulation outcome with real process data is preferred, but often impossible. Data is not available; measurements are time-consuming or have a high cost. Also, for DTC projects, no real data is available. Therefore, the validation model from Browning and Eppinger (2002) developed for DSM-based discrete-event Monte Carlo simulation is seen as a good alternative. Input data is similar as used for verification in 3.6.1. The downside of this technique is that model validity is dependent on the validity of Browning and Eppinger (2002) model. However, after Browning and Eppinger (2002), a lot of research on DSM-based discrete-event Monte Carlo simulation has been published, where Browning and Eppinger (2002) model has been used for validation. This drawback should be noted, but given the circumstances, no other validation method is better. Output is compared on:

- Probability mass distribution
  - Process duration
  - Process cost
- Differences for 6 different process architectures:
  - Mean duration
  - Mean cost
  - Standard deviation duration
  - Standard deviation cost
  - Probability on unacceptable outcome
  - Process architecture risk

Comparison of probability mass distributions for process duration in Figure 3.14 and cost in Figure 3.15 shows some differences between the two models. The developed model shows denser distributions than the model of (Browning and Eppinger, 2002). Due to the denser distribution, mean results of the developed simulation are slightly higher but still close to the model of Browning and Eppinger (2002). The difference can be caused by a difference in used random generator or other software

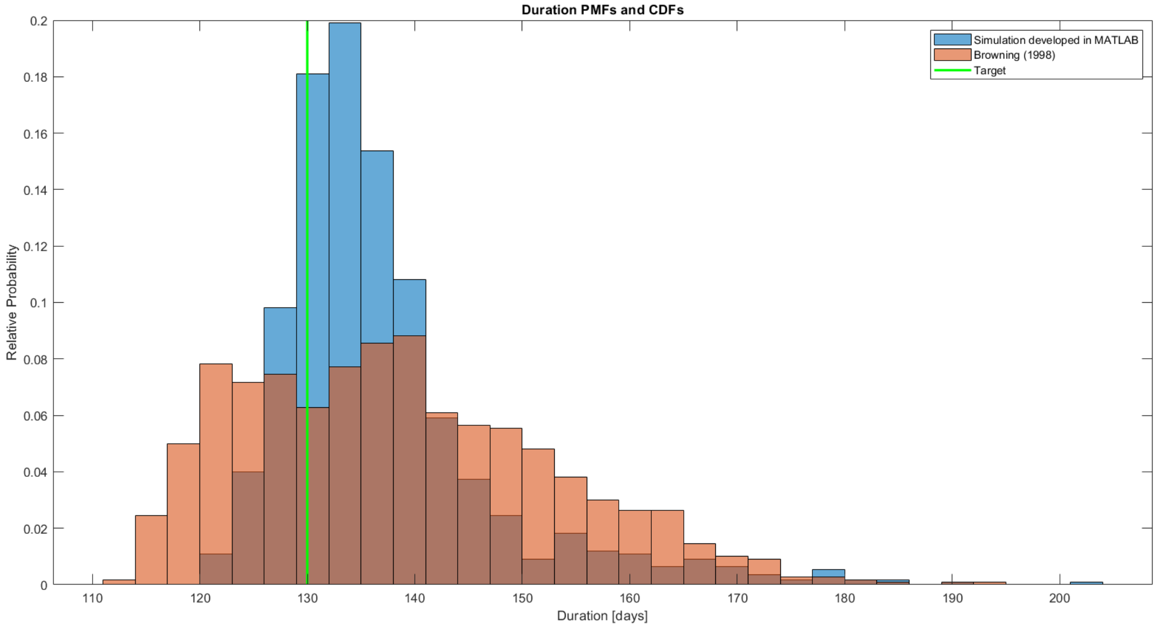


Figure 3.14: Simulation outcome compared to results of Browning (1998) for process duration

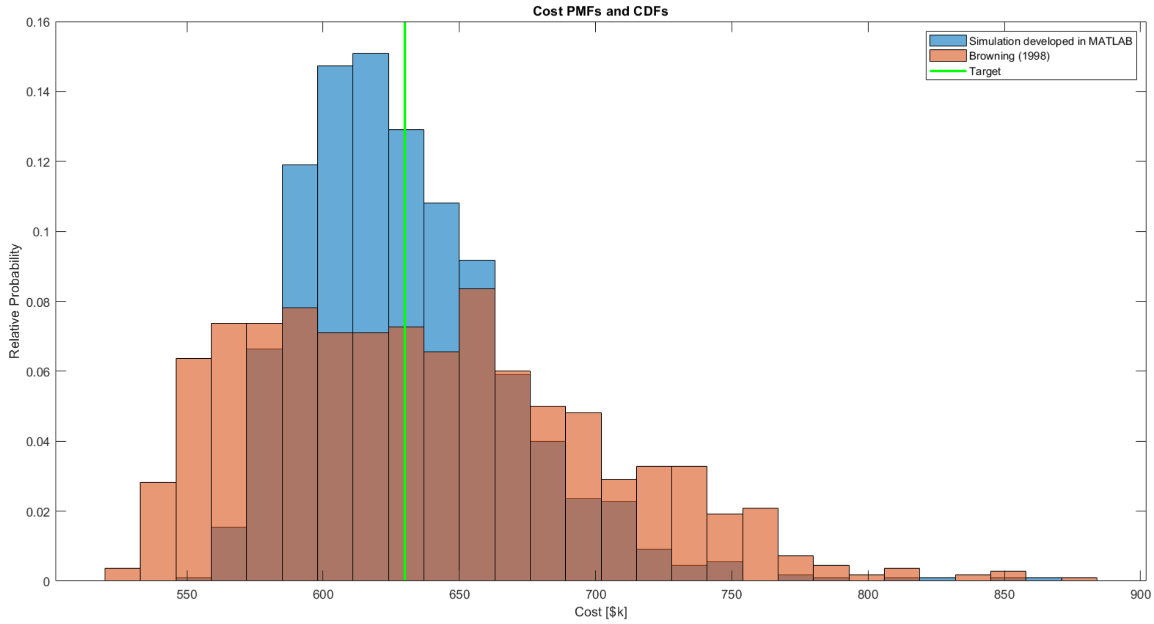


Figure 3.15: Simulation outcome compared to results of Browning (1998) for process cost

related differences. Browning and Eppinger (2002) developed the model in 1998 in Excel. The Excel 1998 random generator can differ from the MATLAB 2020a random generator. A two sampled T-test for means with unequal variance is carried out to examine if both samples can originate out the same sample. An α-value of 0.05 is used. T-test are only conducted for process architecture 1 (initial activity) as no detailed data is available for other process architectures. From the T-tests it is concluded that mean values for process duration and cost can be part of the same sample. Thus deviations in outcome are not significant.

$$H_0 : \mu_1 = \mu_2 \quad (3.28)$$
$$H_1 : \mu_1 = \mu_2 \quad (3.29)$$
$$\alpha = 0.05 \quad (3.30)$$
$$N = 1100 \quad (3.31)$$

Table 3.12 compares outcome for both models for 6 different process architectures. The activity sequence differs for the process architectures. Over all mean values for process duration and cost are slightly higher for the developed simulation. Although some values (mean duration 2 and 5, mean cost 1** and 4) are smaller, the differences here are smaller as well. Documentation flow analysis for DTC-projects with the developed simulation give conservative results. This outcome is not seen as a problem, because no other models examining engineer-to-order shipbuilding projects exist.

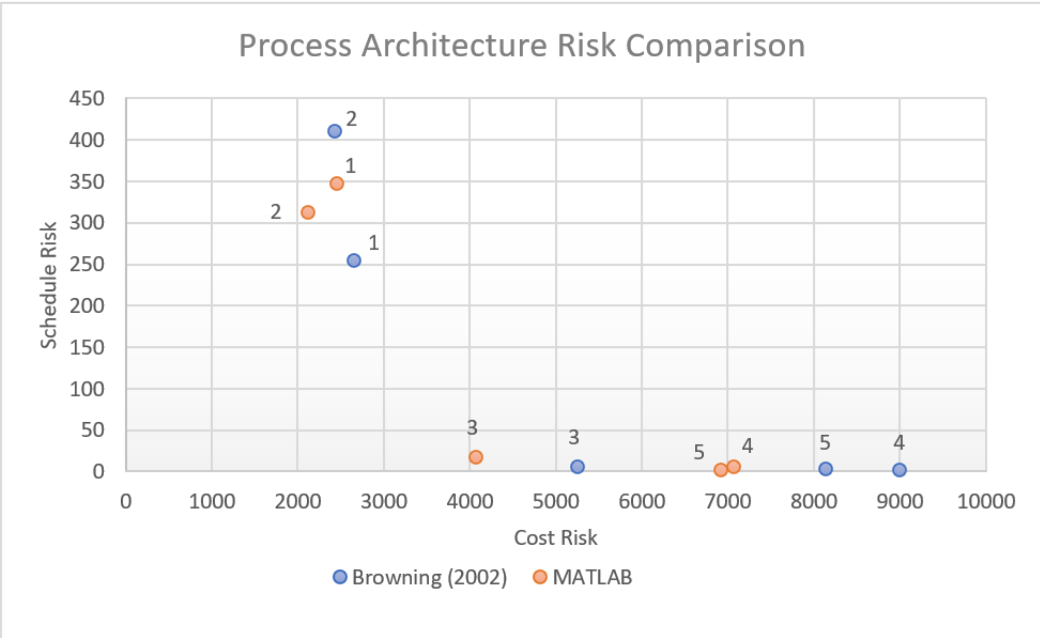


Figure 3.16: Process architecture risk comparison. Schedule and cost risk are dimensionless numbers.

Although mean values are close, a more substantial difference is seen in standard deviation and likelihood for an unacceptable process outcome—especially standard deviation for cost show higher difference.

Figure 3.16 visualises the risk for different process architectures. The report from Browning and Eppinger (2002) is comparable but some differences are found. According to Browning and Eppinger (2002) process architecture 2 has a higher schedule risk than process architecture 1. The devolved model shows an opposite result. Both models show that architecture 1 and 2 have schedule risk, but relative values differ. Taking the application of the model for this master thesis into account is this not seen as a problem. The model is able to give insight in risks of process architecture on project outcome, however, the model cannot be used for close comparison of process architectures.

Table 3.12: Comparison of MATLAB simulation outcome to simulation outcome by Browning and Eppinger (2002)

			Process Architecture					
			1**	1	2	3	4	5
Duration	mean	[days]	133.34	142.67	141.65	110.25	96.94	96.23
	mean*	[days]	133	138	144	108	95	97
	diff		0.26	3.38	-1.63	2.09	2.05	-0.8
	std	[days]	4.9	14.01	13.35	11.50	12.24	11.26
	std*	[days]	13	14	15	14	14	14
	diff	%	-62.31	0.09	-10.98	-17.83	-12.55	-19.59
	P(unacc)	[%]	-	91.1	9.2	6.54	2.40	1.75
	P(unacc)*	[%]	-	67	78	7	2	2
	diff	%	-	35.97	-88.2	-0.07	20	-12.5
Cost	mean	[$k]	613.57	647.65	646.05	664.43	681.19	683.02
	mean*	[$k]	615	637	634	660	681	677
	diff	%	-0.23	1.67	1.9	0.67	0.03	0.89
	std	[$k]	27.64	49.01	46.71	55.74	67.87	64.97
	std*	[$k]	55	63	62	72	84	81
	diff	%	-49.74	-22.21	-24.67	-22.58	-19.80	-19.80
	P(unacc)	-	-	57.90	57.01	70.09	76.02	77.88
	P(unacc)*	[%]	-	51	49	61	70	69
	diff	%	-	13.53	16.35	9.52	8.60	12.87

3.7. Chapter summary

This chapter summary gives an overview of the answered research questions by this chapter. Also, it is explained to the reader how these answers are used in the next chapters.

1. *How to develop a model for documentation flow analysis for engineer-to-order projects?*

    (a) Model Requirements

    i. **Must be able to use for different engineer-to-order shipbuilding projects at DTC.**
    In this chapter non project-specific DSM matrices are developed. Matrices are made project-specific with an additional algorithm. Thus, it is easy to construct DSM matrices for other engineer-to-order projects.

    ii. **Must represent documentation flow (in the form of documentation) between Engineering, Material Coordination, Purchase and a building yard.**
    Information flow among engineering, material coordination, purchase and a yard are included in the DSM matrices. Thus, this requirement is fulfilled.

    iii. **Must be able to represent the characteristics of an engineering-to-order project.**
    In particular:

    A. Uncertainty in initial task duration and cost.
    Latin Hybercube Sampling from a Triangular distribution is used to include uncertainty.

    B. Large (inter)dependency among tasks
    Number of (inter)dependencies is not limited by the model.

    C. Overlapping activities (concurrent)
    Overlapping is included.

    iv. **Must be able to give insight in the effect of rework caused by the documentation sharing strategy.**
    Different scenarios for rework can be tested. While Violin plots provide a clear overview of rework distribution on activities.

    v. **Must be able to adjust documentation sharing strategy based on gained insight in current situation.**
    It is possible to adjust documentation sharing strategy in different ways:

- Activity sequence can be adjusted. DSM matrices are adjusted to an alternative activity sequence by a user defined function.
- Work policy for selection active activities can be adjusted.
- Work policy for rework generation can be adjusted

Above mentioned points demonstrates that a useful model is developed in this chapter.

(b) Data Collection
Process data is collected through company documentation, interviews and surveys. Interviews were held with ten participants based on predefined question forms. The data collection process resulted in required input data for the simulation: a binary DSM, rework probabilities and rework impact.

(c) Model Output
The model can generate a wide variety of interesting output data. Most relevant output is seen in: join distribution plots for process duration and cost, process architecture risk and rework impact plots. This data gives insight in the modelled situation and shows directions for improvement.

2. *How to evaluate a model for documentation flow analysis for engineer-to-order projects?*

(a) Verification
One-step-analysis, simplified model, seed independence and degencry testing are used for verification.

(b) Validation
Validation is done on model output values and conclusions. Validation is based on theoretical results of a likewise model as real system measurements are not available. Results do not lead to required validation on assumptions and input variables.

3. Important results for the next chapters

(a) Assumptions

i. Process duration is simulated from the sampled initial period and the influences of rework. *It is essential to state that durations are described in days and not in hours*. Start and finish dates, define activity duration. How the work is completed within these days is not included in the model. So the model cannot be used to simulate the effect of hiring more or fewer employees. However, this could be a useful addition to the simulation.

ii. In the same line as process duration are process cost defined as the initial cost for an activity and the influence of rework. Also, here made hours are not taken into account.

(b) Use model to find answers to remaining research questions
In the remaining chapters the documentation flow simulation model is used to find answers on the remaining research questions:

i. *How can the impact of quality attributes be prioritized for a DTC-project?*

ii. *How can documentation sharing strategy be used to improve project execution?*

iii. *What potential improvement can be found by the proposed documentation sharing strategy?*

4

Case Study: Analysis of a DTC-project

Chapter 3 explained the development of a model for documentation flow analysis. In this chapter, this model is used to answer research question 4: 'How to use documentation flow analysis to improve the project execution of a DTC-project?'. One DTC-project is chosen for the case study. In this case study rework caused by documentation quality is studied by different scenarios. Documentation quality is defined as the quality of information storage and retrieval systems and it is measured by quality attributes. This chapter also results in a solution to the main research question.

Section 4.1 describes the required assumptions to simulate the DTC-project with the DSM-based discrete-event Monte Carlo simulation. Subsequently, project-specific data collection is discussed in 4.2. Model parameters are scaled in 4.3 and quality attribute ranking scenarios are defined in 4.4. Section 4.5 gives scenario results. Section 4.6 ranks quality attributes, whereafter rework is linked to documentation sharing strategy by 4.7. Then an alternative documentation sharing strategy is proposed, and potential improvements are identified in 4.8. The chapter ends with a chapter summary in 4.9.

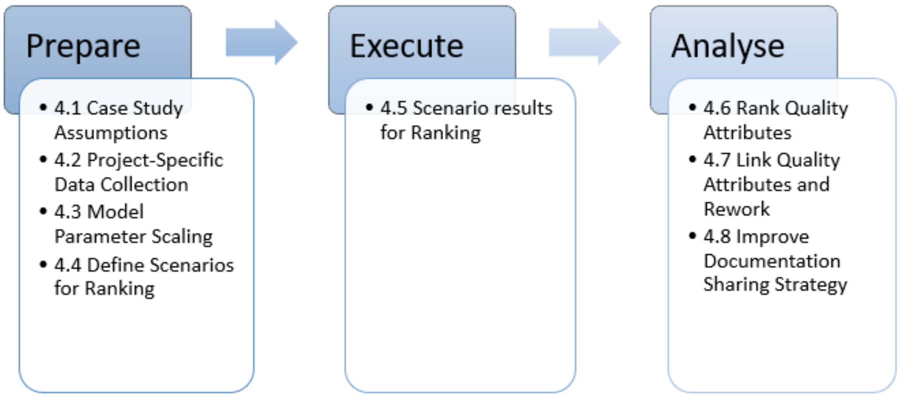


Figure 4.1: Structure of Chapter 4

Prepare

Defining, collecting and scaling of case study data are necessary tasks to prepare the case study. These tasks are captured by Sections 4.1-4.4. Define includes the definition of assumptions and scenarios for the simulation. Collect refers to the data collection and scale concerns parameter scaling.

4.1. Case study assumptions

Several assumptions are made in order to study the documentation flow for a one-off DTC-project. These assumptions can be categorised into two groups: assumptions for the DSM-based discrete-event Monte Carlo Simulation and assumptions for the application of case study results.

To start with assumptions in the first category; assumptions which make the DSM-based discrete-event Monte Carlo simulation suitable for the case study.

First of all, a fixed time step is used. The flexible time step as mentioned in Chapter 3 results in a minimal duration for the case study due to a large number of overlapping activities. The tiny time steps are a burden on computational time and are avoided by a fixed time step. Thus *a fixed time step of 1 day is used for the DTC-project*.

With lacking pure activity duration data (task duration without rework), a factor is used to eliminate rework from scheduled duration. In shipbuilding historical data, including rework, is used to generate project schedules. Decreasing initial duration with a fixed factor prevents the generation of an enormous amount of rework.

Task duration is expressed in days. Hours are not used to exclude an additional variable; the number of people working on a task. Expressing duration in hours is beneficial if duration can be linked directly to cost. However, budgeted hours and cost are not at the same detail level for the DTC-project. Next, different employees work on the same task at other wages. A wage difference in itself is not an issue, but no clear overview is present, which makes it hard to include cost. So no easy link can be made between production hours and cost. Thus the *cost is not fit for this case study*.

In the DSM-based discrete-event Monte Carlo simulation outliers are cause by a repeating sequence of rework on rework. In reality project managers and production managers can take action to prevent excessive delays. However, changing executing strategy once a project is started is not allowed by the simulation. Next, the number of people working on tasks is not included. Thus in reality, managers can assign more workers to a specific tasks when time is an issue but this is not incorporated in the model. Thus, the DSM-based discrete-event Monte Carlo simulation shows the impact if no steps are undertaken to prevent overruns.

To continue with assumptions of the second category: assumptions to the use of the results of this case study. First of all, it is assumed that the first built vessel is representative of the remaining ships. Next, it is assumed that the ship is produced under the integrated production procedure, which means that sections, blocks and zones are used. Furthermore, section and block outfitting is applied. Tasks are executed concurrently; however, yard capacity limits concurrency.

Despite the assumptions, the simulation outcome is still valuable. The main reason is the possibility to study the (relative) impact of quality attributes on project execution under the named assumptions. Also, the assumptions do not influence the links among activities. Finally, building under the integrated production procedure is often seen for DTC-projects.

4.2. Project-specific data collection

Chapter 3 discussed the collection of general data. However, to construct project-specific DSM matrices also project-specific data is required. Project schedules and building plans are used to retrieve this data. The collection of activity data and cost is discussed in Subsection 4.2.1. Subsection 4.2.2 defines the activity sequence and a limit on concurrency. Followed by the required information on dependencies among DSM Matrix building blocks to construct project-specific DSM matrices is explained in Subsection 4.2.3.

4.2.1. Activity duration

Initial task duration, which is expressed in days, is retrieved from the project schedule. For the DTC-project, which is analysis in this case study, result overruns in linear increasing contractual penalties with a gap. This linear behaviour implies that a the quadratic impact function as given by Equation 3.21 is no longer suitable to calculate the schedule risk of this project. Therefore the impact function is adjusted to an linear function, which is displayed in Equation 4.1.

4.2.2. DSM building blocks, activity sequence and concurrency limit

The algorithm to construct project-specific DSM matrices as explained in Section 3.4 requires input on building block numbers. This includes a number of: sections, blocks and zones. The algorithm also requires an activity sequence and a limit on concurrency.

The activity sequence is collected from the building strategy and the project schedule. It should be noted that the activity sequence is implemented at the detail level of DSM building blocks. Thus, the activities are ordered by the tasks named in Tables 3.2, 3.3 and 3.4. This detail level does not lead to complications, because the DSM-based discrete-event Monte Carlo simulation allows concurrency. Figure 4.2 illustrates the definition and detail level of the activity sequence. Numbers 1 to 7 define the sequence at the level of section and block production tasks. At a lower level, section production is described by ten activities, shown in the blue box. Also, ten activities are used to define block production more precisely, displayed in the grey box.

Subsequently a limit on concurrency is required. This data is project-specific since it depends on the project and the building location. The limit prevents the simulation to execute all production activities at the same time. The limited of concurrency is defined by yard capacity. Data is retrieved from project schedules. The collected data has a focus on the building yard as DTC's aim is to deliver according to client wishes. Tables 4.1, 4.2 and 4.3 provide normalized concurrency limit data for this case study.

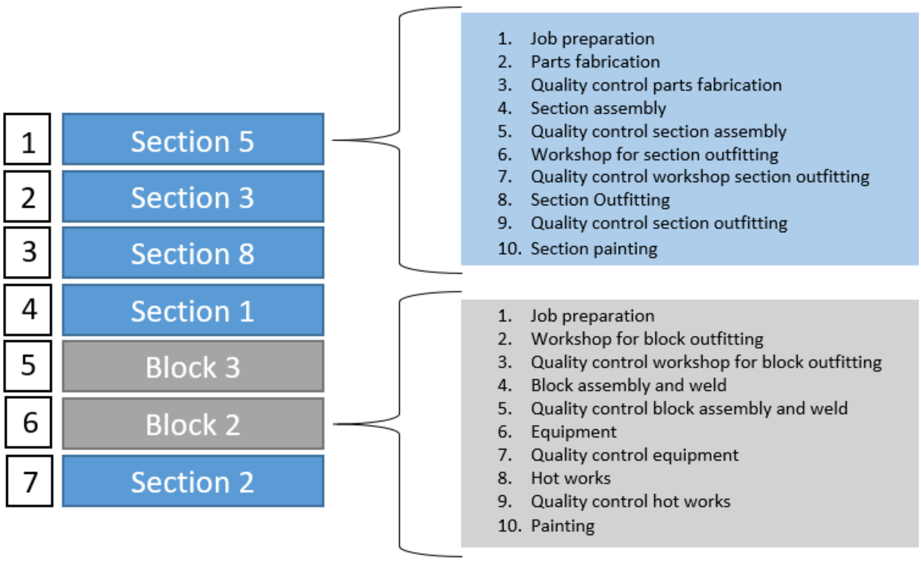


Figure 4.2: Detail level of the activity sequence explained. Used order is not linked to a specific project, it is random. This figure only serves as an example.

Table 4.1: Maximum activities executed in parallel for section production

Activities	**Max in Parallel**
Job Preparation Section X	14%
Parts Fabrication Section X	33%
Quality Control Parts Fabrication Section X	33.3%
Section X Assembly	67%
Quality Control Section X Assembly	67%
Workshop for Section X Outfitting	52%
Quality Control Workshop Section X Outfitting	52.4%
Section X Outfitting	100%
Quality Control Section X Outfitting	100%
Section X Painting	100%

Table 4.2: Maximum activities executed in parallel for block production

Activities	Max in Parallel
Job Preparation Block X	100%
Workshop for Block X Outfitting	100%
Quality Control Workshop for Block X Outfitting	100%
Block X Assembly and Weld	86%
Quality Control Block X Assembly and Weld	86%
Equipment Block X	100%
Quality Control Equipment Block X	100%
Hot Works Block X	100%
Quality Control Hot Works Block X	100%
Painting Block X	100%

Table 4.3: Maximum activities executed in parallel zone outfitting

Activities	Max in Parallel
Job Preparation Zone X	93%
Workshop for Zone X Outfitting	43%
Quality Control Workshop for Zone X Outfitting	43%
Hot Works Zone X	50%
Equipment Zone X	50%
Final Outfitting Zone X	50%
Quality Control Outfitting Zone X	50%

4.2.3. Links among DSM matrix building blocks

Dependencies among DSM Building Blocks are the last required project-specific input. These dependencies define the link between engineering packages and specific sections, blocks or zones. It also includes the links between blocks and sections and zones and blocks. All these data is stored in an additional matrix. Figure 4.3 illustrates the structure of this matrix.

All DSM Building blocks are placed at columns and rows of a large matrix. Simple '1' and '0' are used to represent links between batches. If a link is present, the '1' activates data stored in the Basic non-Project-Specific DSM structure. Thus, dependencies among DSM building blocks is vital information to generate project-specific DSM matrices. Different to the other matrices, a '1' at the diagonal is required. When this '1' is omitted the algorithm does not print all information.

Mostly project schedules are used to define these links.

4.3. Model parameter scaling

Parameter scaling is essential to obtain a useful simulation model. Various parameters need scaling: activity duration, schedule risk and rework probability. First, Subsection 4.3.1 defines initial task duration without rework by activity duration scaling. Scaled initial durations affect schedule risk. Subsection 4.3.2 discusses this impact in more detail. Subsequently, rework probabilities are scaled from a relative Task Volatility (TV) value to a probability value in Subsection 4.3.3.

4.3.1. Activity duration

In the DSM-based discrete-event Monte Carlo simulation, initial activity duration is used to sample activity durations for each run, taking uncertainty in account. During the simulation rework is generated an added to the initial duration. Browning and Eppinger (2002) underlines that this method only gives reasonable results when the pure activity duration is used. Pure activity duration is the duration without any rework.

In shipbuilding, historical data is used to create building schedules. Unfortunately, for this project, this contains duration data with rework. The frequent use of historical data makes it difficult to obtain pure activity duration in shipbuilding. Not using pure activity data in the DSM-based discrete-event

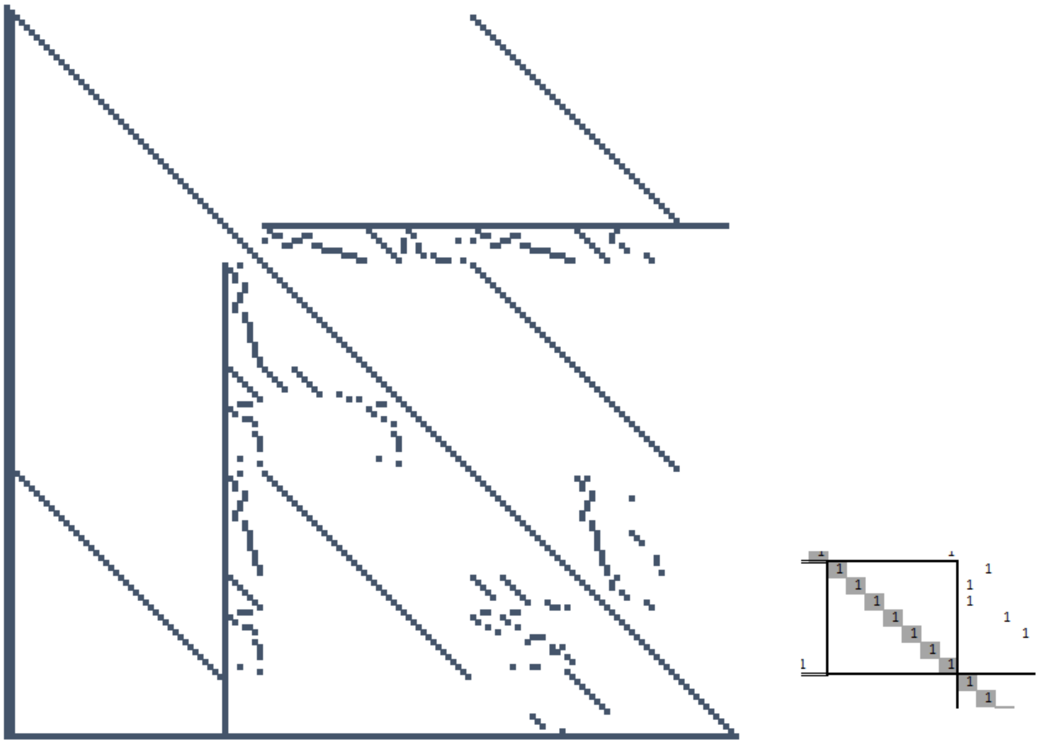


Figure 4.3: An example of the dependency structure among DSM-matrix building blocks. At the right is zoomed in on a part of this dependency matrix.

Monte Carlo simulation, results in to much rework. Figure 4.4 illustrates the pure duration, scheduled duration and extensive rework. In this figure, an arbitrary task is used; it only serves as an example.

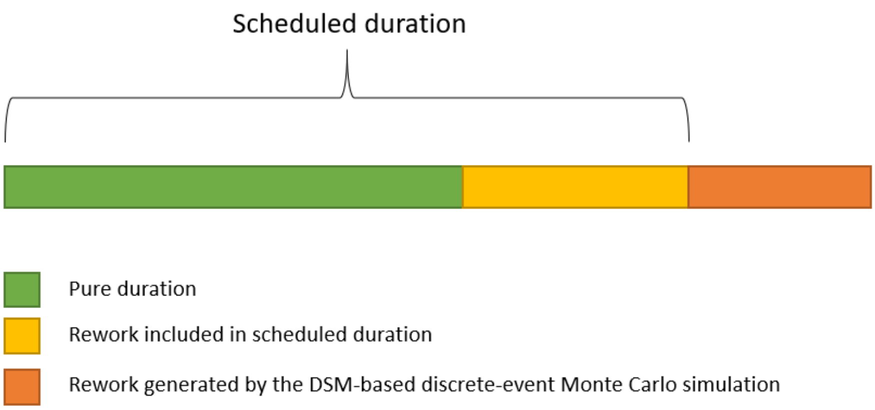


Figure 4.4: Pure activity duration, scheduled duration and extensive rework

The lacking information on pure activity duration also affects rework probability scaling since it requires a reasonable outcome compared to a scheduled duration. Therefore assumptions are necessary. According to Ciobanu and Neupane (2008) roughly 30% of rework present in shipbuilding projects. This number gives information on the entire project and not on specific tasks. Thus if no rework is included in the scheduled activity durations, project duration should be 70% of the planned project duration. A fixed factor is used to decrease the initial duration to ensure a project duration of 70% compared to the scheduled project duration.

Different factors are tested to find an appropriate value for $\boldsymbol{MLV_a}$. A factor of **0.75** results in a duration of 70.56% compared to the scheduled duration, tested with 1 000 runs. This outcome is within a 1% margin and sufficient for the use of the DSM-based discrete-event Monte Carlo simulation in this study.

4.3.2. Schedule risk

Schedule risk is a non-dimensional number that gives insight into the likelihood a project exceeds its deadline, as described in Subsection 3.5.4. Headlines of this subsection were the project-specific impact function and target value. Project-specific refers not only to data collection but also involves parameter scaling.

Exceeding deadlines for the used DTC-project for this case study results in linear contractual penalties with a cap. Therefore the quadratic impact function as mentioned by Browning (1998) in Equation 3.20 is adjusted. Based on the behaviour of the penalties, a linear impact function is used for this case study, given by Equation 4.1. The impact function is not capped, as large overruns create other negative consequences besides contractual penalties.

In Subsection 4.3.1 it is stated that initial duration scaling affects schedule risk. This impact concerns the target value. A process without any rework is the ultimate goal. Although this is not realistic, it is a good strive. Therefore the target value (T_s) is set on 70%. This value is in line with results of Ciobanu and Neupane (2008).

$$\text{Impact Function} = \kappa_s \cdot (\boldsymbol{S} - T_s) \tag{4.1}$$

κ_s : dimensionless constant; 1 is used

$\boldsymbol{S}$: Project duration vector

T_s : Target value (70%)

4.3.3. Rework probability

Rework probabilities need scaling in order to transform the relative Task Volatility (TV) values to absolute probability values. This transformation is project-specific. Subsection 3.3.2 describes the relative TV values on a scale from 1 to 9. Thus, these values need to be transferred to matching probability values for the DTC-project. Scaling is done by the procedure described by Yassine et al. (2001). Figure 4.5 illustrates the procedure.

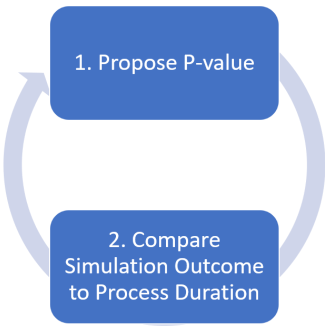


Figure 4.5: Procedure for rework probability scaling with a known mean process duration

1. **Run simulation with small P-value**
   A P-value of 10% is used as starting point
2. **Compare simulated duration to mean process duration**
   Mean process duration for the DTC-project is the scheduled duration. Process duration is normalised; 100% is the scheduled duration.
3. **Adjust P-value if simulated duration does not match mean process duration**
   A matching P-value is a value that results in a mean process duration that deviates less than 1%.

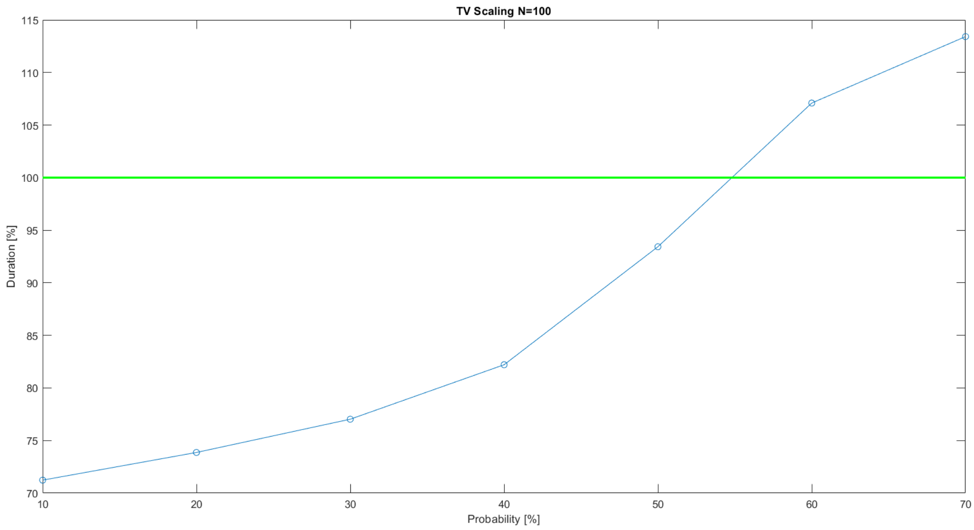


Figure 4.6: TV Scaling amount of runs N = 100

The scaling procedure requires a stable simulation output, defined by the stability criteria described in Subsection 3.4.2. With an increasing number of process runs, the output becomes more stable. On the other hand, more runs result in a longer computational time. For instance, simulating 1000 runs with the MATLAB parallel computing toolbox requires 3 hours of calculation time. Thus for scaling purposes a balance between computational time and accuracy has to be found.

Table 4.4: TV value scaling amount of runs N = 100, target: 100%

P-Value [%]	Mean [%]
0	70.24
10	71.23
20	73.87
30	77.03
40	82.22
50	93.43
60	107.1
70	113.4

The scaling procedure is split-up in two parts to reduce time. The first 100 runs are used to get a first impression of a suitable P-value. The number of runs is increased in a later stage. It is understood that 100 runs do not generate accurate results. However, it gives direction for a potential P-value. Results of the initial scaling are given by Table 4.4. From the results, it is likely to find a matching TV number in the range of 50%-60%. More runs are needed to determine a matching P-value with more certainty. Figure 4.6 visualises the data from Table 4.4. In this figure, the target value is represented by the green line, and simulation outcome is shown by the blue line.

A subsequent step uses 1000 runs. In this step not the entire range, 10%-70% is tested. Since potential is seen for a P-value in the range 50%-60%. Table 4.5 provides simulation outcome for P-values in the range 50%-60%. It is noticed that more runs lead to lower mean values. Figure 4.7 visualises the data of Table 4.5. In this figure, the target value is represented by the green line, and simulation outcome is shown by the blue line. It should be noted that, similar to Yassine et al. (2001), linear interpolation is used to construct this blue line. The simulation is only run for the P-values 50% and 60%. From the intersection point of the two lines is concluded that a P-value of 55% is suitable in the context of this case study. This P-value results in a mean process duration of 99.1 %, within the margin of 1%.

Table 4.5: TV value scaling amount of runs N = 1000, target mean: 100%

P-value [%]	Mean [%]
50	93
60	107
55	99,1

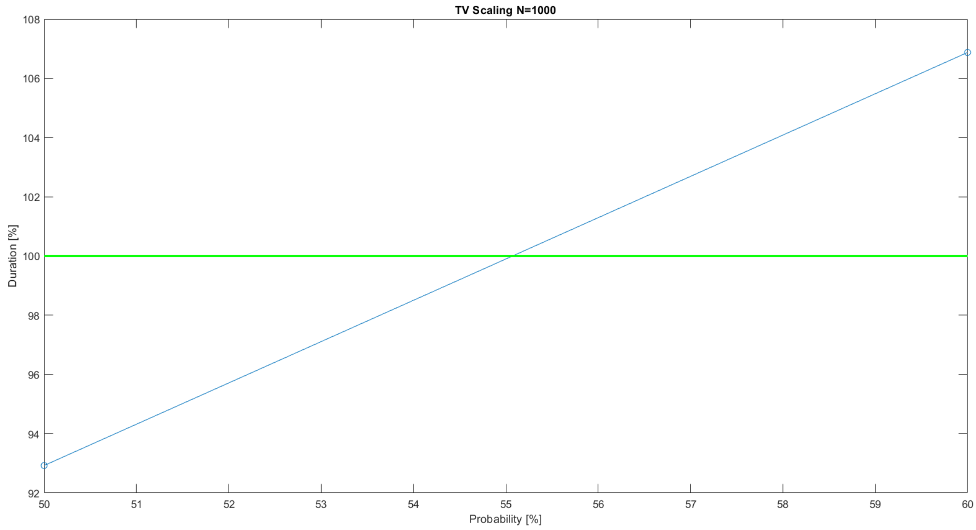


Figure 4.7: TV scaling amount of runs N = 1000

The last step is to determine the required number of runs to generate stable output distributions. From Equation 4.3 it is clear that 1000 runs do not result in a stable distribution. Thus more runs are required. The number of runs is increased with b (100 runs) until a stable distribution is found. Equations 4.4 and 4.5 show that a stable distribution is generated when 1100 runs are used. For a P-value of 55% a mean value of 99.28% is obtained, within the 1% margin.

$$D_{mean}(1000) = 0.0015 < 0.01 \Rightarrow \text{Pass} \quad (4.2)$$

$$D_{var}(1000) = 0.017 \nless 0.01 \Rightarrow \text{Fail} \quad (4.3)$$

$$D_{mean}(1100) = 0.0019 < 0.01 \Rightarrow \text{Pass} \quad (4.4)$$

$$D_{var}(1100) = 0.0017 < 0.01 \Rightarrow \text{Pass} \quad (4.5)$$

4.4. Define scenarios for quality attribute ranking

Six scenarios are defined to study the effect of project documentation quality on the project execution of DTC-projects. Five scenarios are based on the by Table 1.1 named causes of rework. In addition, a baseline scenario is defined in Subsection 4.4.1; a scenario without any rework. Remaining scenarios are explained in Subsections 4.4.2-4.4.6.

4.4.1. Scenario 0: No rework - Baseline

Simulating a scenario without rework sets a baseline. Although this is not a realistic scenario, it contributes to quality attribute ranking. Including this scenario makes it possible to compare other scenario outcomes to a baseline. Besides, it serves as the ultimate goal of process optimisation.

4.4.2. Scenario 1: Delayed documentation - Waiting

Late deliverables affect depending tasks. Tasks either need to wait for deliverables or can start with assumptions. This scenario simulates a worst-case situation to study the effect of waiting; all planned durations are increased with **5%**. Further, all activities are forced to wait for the delayed deliverables. Also no adjustments are made to DSM matrices.

Figure 4.8 gives an overview of the used simulation rules. In this figure, the orange part of Task A represents the delay, and the waiting time of Task B. Task B is started once Task A is completed. Delays in Task B are not included in this figure but are present in the scenario.

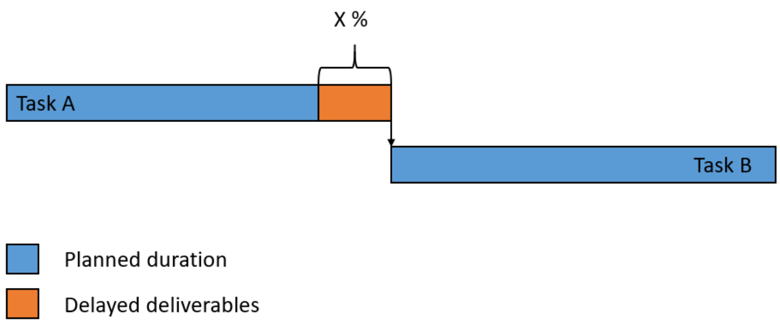


Figure 4.8: Scenario 1: Delayed documentation - Waiting; subsequent activities are forced to wait on documents

4.4.3. Scenario 2: Delayed documentation - Assumptions

Using assumptions to start activities early is an alternative approach to deal with delayed deliverables. Delays are similar to scenario 1 to keep a fair comparison. Thus again, all initial activity durations are delayed by **5%**. In contrast to scenario 1, activities are no longer forced to wait for delayed activities. Following activities are allowed to start if **95%** of an activity that it depends on is completed. Finalised documentation is sent when the activity is **100%** fulfilled. Additional documentation can cause rework on the already started activities.

Figure 4.9 gives an overview of the different documentation flows. The figure shows two documentation flows from task A to task B. The yellow part of task B represents the early start based on assumptions. Again delays on task B are not displayed in this figure but are included in the scenario.

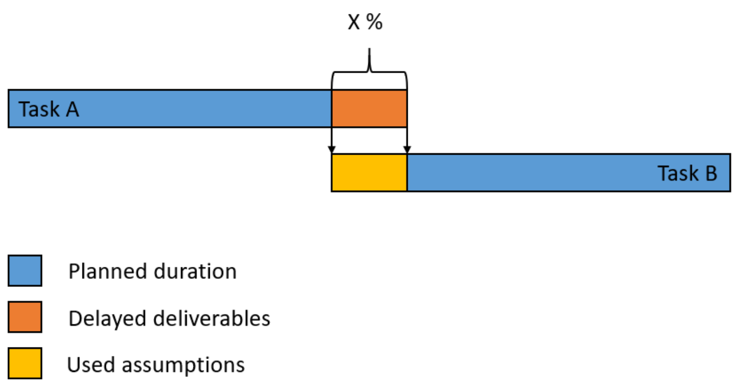


Figure 4.9: Scenario 2: Delayed documents - Assumptions; subsequent activities uses assumptions to start activity

In contrast to the waiting scenario, using assumptions requires adjustments to the DSM-based discrete-event Monte Carlo simulation. Adjustments are listed below.

1. **Define early start subsequent activities**
   The set of active activities must be expanded in order to allow activities to start early. In the

original simulation, activities are forced to wait on required documentation. Under a scenario that uses assumptions to start activities early the active set is defined by all activities that satisfy the below-listed constraints.

(a) Activities that have work remaining; and

(b) Activities that are not dependent on activities that are not for 95% completed OR Activities are dependent but at least 95% of the activity where it depends on is completed.

2. **Include update information stream**
Assumptions are based on primarily information and updated once the final documentation is available. This means that for each activity two information flows are present when assumptions are used to start an activity early. Two documentation flows creates input changes for activities, which can cause second-order rework.

4.4.4. Scenario 3: Poor communication

Poor communication creates confusion. For this scenario it is assumed that confusion leads to a request for clarification at the depending task. It is understood that poor communication can also lead to an increase in mistakes. However, poor communication here is limited to a request for clarification. This is different to late deliverables with assumptions as an activity can create first-order rework before it is finished. Poor communication can cause rework on both task A and task B.

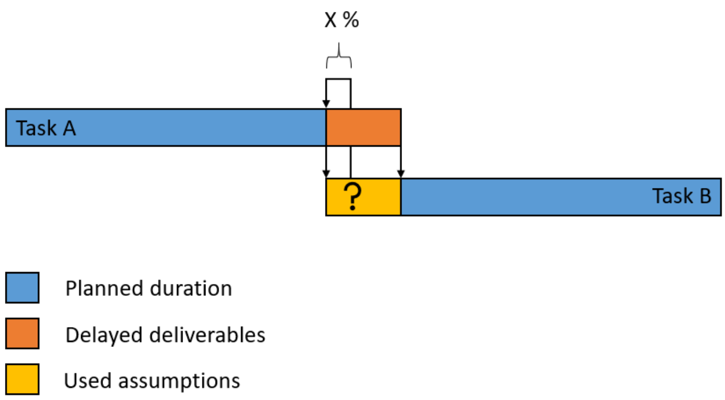


Figure 4.10: Scenario 3: Poor communication, request for clarification

Figure 4.10 illustrates this scenario. In this figure, task A is completed once it shares documentation with Task B for the first time. However, this documentation is not clear to task B, and task B asks for clarification at task A. Providing task B with additional documentation creates rework for task A. On the side, Task B has already started with assumptions. Once the clarified documentation is received, it also can generate rework on task B. A request for clarification does not occur for all tasks in the project; a random number generates occurrence to prevent endless rework loops. The number of requests for clarifications on specific tasks is capped at 1 to prevent endless rework loops.

4.4.5. Scenario 4: Input changes

Input changes are defined as changes in input information once a project has started. For the DTC-project input, changes are implemented as changes of basic/detailed engineering information during the execution of the project. The effect on basic engineering activities is kept limited to prevent endless rework loops. In reality, first-order rework-rework on the engineering task-on basic engineering depends on the type of change orders. Nevertheless, it is seen that second-order rework-rework on all subsequent tasks- has a more considerable impact on project execution. Thus by this simplification, the most significant effect is included.

DSM matrices are adjusted to implement this scenario in the DSM-based discrete-event Monte Carlo simulation. This adjustment concerns upper diagonal entities in all matrices. Links between detailed- and basic engineering are added. Also, connections between production tasks and engineering tasks are added. By this method, the below-listed tasks are sensitive to input changes.

- Detailed Engineering
- Job Preparation (at section, block and zone level)
  Rules and Regulations, Carpentry information
- Section assembly
  Arrangements, Carpentry
- Section outfitting
  Arrangements, Carpentry
- Block assembly
  Arrangements
- Block Outfitting (Including Equipment and Hot works)
  Construction
- Zone Outfitting
  Arrangements
- Hull assembly
  Arrangements and Construction

4.4.6. Scenario 5: Mistakes

Mistakes in deliverables create errors in subsequent tasks. These mistakes can be identified at various stages, and cause rework on multiple tasks. Late discoveries affect a project more than early findings. A random number implements mistake discovery in the DSM-based discrete-event Monte Carlo simulation.

Figure 4.11 illustrates the main points of this scenario. In this figure, different line colours are used for arrows. Black arrows represent the initial path without rework. The orange arrow represents the discovery of a mistake in Task's A documentation. The feedback from task C to task A creates rework on task A which is called first-order rework. Task B and C used the documentation of Task A; thus, these tasks are affected by the mistake and experience also rework. This rework is called 'second-order' rework and represented by yellow lines and blocks. All combined, the error in the documentation of Task A discovered at the end of task C causes rework on all three tasks. When the first-order path becomes more prolonged, more activities are affected, and the mistake has more impact on the project. In this example, all activities suffer from the error; however, in the DSM-based discrete-event Monte Carlo simulation, this is decided by a random number. Thus first- and second-order rework is not predefined but includes uncertainty.

Adjustments to the DSM-based discrete-event Monte Carlo simulation are needed to run this scenario. The rework probability matrix DSM_1 is partly adjusted to increase the likelihood of mistakes in the documentation. All upper diagonal entities are increased by 1 on the TV scale. TV values are already scaled for the DTC-project. Thus a transformation scheme from TV values to probability values already exists. Other input values are not changed.

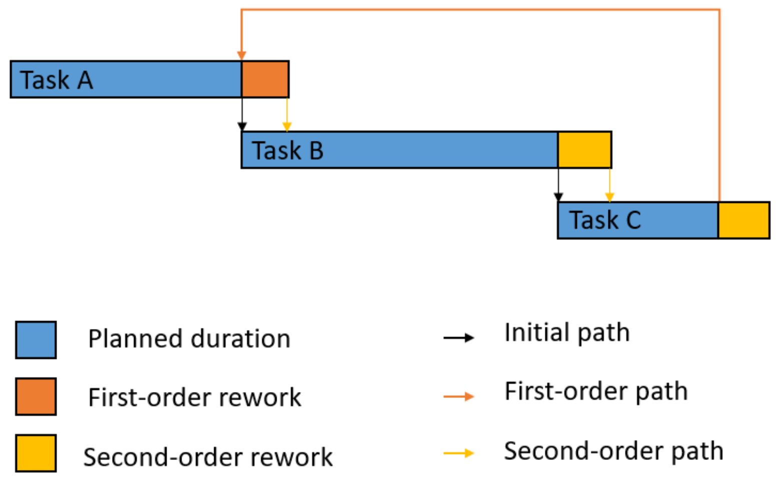


Figure 4.11: Scenario 5: Mistakes

4.5. Scenario results for quality attribute ranking

This section deals with the execution part of this chapter. The DSM-based discrete-event Monte Carlo simulation is run for the different scenarios defined in Section 4.4. The effect of the different scenarios on project execution is discussed. The influences are expressed by duration distribution graphs, schedule risk, mean, standard deviation and the likelihood for an unacceptable outcome. First, the baseline, a scenario without rework is set in Subsection 4.5.1. Then the effect of other scenarios is handled in Subsections 4.5.2-4.5.6. Results are the basis of quality attribute ranking discussed in the last part of this chapter, analysis.

4.5.1. Scenario 0: No rework - Baseline

Figure 4.12 reports the outcome of a scenario without rework. In this figure, both relative and cumulative probability are displayed. The vertical red line represents the mean value. The histogram uses a bin width of 1% and reports the highest chance on an outcome in the range 69-70%.

Also, no outliers exceeding 100% are mentioned. Thus, chance on an unacceptable outcome is equal to zero. Schedule risk is related to the likelihood for an unacceptable outcome and is therefore also equal to zero. The simulated duration is the most optimal, and achieving this duration is not realistic. However, it is seen as the ultimate goal of process optimisation.

Since no rework is present differences between runs are small. Thus, it is expected to find a lower number of runs compared to situations with rework. At first, 800 runs are used to generate distribution plots for process duration. However, from Equations 4.6 and 4.7 it is made clear that 800 runs does not result in a stable distribution. Therefore the number of runs is increased to 900. Equations 4.8 and 4.9 show that outcome complies with the stability criteria for 900 runs. Although no rework is present, deviations in process duration occur due to uncertainty in initial activity duration. Figure 4.12 illustrates the distribution of process duration for this scenario.

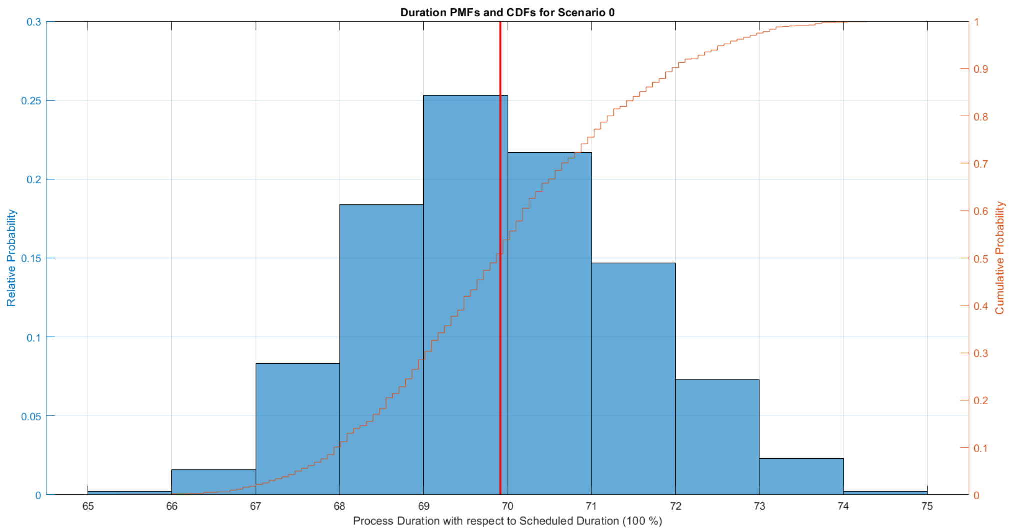


Figure 4.12: Results scenario 0: No rework, N = 900, mean = 69.91%, standard deviation 1.52%, bin width = 1

$$D_{mean}(800) = 6.16 * 10^{-4} < 0.01 \Rightarrow \text{Pass} \quad (4.6)$$

$$D_{var}(800) = 0.0143 \nless 0.01 \Rightarrow \text{Fail} \quad (4.7)$$

$$D_{mean}(900) = 7.15 * 10^{-6} < 0.01 \Rightarrow \text{Pass} \quad (4.8)$$

$$D_{var}(900) = 0.0093 < 0.01 \Rightarrow \text{Pass} \quad (4.9)$$

Since 900 runs give stable output for a process without rework, it is expected that other scenarios require at least 900 runs to generate stable output distributions. Thus all remaining scenarios are simulated with 900 runs first. More runs are used if stability criteria require so.

4.5.2. Scenario 1: Delayed documentation - Waiting

A shift to the right is expected for a scenario that forces activities to wait on delayed deliverables compared to the situation without rework. Figure 4.13 illustrates simulation outcome, which complies with the stated expectation. Again both relative and cumulative probability are shown, and the red vertical line represents the mean value. The histogram uses a bin width of 5% and reports the highest likelihood on an outcome in the range 90-95%. The distribution reports outliers that exceed 100%. Thus schedule risk and the possibility of an unacceptable outcome are no longer equal to zero.

Waiting on delayed deliverables results for 44% in an unacceptable outcome with a schedule risk of 111.49. The wider spread compared to a situation without rework is in line with an increased standard deviation.

It is expected that this scenario requires more runs to generate stable outcome distributions than scenario 0. From the stability criteria in Equations 4.10 and 4.11 is concluded that 1100 runs lead to a stable outcome for this scenario. Thus the number of runs is in line with expectations.

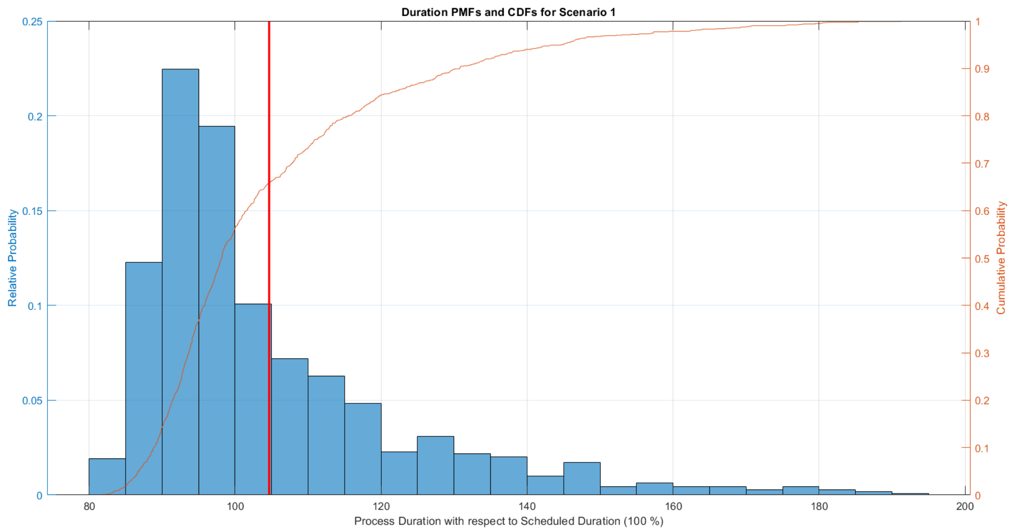


Figure 4.13: Results scenario 1: Delayed documentation - Waiting, N = 1100, mean 104.67%, standard deviation 18.4%, bin width = 5%

$$D_{mean}(1100) = 0.0014 < 0.01 \Rightarrow \text{Pass} \quad (4.10)$$

$$D_{var}(1100) = 0.0076 < 0.01 \Rightarrow \text{Pass} \quad (4.11)$$

4.5.3. Scenario 2: Delayed documentation - Assumptions

Figure 4.14 reports a signifiant change in project duration distribution when assumptions are used to start activities early. Similar to the other figures, both relative and cumulative probability are displayed, and the red vertical line represents the mean value. A bin width of 3% is used to construct the histogram.

The distribution is characterised by two peaks; one around 87% and around 116%. These peaks illustrate that using assumptions either has a positive effect or a negative effect. When assumptions are close to actual values, rework after receiving final information is limited. Even though the likelihood of an unacceptable process outcome increases compared to waiting, schedule risk is decreased. The decrease is explained by less severe outliers compared to waiting.

The reduction in outliers is also seen in standard deviation; a decrease is noticed compared to scenario 1. Although the reduced standard deviation no difference is seen for required runs. From Equations 4.12 and 4.13 is concluded that 1100 runs are required to create stable outcome distributions.

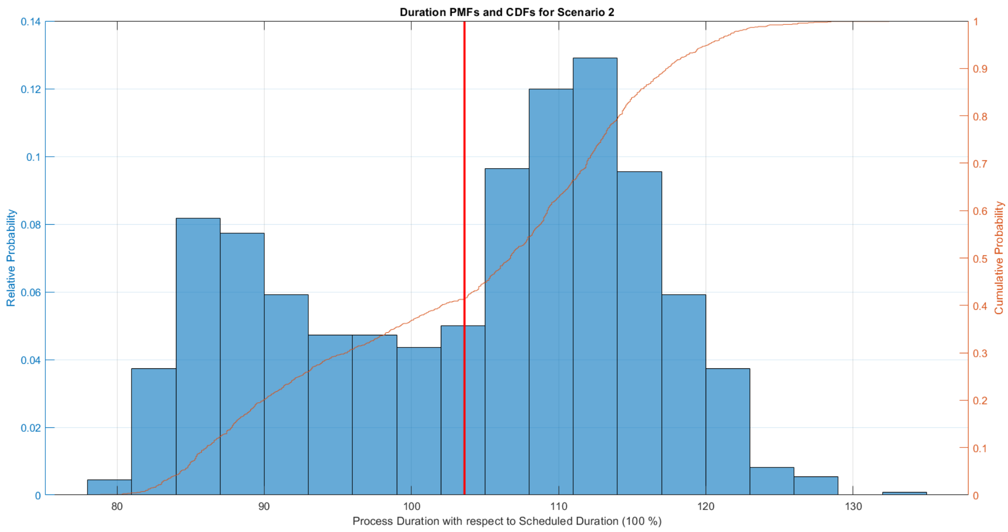


Figure 4.14: Results scenario 2: Delayed documentation - Assumptions, N = 1100, mean = 103.61%, Standard Deviation = 11.78%, bin width = 3%

$$D_{mean}(1100) = 0.0013 < 0.01 \Rightarrow \text{Pass} \tag{4.12}$$

$$D_{var}(1100) = 0.0013 < 0.01 \Rightarrow \text{Pass} \tag{4.13}$$

4.5.4. Scenario 3: Poor communication

The project duration distribution under poor communication is presented by Figure 4.15 and looks like the distribution for waiting as displayed in Figure 4.12. Again, both relative and cumulative probability are shown, and the red vertical line is the mean value. Further, a bin width of 5% is used to construct the histogram.

Outliers are less extensive in case of poor communication than waiting because the initial duration of activities is not delayed, with 5% for poor communication. The large chance for an outcome in the range of 85%-110% marks this scenario. The assumption that requests for clarifications are mainly used in the starting phase of activity explains this characteristic.

Although poor communication generates less extensive outliers than waiting for delayed deliverables, a more considerable schedule risk is found. Schedule risk depends on the number of runs that exceed 100% and the severity of this exceedance. Thus, this combination is less beneficial for poor communication than for waiting.

No difference is seen in the required number of runs to get a stable outcome. From the stability criteria given by Equations 4.14 and 4.15 is concluded that 1100 runs are sufficient to achieve a stable outcome.

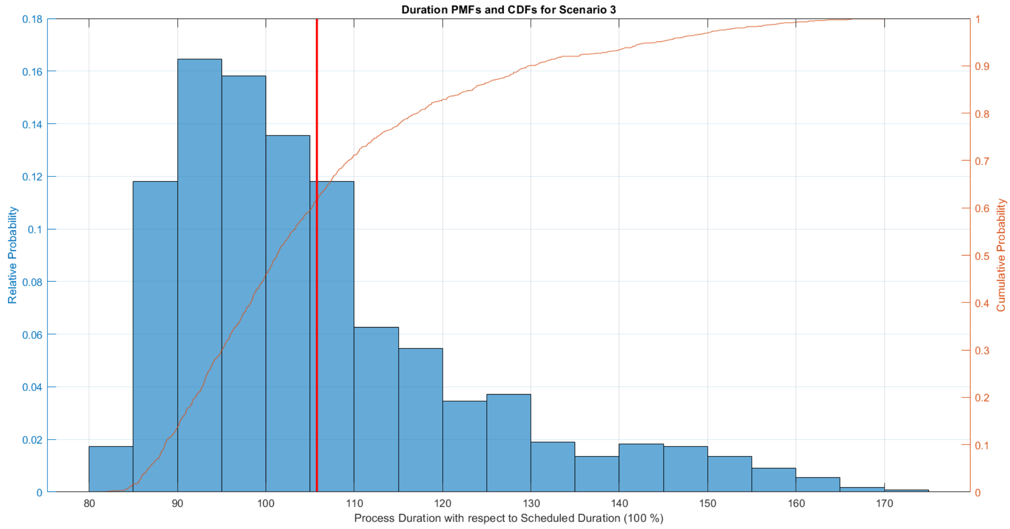


Figure 4.15: Results scenario 3 Poor communication, N = 1100, mean = 105.78%, standard deviation 17.00%, bin width = 5%

$$D_{mean}(1100) = 0.0017 < 0.01 \Rightarrow \text{Pass} \quad (4.14)$$

$$D_{var}(1100) = 0.0070 < 0.01 \Rightarrow \text{Pass} \quad (4.15)$$

4.5.5. Scenario 4: Input changes

Input changes affect process duration. The duration distribution is displayed in Figure 4.16. Similar to the other distribution plots, both relative and cumulative distribution is shown, and the red vertical line represents mean value. The histogram uses a bin width of 5%. Distribution shape is quite similar to the distribution of scenario 1, waiting, in Figure 4.13. Thus it is most likely that input changes alone result in a process duration below 100%. However, some activities are sensitive to input changes and cause massive overruns.

It is most likely to find a project duration in the range 90-95%. This characteristic keeps the likelihood of an unacceptable outcome under the 50%. However, large outliers cause a considerable schedule risk of 134.35. The presence of large outliers is in line with the large standard deviation of 19.80%. The large outliers and standard deviation tell that some activities are more sensitive to input changes than others. It should be mentioned that results depend on included possible input changes as defined by Subsection 4.4.5. From the stability criteria given by Equations 4.16 and 4.17 900 runs are sufficient to generate stable outcome distributions.

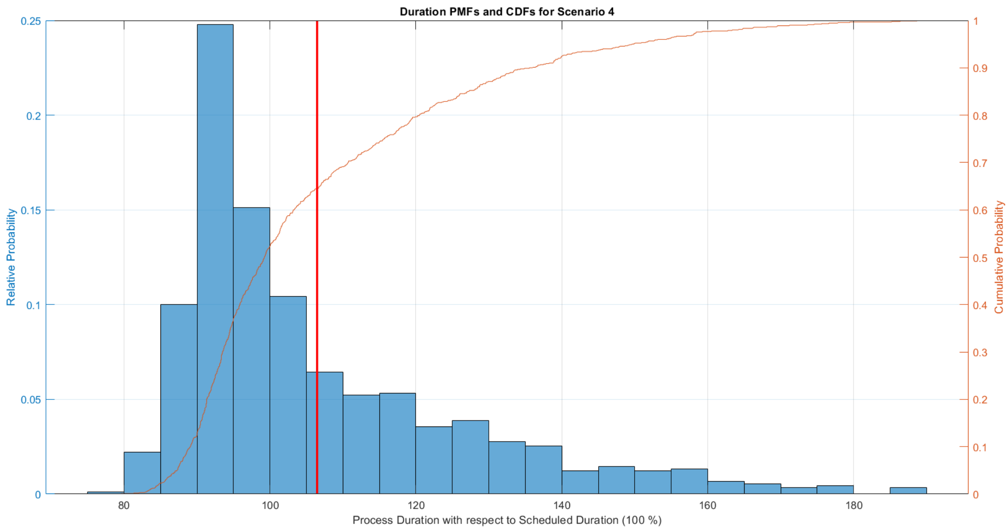


Figure 4.16: Results scenario 4: input Changes N = 900, mean = 106.46%, standard deviation 19.80%, bin width = 5%

$$D_{mean}(900) = 0.0036 < 0.01 \Rightarrow \text{Pass} \tag{4.16}$$

$$D_{var}(900) = 0.0097 < 0.01 \Rightarrow \text{Pass} \tag{4.17}$$

4.5.6. Scenario 5: Mistakes

Increased mistakes in deliverables cause a different distribution shape than the earlier discussed scenarios. Figure 4.17 shows duration distribution, relative probability, cumulative probability, and mean value (red vertical line). Compared to the other scenario outcomes, a more considerable shift to the right is noticed. The shift is explained by the partly increased rework probabilities as defined by Subsection 4.4.6. The adjusted rework probabilities cause a higher likelihood for first-order rework. As a result, *mistakes do almost always lead to a process outcome that exceeds the target value*.

Mistakes almost always exceed the target value of 100%. Thus, it is expected to find enormous values for the probability of unacceptable process outcome, schedule risk and mean. Obtained results comply with this expectation. Increased mistakes in deliverables result in a large probability of an unacceptable project outcome, namely 90.86%. Also, a substantial schedule risk is found; 229.91. Furthermore, the distribution reports a large mean value of 117.21%. The stability criteria in Equations 4.18 and 4.19 show that 1200 runs are required to obtain stable output distributions.

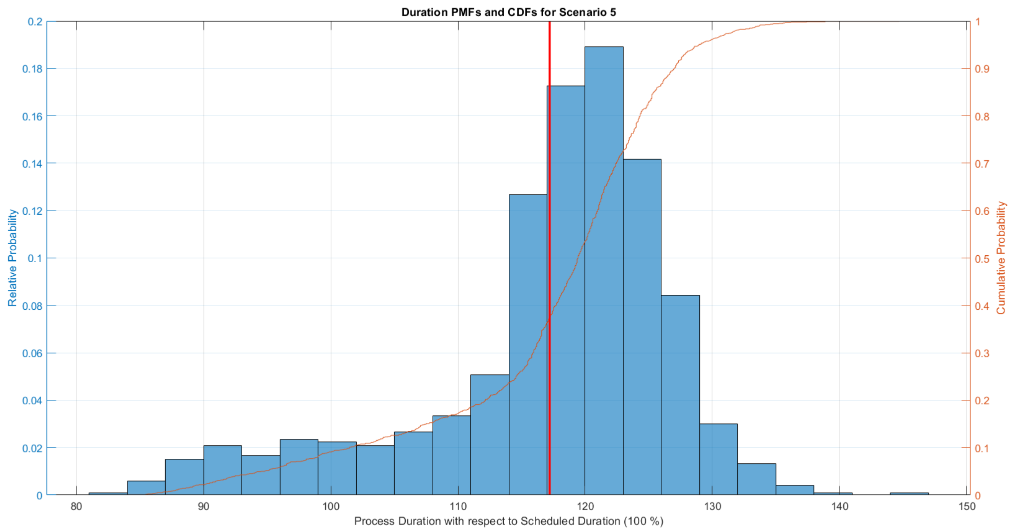


Figure 4.17: Results scenario 5: Mistakes N = 1200, mean = 117.21%, standard deviation 9.9%, bin width = 3%

$$D_{mean}(1200) = 3.78 * 10^{-4} < 0.01 \Rightarrow \text{Pass} \tag{4.18}$$
$$D_{var}(1200) = 0.0083 < 0.01 \Rightarrow \text{Pass} \tag{4.19}$$

Figure 4.18 gives an overview of the distribution plots of the different scenarios. The figure shows an approximate project duration of 70% for a situation without rework (scenario 0). This scenario has an approximated duration below 100% due to the use of historical data for shipbuilding planning purposes. The DSM-based discrete-event Monte Carlo simulation requires pure activity duration (without rework). As described in Subsection 4.3.1, a factor is used to decrease initial task duration to obtain pure durations. Also, all scenarios are plotted at the same bin width (7%) to make comparison easier. Therefore, bin width differs from earlier displayed graphs.

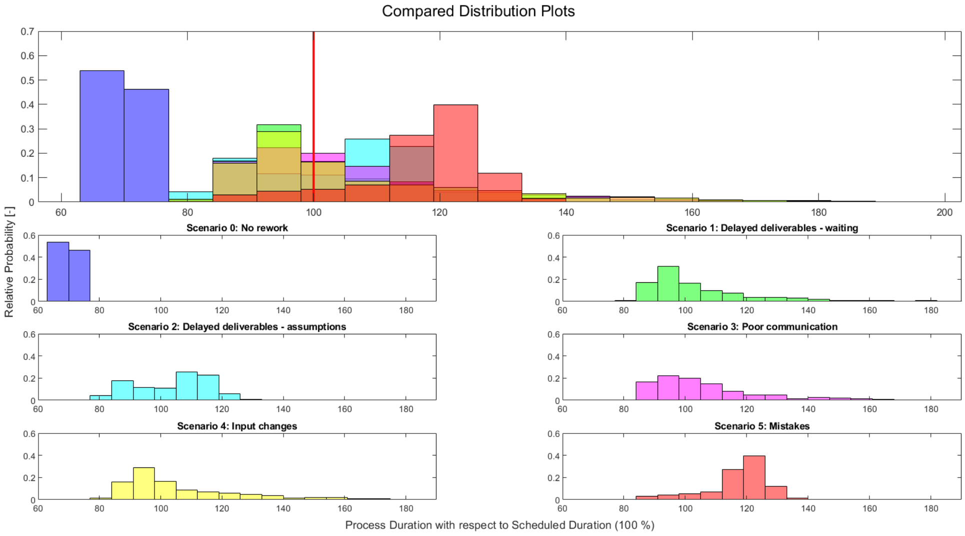


Figure 4.18: Compared distributions of multiple scenarios. An equal bin width of 7% is used.

Analysis of results

Analysis of case study results is the last task remaining in this chapter. The analysis aims to define a rank for quality attributes and to get insight into the effect of specific quality attributes on the execution of a DTC-project. Section 4.6 presents the ranking and Section 4.7 states valuable insights from detailed impact analysis. This insight is used in Section 4.8 to propose an alternative documentation sharing strategy that improves the execution phase.

Box plots visualises summarised statistical data for data sets and are therefore used to compare different datasets. In this case study, box plots are used to analyse data; an example is given by Figure 4.19. Data is generated randomly and is not related to the case study. Percentiles are used to divide data in proportions smaller and larger than a certain number. The median, the red horizontal line in the blue box, is the 50th percentile. Thus, half of the data lay below this line and half above. The boundaries of the blue box are defined by 25th and 75th percentile. For the 25th and 75th percentile, the data is no longer divided in equal groups. The distance between 25th- and 75th percentile is called the interquartile range (IQR). This range is used to define outliers. Data that exceeds 1.5 times the IQR distance from 75th or 25th percentile is called an outlier and is marked with a red cross. It should be mentioned that it is also possible to have outliers below the lower adjacent. However, these are not included in Figure 4.19. When a minimum or maximum does not exceed the 1.5 times IQR range, the minimum or maximum determines the lower or upper adjacent.

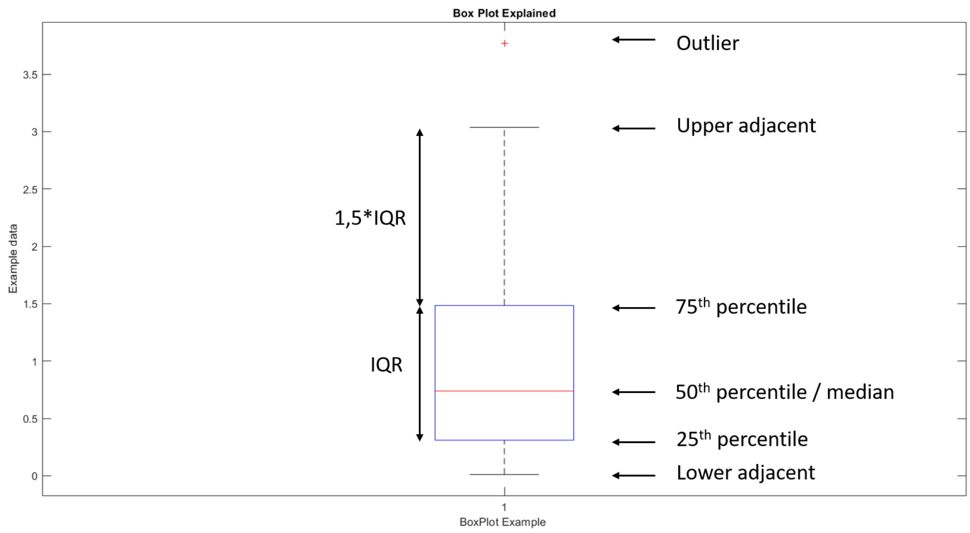


Figure 4.19: Box plot explained

4.6. Rank documentation quality attributes

Scenario results, as presented in Section 4.5, are the basis for quality attribute ranking. The various scenarios are compared by mean, median, standard deviation, the likelihood of an unacceptable outcome and schedule risk. Finally, an overall order is made.

Figure 4.20 gives the duration distributions of the various scenarios in a box plot. Table 4.6 provides more detailed information related to this box plot. The box plot gives a clear overview of the presence of outliers. Most scenarios have outliers; only scenario 0 and 2 report no outliers. Most outlies are seen for scenario 5. Most outliers are not a direct representation of reality, as they represent cases generating rework on rework. It is not helpful to filter out outliers nor to set a strict limit on rework for this study since the effect of rework differs from task to task. Thus limiting rework, for example, at 300% of the original duration, does not have the same impact on all activities. Rework on activities with a shorter duration can have less impact than on longer activities. This phenomenon leads to outliers in the model; yet, the model can still be used for quality attribute ranking. Outliers can be prevented, if managers take action.

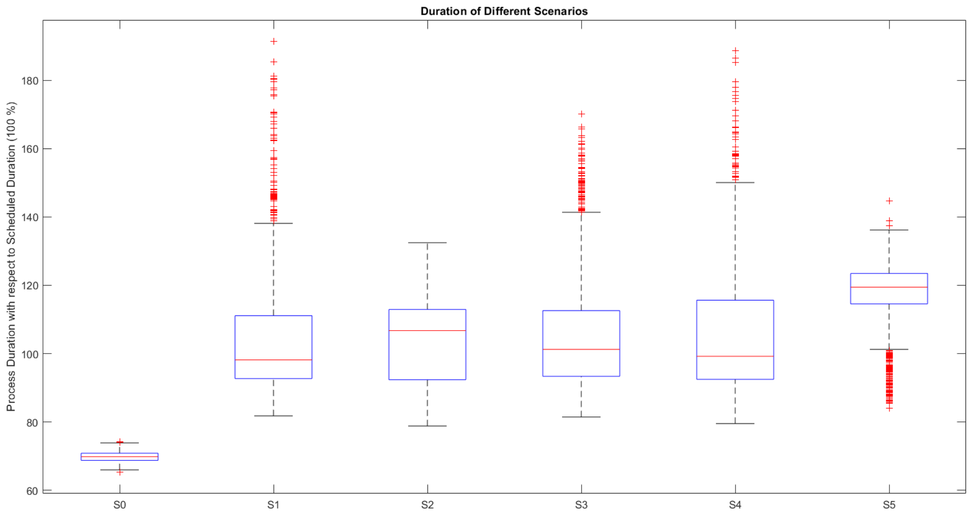


Figure 4.20: Durations of Different Scenarios Compared

Table 4.6: Numerical data to the different scenarios

Scenario	Min	Max	Mean	Median	25th	75th	Std	P_un	Schedule Risk [-]	Outliers [#]
0	65.4%	74.3%	69.9%	69.9%	68.8%	70.9%	1.52%	0%	0	3
1	81.8%	191.3%	104.7%	98.2%	92.7%	111.1%	18.4%	43.6%	111.5	70
2	78.9%	132.5%	103.6%	106.7%	92.4%	112.9%	11.8%	63.1%	94.4	0
3	81.5%	170.0%	105.8%	101.2%	93.4%	112.6%	17.0%	54.0%	117.8	67
4	79.6%	188.7%	106.5%	99.2%	92.5%	115.7%	19.8%	47.6%	134.4	43
5	84.0%	144.7%	117.2%	119.4%	114.6%	123.5%	9.9%	90.9%	229.9	119

Based on the median values in Figure 4.20, scenario 5 has the most impact on project execution. Table 4.6 reports a large likelihood for an unacceptable outcome (90.9%) for this scenario. This table also gives a large mean value (117.2%).

Scenario 4, input changes, also shows serious impact on execution. Figure 4.20 reports massive outliers. Table 4.6 reports a large maximum value for scenario 4 (188.7%) and a large value for the 75th percentile (115.7%). According to the results in Table 4.6 scenario 4 has a large mean value (106.5%), large standard deviation (19.8%) and a large schedule risk (134.4).

It should be noticed that the presence of outliers influences mean and schedule risk. Not all outliers give a direct representation of reality, which should be taken into account when determining a ranking. The standard deviation is a measure for the spread of the data, which is enormous. A large standard deviation can indicate that scenario outcome differs for different circumstances. *Some activities are more sensitive to input changes than other activities*.

Table 4.7: Score table for documentation quality attribute ranking

Scenario	Mean	Median	25th	75th	Std	P_un	Schedule Risk	Total	Rank
0	6	6	6	6	6	6	6	42	**6**
1	4	4	4	5	2	5	4	28	**4**
2	5	5	5	3	4	2	5	29	**5**
3	3	3	3	4	3	3	3	22	**3**
4	2	2	2	2	1	4	2	15	**2**
5	1	1	1	1	5	1	1	11	**1**

A scoring table is used to rank the quality attributes. In Table 4.7, scenarios are ranked on mean, median, 25th percentile, 75th percentile, standard deviation (Std), the probability for an unacceptable outcome (P_{un}) and schedule risk. In this table, each scenario is ranked on the criteria. For each criteria, the highest rank (1) is assigned to the scenario with the highest value. Eventually, all ranks are summed for each scenario and displayed in the *'Total'* column. The scenario that scores the lowest ranks highest (1). The scenarios represent documentation quality attributes as defined in Table 2.1. Thus, with a scenario ranking, also a documentation quality attribute ranking is found.

Based on the scoring table, scenarios for quality attributes are placed in the following ranking:

1. **Mistakes** - Scenario 5
   Accuracy
2. **Input Changes** - Scenario 4
   Certainty
3. **Poor Communication** - Scenario 3
   Clarity, Conformance, Coordination and Legibility
4. **Delayed Deliverables - Waiting** - Scenario 1
   Timeliness
5. **Delayed Deliverables - Assumptions** - Scenario 2
   Completeness

4.7. Documentation quality attributes and rework

In this section, a detailed analysis is conducted to study the relationship between documentation quality and rework. Insights are used to propose an alternative documentation sharing strategy in Section 4.8. For this analysis, the data of each scenario is equal divided into four groups. The groups make it possible to study the impact of rework on specific tasks for the variety in scenario outcome for project duration. The groups are based on statistical numbers, also used in the box plot: the minimum, 25th percentile, 50th percentile/median, 75th percentile and the maximum. Table 4.8 gives an overview of the different groups.

Detailed analysis is time-consuming; thus, it is limited to a set of activities. Subsection 4.7.1 describes the use of interface criticalities to determine the set of activities. Subsections 4.7.2-4.7.6 discuss the analysis for each scenario.

Table 4.8: Grouped box plot data is used for the detailed analysis

Groups	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Group 1 [min-25th]	82%-93%	79%-92%	81%-93%	80%-92%	84%-115%
Group 2 [25th-50th]	93%-98%	92%-107%	93%-101%	92%-99%	115%-119%
Group 3 [50th-75th]	98%-111%	107%-113%	101%-113%	99%-115.65%	119%-123%
Group 4 [75th-max]	111%-191%	113%-132%	113%-170%	116%-189%	123%-145%

4.7.1. Interface criticality

Browning and Eppinger (2002) state that interface criticalities can be used to identify activities activities that drive schedule risk. An overview of dependencies that result in a high interface criticality give insight into the relation between documentation structure and rework.

Because the DTC-project consists out of many tasks (1890), a three-dimensional bar plot is unsuitable to present the interface criticalities. Therefore, it is chosen to keep values numerical and not visual. General tasks can be found in Tables 3.2-3.4. Based on the interface criticalities a list of relevant activities for the detailed analysis is defined.

1. All detailed engineering tasks
2. Section production tasks:
   (a) Workshop for section outfitting
   (b) Section outfitting
3. Block production tasks:
   (a) Workshop for block outfitting
   (b) Block outfitting: hot works
   (c) Block outfitting: equipment
4. Zone outfitting tasks:
   (a) Hot works
   (b) Equipment
   (c) Finalise

The in Subsection 4.7.1 selected set of activities is analysed in more detail. The analysis aims to get an understanding of the outcome differences reported for the various scenarios in Figure 4.20. This understanding gives direction for potential adjustments to the documentation sharing strategy. Subsections 4.7.2-4.7.6 summarise the results of the analysis, but a full description is available in Appendix D. Both discuss the results by Detailed Engineering, Section Production, Block Production and Zone Outfitting.

4.7.2. Scenario 1: Delayed documentation - Waiting

Table 4.8 reports a widespread for the data of scenario 1, with a minimum value of 81.8% and a maximum value of 191.3%. In Section 4.6, it is mentioned that some activities are more sensitive to mistakes than others, which can explain this spread. This subsection aims to identify sensitive activities and to define the circumstances that cause enormous overruns.

Waiting on delayed documentation during detailed engineering affect project duration. Rework on the below-listed engineering activities have the most impact on the overall project duration.

- **Construction drawings**
- **Mechanical diagrams**

Other activities also affect project duration, but their impact is less severe. This concerns the following activities:

- **Hull outfitting drawings**
- **Superstructure outfitting drawings**,

Little rework is reported for the section production tasks. Thus *it is not expected that rework on section production has a large impact on outcome differences for a situation where activities are forced to wait on delayed documentation*. However, in general more rework is reported for **workshop for section outfitting** than for **section outfitting**.

From all tasks, the most rework is reported for block production. Block outfitting equipment and block outfitting hot works suffer most from delayed documentation. The results also show rework for workshop for outfitting, but this does not contribute to the variation in project duration.

Zone outfitting also is sensitive for delayed deliverables. Rework on hot works during zone outfitting has the most influence on the variations. Rework on the equipment during zone outfitting and rework on remaining outfitting tasks also has influence, but less than rework on hot works.

4.7.3. Scenario 2: Delayed documentation - Assumptions

Figure 4.20 also reports variation for a scenario using assumptions to start activities early. The figure shows a large median value, but no outliers. Thus using assumptions is beneficial when overruns are expected. This subsection aims to identify sensitive activities that cause outcome differences. Also, it is identified when using assumptions becomes beneficial.

Of all detailed engineering activities, carpentry drawings, affect project outcome most in a situation that uses assumptions to start activities early. Carpentry drawings includes 2D drawings of the accommodations, and floor and wall plans. The below-listed detail engineering tasks also affect project outcome, but their impact is lower than of the carpentry drawings.

- **Hull outfitting drawings**
- **Superstructure outfitting drawings**

Section production tasks have limited influence on project outcome under this scenario. That does not mean that no rework occurs for section production, but that rework on these tasks does not contribute to the variation in project duration.

Block production activities have more impact on project duration than the previously discussed tasks. Rework on the equipment, and hot works during block outfitting affect the outcome the most. Workshop for block outfitting has limited influence on the variation. Thus, using assumptions to start activities early cause a small increase in rework for workshop for block outfitting, but the increase is large for the equipment and hot works during block outfitting.

All zone outfitting tasks included in the detailed analysis affect project outcome. Thus, hot works, the equipment and completion of remaining outfitting tasks are sensitive tasks for assumptions.

4.7.4. Scenario 3: Poor communication

Project outcome under poor communication also shows variation in Figure 4.20. The distribution has a median of 101.24% and reports outliers. This subsection aims to name activities that contribute to the shown outcome variation.

Detailed engineering tasks affect project outcome under poor communication. The impact is not equal for all tasks; rework on some tasks affect project duration more than rework on other tasks. In the detailed analysis, the most impact is seen for the below-listed tasks.

- **Construction drawings**
- **Mechanical diagrams**

By this list, it is not meant that other detailed engineering activities do not suffer from poor communication. However, rework caused by poor communication on other activities is not related to the outcome variations. For example, substantial rework is reported for hull outfitting drawings, but this rework is seen for both shot and long durations.

Again, rework on section production tasks have little contribution to the outcome differences. Two reasons explain this. First, little rework is reported compared to other activities. Second, the little rework does not change for short or long durations. Thus, short or long durations are not caused by rework on section production. However, this does not mean that section production in itself does not suffer from poor communication.

Rework on block production influences the different project outcomes, but impact differs for the activities. The detailed analysis report very little rework for workshop for block outfitting. Little rework on the related detailed engineering tasks explains this finding. Rework on the equipment, and hot works during block outfitting do influence project outcome, as the rework increases for projects with a longer duration. Thus, the equipment and hot works during block outfitting are sensitive to poor communication.

Rework on all zone outfitting tasks, included in the detailed analysis, affect project outcome. Here, the most impact is seen for zone outfitting hot works. Zone outfitting equipment also has a notable effect. Least impact is measured for zone outfitting finalising space.

To summarize, ***care is required to communicate the following documentation:***

- Construction drawings

- Mechanical diagrams
- Block outfitting equipment
- Block outfitting hot works
- Zone outfit hot works
- Zone outfit equipment
- Zone outfitting finalize space

Especially, mechanical diagrams and construction drawings require attention, due to the relative massive impact (10%) on block outfitting. Hot works during zone outfitting also has a feedback loop with a relatively large effect (6%) to the equipment for block outfitting.

4.7.5. Scenario 4: Input changes

Large outliers are reported for scenario 3 in Figure 4.20. Table 4.6 presents a widespread for scenario 3 with a minimum value of 79.6% and a maximum of 188.7%. This subsection aims to identify tasks that contribute to this large spread.

Input changes affect the detailed engineering tasks by the generation of rework. Here, two influence levels are identified: sensitive and less sensitive. Sensitive activities contribute to all output differences; less sensitive activities contribute only to output differences in the range minimum-median. The sensitive activities are below listed.

- **Superstructure outfitting drawings**
- **Mechanical arrangement drawings**

The less sensitive activities also are listed:

- **Construction drawings**
- **Mechanical diagrams**

Remaining detailed engineering activities can generate rework due to input changes, but this rework does not contribute to output differences. However, it harms project execution.

Again, no large influence is seen for section production tasks. Rework values also are here limited and rework does not change for smaller or larger durations. Thus, rework on section production due to input changes is not the primary driver of deviations in project duration.

Similar to other scenarios, project outcome is affected by rework on block production. Rework on equipment and hot works has the most impact on project duration. Compared to rework on equipment and hot work has rework on workshop for block outfitting less effect. The difference is explained by rework on the required documentation. There is more rework on outfitting documentation (related to the equipment and hot works) than on workshop documentation. Placing equipment and hot works during block outfitting also is affected by rework from subsequent tasks such as zone outfitting.

The project outcome is influenced by all zone outfitting tasks that are included in the detailed analysis. Rework on hot works during zone outfitting has the most impact. Rework on the equipment during zone outfitting also has a notable effect. This finding is in line with gained insight from interviews with site managers. Site managers state that input changes have a severe impact on hot works and related activities. Besides, the moment of zone outfitting in the project timeline explains the enormous impact; zone outfitting is one of the last stages. Thus changes affect a large number of tasks.

Input changes have a severe impact on project execution, especially when it concerns block outfitting equipment, block outfitting hot works, zone outfitting hot works and zone outfitting equipment.

4.7.6. Scenario 5: Mistakes

According to the ranking in Section 4.6, mistakes caused by documentation affect project execution the most. Figure 4.20 reports the largest median value for scenario 4. Outliers below the lower adjacent are unique for this scenario. From Figure 4.20, it is seen that project duration varies. This subsection aims to identify activities that contribute to outcome differences for a scenario with mistakes.

The project outcome is affected by mistakes in detailed engineering activities, but not all activities influence the project outcome. Impact of activities that has an influence is limited. Rework on detailed engineering activities defines whether it is possible to find a project duration below or above the median. However, rework on detailed engineering tasks do not influence the severity of the overruns. Thus, other activities (other than detailed engineering) are responsible for massive overruns.

Rework on section production tasks is limited, but a minor impact is seen on workshop for section outfitting for some sections. Although reported values are small, workshop for section outfitting can contribute to the variations in project duration. Nevertheless, it is expected that the variety is influenced more by other activities.

Mistakes create the most rework on block production tasks, and this rework contributes to a variation in the project outcome. Rework on the equipment during block outfitting affects the result of the project. Rework on the hot works during block outfitting also has a considerable influence, but it is limited to the 75th percentile. Thus, rework on the hot works during block outfitting contribute to the variations in project duration, but not for project durations that exceed the 75th percentile.

Zone outfitting contributes to the variations in project duration. Significantly, rework on hot works during zone outfitting affects project duration. Rework on the equipment during zone outfitting also influences project outcome, but the impact is limited to the 75th percentile. Thus rework on equipment does not cause project overruns larger than the 75th percentile.

To summarise, *construction drawings, mechanical diagrams, and block and zone outfitting suffer most from mistakes in documentation*. This outcome is reasonable since outfitting tasks have a large number of dependencies and are time-consuming. Outfitting activities are located at the end of the process, which gives less freedom to find creative solutions once a mistake is identified then early in the process. This outcome stresses the importance of correct project documentation used for the outfitting phase.

4.8. Improve documentation sharing strategy

Earlier documentation quality attributes are ranked by different scenarios in Section 4.6. The relation between documentation quality attributes and rework is analysed by Section 4.7. This section aims to improve project execution by proposing an alternative documentation sharing strategy. Improvement is achieved when the negative of the documentation quality attributes is reduced. Ranked scenarios and their related documentation quality attributes are below-listed. Highest rank is given to the scenario and quality attributes that affect project duration most.

1. **Mistakes**
   Accuracy
2. **Input changes**
   Certainty
3. **Poor communication**
   Clarity, Conformance, Coordination and Legibility
4. **Delayed documentation - Waiting**
   Timeliness
5. **Delayed documentation - Assumptions**
   Completeness

Although all ranked items relate to documentation quality, not all risks are reduced by the documentation sharing strategy alone. For some quality attributes, activities are identified that lead to the highest impact on project outcome. Monitoring and actively keep an eye on the progress these tasks can reduce risk. For other documentation quality attributes, adjustments to the documentation sharing strategy are made to reduce risk. Both risk-reducing measures are discussed according to the ranked order.

Mistakes in the project documentation harms project execution. This outcome is not surprising, referring to the gained insights from various interviews during this study. Of course, it is preferred to avoid mistakes in documentation. However, we do not live in an ideal world; mistakes are human. Fortunately, some mistakes in documentation do not lead to high overruns immediately. Tasks that have great impact on project duration need monitoring to reduce the risk of mistakes in documentation. For these tasks,

1. Construction drawings;
2. Mechanical drawings;
3. Block outfitting; and
4. Zone outfitting

it should be monitored if rework values come close to threshold values given by Figure 4.21. When threshold values are exceeded, large overruns do happen. Thus, when rework comes near the threshold value action is required to prevent massive overruns on project duration. Figure 4.21 does not name all project tasks; however, it does not mean that mistakes in documentation for other activities do not harm project execution. It does mean that rework on these activities have less impact on project duration. With this insight, managers can spend their limited time on activities with the highest impact.

Detailed engineering:	**Detailed engineering:**	**Detailed engineering:**
Rework < 1.125 times scheduled duration	**Rework < 1.5 times scheduled duration**	**Rework > 1.5 times scheduled duration**
⇒ Mechanical diagrams	⇒ Mechanical Diagrams	⇒ Mechanical Diagrams
⇒ Construction drawings	⇒ Construction Drawings	⇒ Construction Drawings
Block production	**Block production**	**Block production**
⇒ **Equipment**	⇒ **Equipment**	⇒ **Equipment**
Rework < 0.525 times scheduled duration	Rework < 1 times scheduled duration	Rework > 1 times scheduled duration
⇒ **Hot works**	⇒ **Hot works**	⇒ **Hot works**
Rework < 0.4 times scheduled duration	Rework < 0.8 times scheduled duration	Rework > 0.8 times scheduled duration
Zone outfitting	**Zone outfitting**	**Zone outfitting**
⇒ **Equipment**	⇒ **Equipment**	⇒ **Equipment**
Rework < 0.35 times scheduled duration	Rework < 0.5 times scheduled duration	Rework > 0.5 times scheduled duration
⇒ **Hot works**	⇒ **Hot works**	⇒ **Hot works**
Rework < 0.4 times scheduled duration	Rework < 0.7 times scheduled duration	Rework > 0.7 times scheduled duration
⇒ **Finalizing**	⇒ **Finalizing**	⇒ **Finalizing**
Rework < 0.25 times scheduled duration	Rework < 0.4 times scheduled duration	Rework > 0.4 times scheduled duration

Figure 4.21: Rework boundaries on activities to prevent large overruns due to mistakes in documentation

Input changes also have an notable impact on project execution. Especially, input changes that concern the equipment and hot works during block outfitting and zone outfitting tasks harm project execution. Other tasks that affect project duration, when rework occurs are:

1. Superstructure outfitting drawings
2. Mechanical arrangement drawings
3. Construction drawings
4. Mechanical drawings

Again this is not surprising. However, a variation is seen in project duration under a scenario with input changes. The variation means that some input changes harm project execution more than others. Activities that have great impact on project outcome, require monitoring during project execution. Further research also can gain more detailed insight into the effect of input changes. The DSM-based discrete-event Monte Carlo simulation focusses on documentation quality-related rework. It does not separate first-order rework (rework caused by feedback) and second-order rework (rework as a consequence of first-order rework). It is expected that a point in the project timeline exists where after input changes have an enormous impact on project execution. By some adjustments to the current model, it is possible to identify this point, but this requires further research. It is advised to separate first-order and second-order rework as a first step.

Poor communication also affects project execution. Here, poor communication is defined as a change that a receiver of documentation does not understand the documentation and has to call for a request for clarification. From analysis is concluded that some activities are more sensitive to poor communication than others. The below-listed activities affect project duration significantly:

1. Construction drawings
2. Mechanical diagrams
3. Block outfitting equipment
4. Block outfitting hot works
5. Zone outfitting

This list does not mean that other activities that are not listed, do not sensitive to poor communication. It represents activities that should get special attention of managers during project execution. Since a fast-paced environment characterises engineer-to-order projects, the list serves as a guideline to make sure the limited time is spent on the right activities. Since rework on these activities makes it difficult to complete a project within the deadline.

Both **waiting** on delayed documentation and the use of **assumptions** to start activities early can harm project execution. However, at the moment, no clear answer is present when one should use assumptions to start early or when one should wait on the completed documentation. Results from the detailed analysis form the basis for an alternative documentation sharing strategy. This strategy aims to combine best of two worlds for the waiting and assumption scenarios. According to Table 4.8 scenario 1 has large outliers in group 4, where scenario 2 has a better performance in group 4. Waiting on delayed deliverables caused the highest outliers, whereas no outliers are reported for the use of assumptions to start activities early. On the other side, using assumptions to start early has a high median value where the median value for waiting is way lower. The alternative documentation sharing strategy combines the benefits of the two scenarios by:

1. **forcing activities to wait for delayed documentation.**
   An delay of 5% is implemented for all activities.
2. **using assumptions to start activities early when an overrun of 11% is forecasted**
   Activities can start if 95% of a dependent activity is completed.

An improvement is expected in terms of reduced schedule risk, median and the likelihood for an unacceptable outcome for the alternative strategy compared to the earlier discussed strategies. However, no significant differences are expected for the mean value. The proposed alternative strategy is run in the DSM-based discrete-event Monte Carlo Simulation for a DTC-project.

Figure 4.22 illustrates the duration distribution for the alternative strategy. The shape of this distribution is a combination of the shapes shown in Figures 4.13 and 4.14. It can be seen that an outcome below 100% is most likely, as a pique is present for the range 90%-95%. Duration distribution of the alternative, waiting, and assumption strategy are plotted in Figure 4.24. Figure 4.23 shows three box plots: one for the alternative strategy, one for the situation where activities are forced to wait and one for a situation using assumptions to start activities early. Figure 4.23 gives a clear overview of overruns, median, 25th and 75th percentile values under the different approaches. It can be seen that outliers are reduced for the alternative strategy compared to a situation where activities are forced to wait for documentation. However, reduced outliers are not the only improvement for the alternative approach.

Moreover, waiting on documentation results in a smaller duration in the category 25th percentile - median compared to using assumptions to start early. Comparing the alternative strategy with using assumption to start early is seen that the alternative strategy has outliers and using assumptions not.

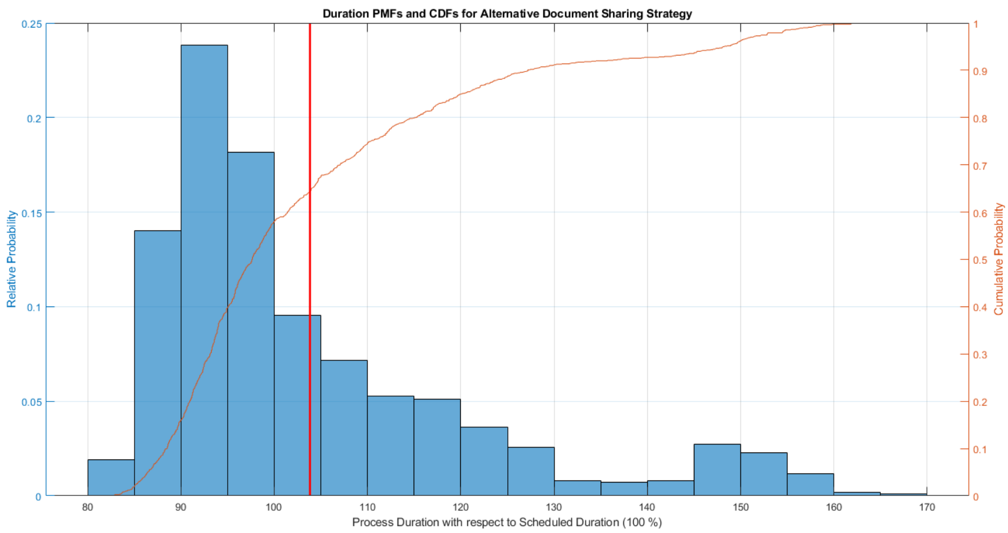


Figure 4.22: Histogram of the duration distribution under the alternative strategy, a bin with of 5% is used

Although it can look like using assumptions is a better strategy concerning duration, these outliers must be placed in perspective. Extreme outliers are caused by rework-loops. In real-life managers will take action to avoid such extensive outliers. This possibility is not incorporated in the model. Thus the model outcome is conservative. So the findings are in line with the earlier named expectations.

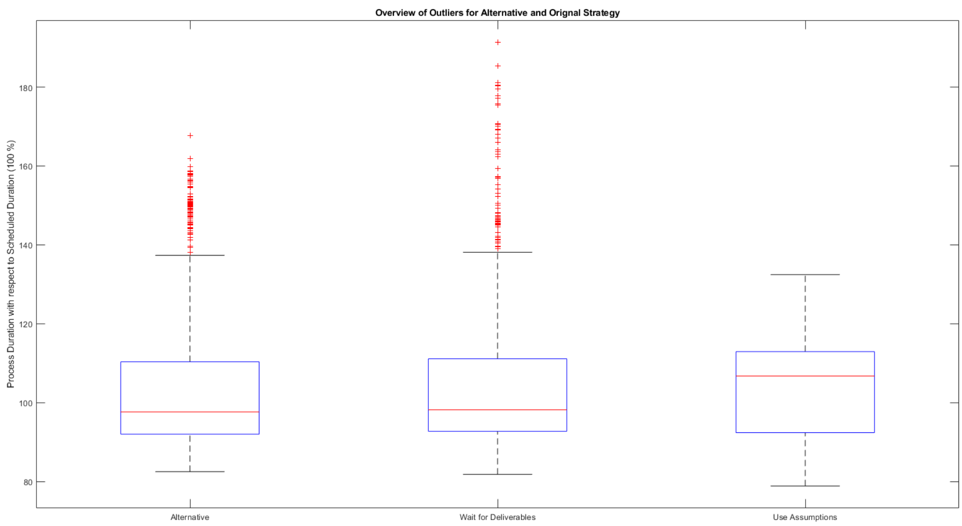


Figure 4.23: Box plot of the alternative, waiting and assumptions strategy

Table 4.9: Numerical results of alternative and original documentation sharing strategies

Scenario	Mean	Median	25th	75th	Std	P_un	Schedule Risk
Alternative	103.8%	97.6%	92.7%	110.8%	17.4%	41.9%	437.5
Waiting	104.6%	98.2%	92.7%	111.12%	18.4%	43.6%	448.3
Assumptions	103.6%	106.7%	92.4%	112.9%	11.8%	63.09%	434.6

Table 4.9 gives the numerical results of the alternative strategy, waiting on delayed deliverables and

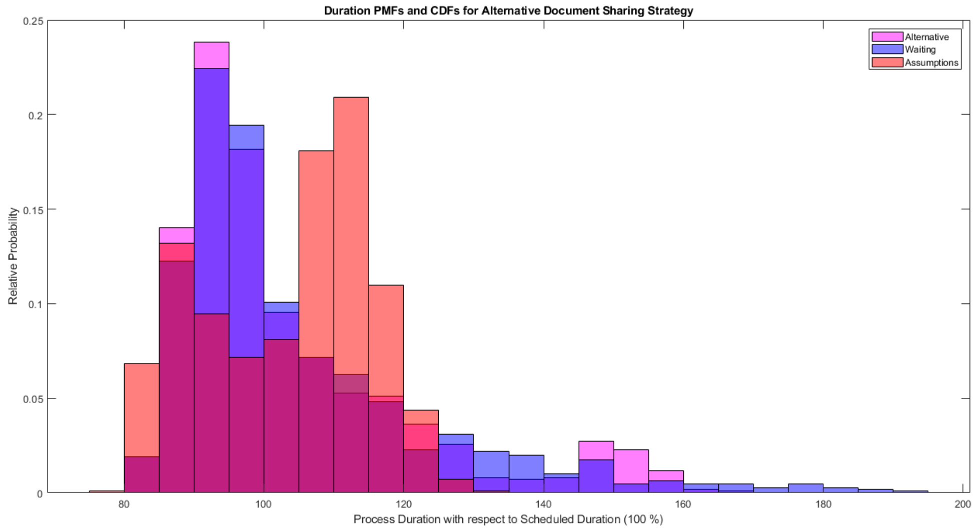


Figure 4.24: Result alternative documentation sharing strategy compared to waiting and assumptions strategy

using assumptions. Lowest schedule risk is found for a situation where assumptions are used to start activities early. In all cases, all activities are delayed by 5%, to create a worst-case scenario. One could think that using assumptions in this environment is the best strategy based on schedule risk. However, in 3.6.2 is described that schedule risk is a good indicator for risk but that the DSM-based discrete-event Monte Carlo Simulation is not suitable for close schedule risk comparison. Thus, strategies must not be examined on schedule risk alone. The likelihood for an unacceptable outcome, median, 25th, and 75th percentile values also need to be considered. A scoring table is used to test the different strategies for these measures.

Table 4.10: Score table to rank alternative strategy

Scenario	Mean	Median	25^{th}	75^{th}	Std	P_un	Schedule Risk	Total	Rank
Alternative	2	1	1	1	2	1	2	10	**1**
Waiting	3	2	1	2	3	2	3	16	**3**
Assumptions	1	3	3	3	1	3	1	15	**2**

Table 4.10 displays this scoring table, which is similar to the scoring table used in Section 4.6. For each measure, scores are assigned to the strategies, strategies that have least impact score highest (1). Scores are summed for each strategy and displayed in the *'Total'* column. The strategy with the lowest total score receives the highest rank (1). Highest ranked strategy deals best with the circumstances.

The data in Table 4.10 shows that the alternative strategy ranks best. Although, the alternative strategy does not score best on all measures. Outliers cause 2nd ranking on mean, median, standard deviation and schedule risk. Further, both the alternative and the waiting strategy score best on the 25th percentile. Until a project overrun of 11% is reached, the alternative strategy is similar to the waiting strategy. Because the 25th percentile is below 111%, equal values are found for the 25th percentile.

To conclude, the alternative strategy scores best on the measures. Assumptions score second, but the height of the red horizontal line in Figure 4.23 and the numerical data in Table 4.6 makes clear that assumptions must be avoided for project durations that do not exceed 111%. Waiting for documentation scores third due to its massive overruns.

4.9. Chapter summary

1. *How can the impact of documentation quality attributes be prioritized for a DTC-project?*

   (a) **Accuracy**
   Mistakes

   (b) **Certainty**
   Input changes

   (c) **Clarity, Conformance, Coordination and Legibility**
   Poor communication

   (d) **Timeliness**
   Delayed documentation - Waiting

   (e) **Completeness**
   Delayed documentation - Assumptions

2. *How can documentation sharing strategy be used to improve project execution?*
   Quality attributes affect project execution. From the detailed analysis is concluded that rework on the below-listed activities affect project duration the most.

   - Construction drawings
   - Mechanical diagrams
   - Block outfitting equipment
   - Block outfitting hot works
   - Zone outfit hot works
   - Zone outfit equipment
   - Zone outfitting finalize space

   Depending on the scenario, tasks suffers more or less from rework.

   Methods to use documentation sharing strategy to reduce the adverse effect on project execution is named per scenario in their ranked order.

   (a) **Mistakes**
   Figure 4.21 gives threshold values for rework to prevent massive overruns when mistakes occur. Project managers can use this list to spend their limited time on activities with large impact. The list also gives information when project managers should take action.

   (b) **Input changes**
   Input changes that affect block and zone outfitting should be avoided, as these changes have a large impact on project duration. Additional research is required to determine a clear boundary when to allowed or not to allowed input changes.

   (c) **Poor communication**
   Coordinating documentation that is related to the activities that have, in case of rework, the most impact on project duration requires attention. Documentation should be monitored and it should be checked if the documentation is well understood by the receiver, to prevent large overruns. Additional research, studying the causes of poor communication in more detail, can be valuable for document sharing adjustments in the future.

   (d) **Delayed documentation - waiting and assumptions**
   Waiting on documentation and using assumptions to start activities early are discussed at the same time, since a balance between waiting and assumptions lead to an improvement in project execution. Waiting on delayed documentation is advised for projects with a forecasted duration that does not exceed an overrun of 11%. The use of assumptions to start activities early is advised for projects that have a forecasted duration that exceeds an overrun of 11%. By this approach, the beneficial characteristics of both scenarios are used, and the negative influence is limited.

3. *What potential improvement can be found by the proposed documentation sharing strategy?*
   The alternative documentation sharing strategy is the proposed documentation sharing strategy. The alternative documentation sharing strategy shows improvement in terms of a reduction in the likelihood of an unacceptable process outcome. Also, the alternative strategy reports a declined median value. Compared to a situation where activities are forced to wait on documentation is schedule risk reduced by 2.4%. Compared to a scenario were assumptions are allowed a small increase in schedule risk is reported of 0.7%. However, in chapter 3 is described that schedule risk should not be used for close comparison. Therefore, the alternative documentation sharing strategy is ranked by the scoring table in Table 4.10. Improvement is seen in the obtained insight into the balance between waiting and assumptions, and the identification of activities that affect project duration most when rework occurs. *Outcome of this study allows process owners to avoid the most negative impact of waiting and assumptions to improve project execution.*

5

Conclusions and Recommendations

This chapter aims to give an overview of the research conclusions and recommendations. Results from the previous chapters, especially chapter 4, are used to formulate an answer to the central research question. Recommendations are formed based on the conclusions. This includes recommendations to the DTC Project Management Department, and recommendations for further research on this topic.

5.1. Conclusion and reflection

In this section the conclusions are discussed by Subsection 5.1.1. A reflection on this conclusions is given in Subsection 5.1.2.

5.1.1. Conclusions

The conducted research leads to an answer to the central research question and fulfilled the research goals. At the beginning of this thesis, a central research question is stated:

'In what manner can project execution at DTC be improved, by providing insight into the effect of the quality of shared project documentation on project success?'

The simulation of the various scenarios for documentation quality attributes in the c has obtained valuable insight into the documentation flow for DTC-projects. The construction and product development industries related documentation quality to rework. With the DSM-based discrete-event Monte Carlo simulation it has become possible to rank causes of documentation related rework for a DTC-project:

1. **Mistakes in the documentation**
   Accuracy
2. **Input changes in the documentation**
   Certainty
3. **Poor communication**
   Clarity, Conformance, Coordination and Legibility
4. **Delayed documentation - Waiting**
   Timeliness
5. **Delayed documentation - Assumptions**
   Completeness

In this ranking, the cause with the highest impact on project duration is ranked the highest.

Further detailed analysis are carried out on the different documentation quality attributes to identify activities that affect project duration the most when rework is generated:

- Construction drawings

- Mechanical diagrams
- Block outfitting equipment
- Block outfitting hot works
- Zone outfit hot works
- Zone outfit equipment
- Zone outfitting finalize space

Project managers must keep a close eye on these activities during the entire project execution phase. The progress and rework on these activities should be monitored to prevent unforeseen large overruns due to rework.

Further an alternative documentation sharing strategy is proposed. Under the alternative strategy activities:

1. Should wait on documentation when forecasted project duration does not exceed an overrun of 11% compared to the scheduled duration.
2. Should use assumptions to start activities early when forecasted project duration does exceed an overrun of 11% compared to the scheduled duration.

By this approach, beneficial characteristics of both scenario are used and negative impact is reduced.

5.1.2. Reflection

The presented conclusions in 5.1.1, are mainly based on results of the DSM-based discrete-event Monte Carlo simulation of a DTC-project. Used assumptions and model limitations affect these results. Thus the conclusions must be placed in the right perspective. The most important limitations of the mode and their impact on the findings, are discussed.

First of all, assumptions are used to include uncertainty in initial task duration, a characteristic of engineer-to-order projects. As described in Section 3.2 is a triangular distribution used to sample activity durations. However, assumptions are used to determine potential under- and overruns. This means that care is required once these results are used for a process with different under- and overrun vales. The impact of an increase from 1.2 to 1.3 is tested, and detailed information is presented in Appendix B.3. In this analysis, an overall shift to the right is identified for process duration. This observation is reasonable as activities can have a potential larger overrun. Under the proposed alternative documentation sharing strategy outliers are still reduced, and the median value is kept low—however, schedule risk and the likelihood of an unacceptable outcome report less favourable numbers. Due to the overall shift to the right, the presence of outliers affects schedule risk more than in the initial situation. Although the outliers affect schedule risk, the effect is limited from minimum to median value. Overall care is required once the results are applied to processes with other overrun numbers than the used assumptions. However, when differences are small results will not change much. When the outcomes are used for operations that show significant deviations in under- and overrun numbers, this conclusion should not be implemented without additional analysis.

Also, any other rework than project documentation-related rework is eliminated in this study. It is understood that other causes of rework, outside the scope of this study, can influence the potential outcome. However, this study shows important areas for project documentation-related rework and its impact.

Further it is assumed that project schedules give a reasonable estimation of activity durations. Initial activity durations are a crucial input value for the DSM-based discrete-event Monte Carlo simulation and its output.

Besides, it is assumed that rework impact is a fixed number. Thus, the consequence of rework does not change with the number of interactions. In reality, the rework impact is not a fixed number. Also, it is possible to take action when rework escalates. In the DSM-based discrete-event Monte Carlo simulation, it is not possible to take action once a project is started. Combined, simulation outcome is based on conservative numbers. Thus, it is likely that including the described features will lead to less rework.

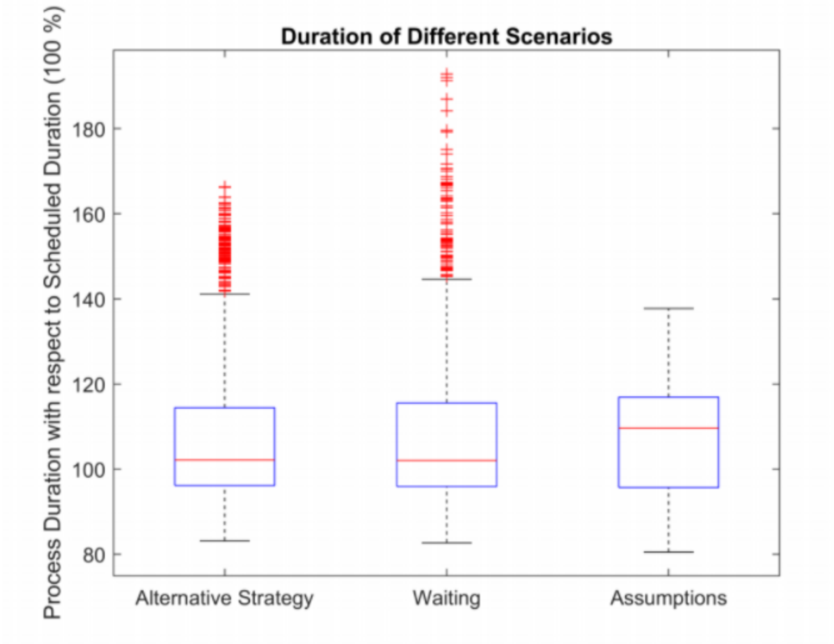


Figure 5.1: Check the influence of the assumptions for uncertainty in initial task duration

Altogether, one should keep in mind that this study is an improvement study and not an optimisation study. However, an optimisation study can potentially give more precise advises on the improvement of project execution, are the results of this study helpful to set the first steps in the direction of the progress of project execution.

5.2. Recommendations

As a result of this study, various recommendations are made to the DTC Project Management Department to improve the execution of their projects.

5.2.1. Recommendations to the DTC Project Management Department

DTC-projects are unique in there executing approach as vessels are built at non-Damen owned yards. This approach requires sophisticated documentation sharing strategy, as every building locations has different information demands. Although, documentation sharing strategy is vital for all projects, is it even more important for the performances of DTC-projects. Since the changing locations and engineering-to-order projects, it must be first-time-right.

Due to the importance of the documentation sharing strategy for DTC-projects is additional research on this topic advised. Answers to the following potential research questions will help to improve the documentation sharing strategy of DTC-projects.

1. When should input changes no longer be accepted?
2. What are the causes of mistakes?
3. What are the causes of poor communication?
4. Does the proposed ranking changes for DTC-projects where DTC has different responsibilities?
5. What is the optimal balance between waiting and using assumptions?
6. What percentage of a tasks needs to be completed to make good enough assumptions?
7. What are the ”clean” activity duration values?
8. Does the proposed ranking changes if cost is included?

Besides this potential research questions is the following recommended:

- Monitor actual rework numbers for the planned activities. In terms of:
  - Frequency
  - Severity

A

Data Collection

A.1. Task decomposition and dependencies

A.1.1. Question form engineering

Confidential

A.1.2. Question form procurement

Confidential

A.1.3. Question form material coordination

Confidential

A.1.4. Question form project management

Confidential

A.1.5. Question form production

Confidential

A.2. Rework probability assessment

A.2.1. Stage 1: subjective assessment

To start with Information Variability (IV). This value represents the stability of information about specific tasks. So each input tasks has its own IV value. Similar to the determination of rework probabilities, it is also hard to define a value for IV. To solve this problem, Yassine et al. (2001) uses a subjective measurement scale, shown in Table A.1.

Table A.1: Levels of information variability as mentioned by Yassine et al. (2001)

Value	Description	Likelihood of Change
1	Stable	Low
2	Unknown	Medium to high
3	Unstable	Very high

To continue with the next dimension; task sensitivity (TS). TS defines the sensitivity of a task to changes or modifications to input information. Again a three-level scale is used to rank TS, displayed by Table A.2.

Table A.2: Levels of task sensitivity (TS) as mentioned by Yassine et al. (2001)

Value	Description	Dependent Task is:
1	Low	Insensitive to most information changes
2	Medium	Sensitive to major information changes
3	High	Sensitive to most information changes

Finally, IV and TS are multiplied to get *Task Volatilities (TV)*, as shown by equation 3.6 (Yassine et al., 2001). Outcomes of TV can be used to define a strategy for process architecture design, as shown in Table A.3. However, the outcome does not lead to a rework probability directly; it gives information on how rework of different activities relates to each other. A task with a TV value of 9, has more rework then a task with a TV value of 2. Additional operations are needed to convert the 1-9 scale into probability values.

$$TV = IV \times TS \tag{A.1}$$

Table A.3: Levels of task volatilities as mentioned by Yassine et al. (2001)

Value	Description	Strategies
1 or 2	Dependency is weak, Low risk of rework	Feedback can be used, especially if it promotes process robustness
3 or 4	Dependency is moderate, Moderate risk of rework	Avoid feedbacks were possible
6 or 9	Highly sensitive to change, high risk of rework	Do not use feedbacks

A.2.2. Stage 2: Mapping and calibration

As before mentioned, in the first stage TV values are determined for each task. But these TV values are not reworking probabilities yet. Therefore an additional step is required. This entails a simulation of the process environment, described in detail later in this chapter in Section **??**.

TV values represent the probability that rework will occur on a scale from ’1’ to ’9’ (Yassine et al., 2001). Where an activity with TV value 9 is likely to have more rework then an activity with TV value 1. Thus the TV values only indicate how the rework probabilities of the different activities are related. This gives no information about the actual probability value, that should be linked to these TV values. TV values are linked to reworking probabilities from 0% to a maximum value P to make the connection with actual probability values (Yassine et al., 2001). Where the maximum value is assigned to the highest

TV value, other values are computed by a linear relationship. Yassine et al. (2001) demonstrates that it is also possible to use non-linear relationships for this mapping. It is decided to use a linear relationship here as it gives conservative values. As no other work is found applying this assessment to shipbuilding projects, findings cannot be compared to other results in shipbuilding.

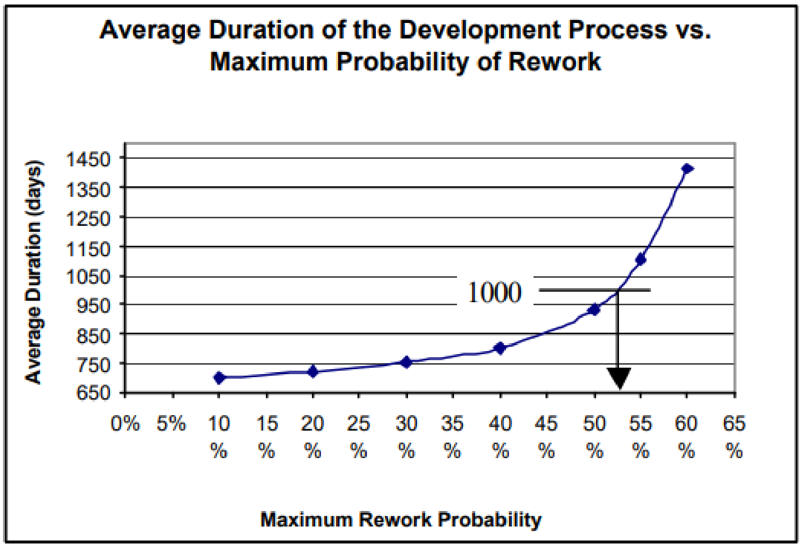


Figure A.1: Example of linking TV values to probabilities by max. rework probability, (Known Duration = 1000 days) (Yassine et al., 2001)

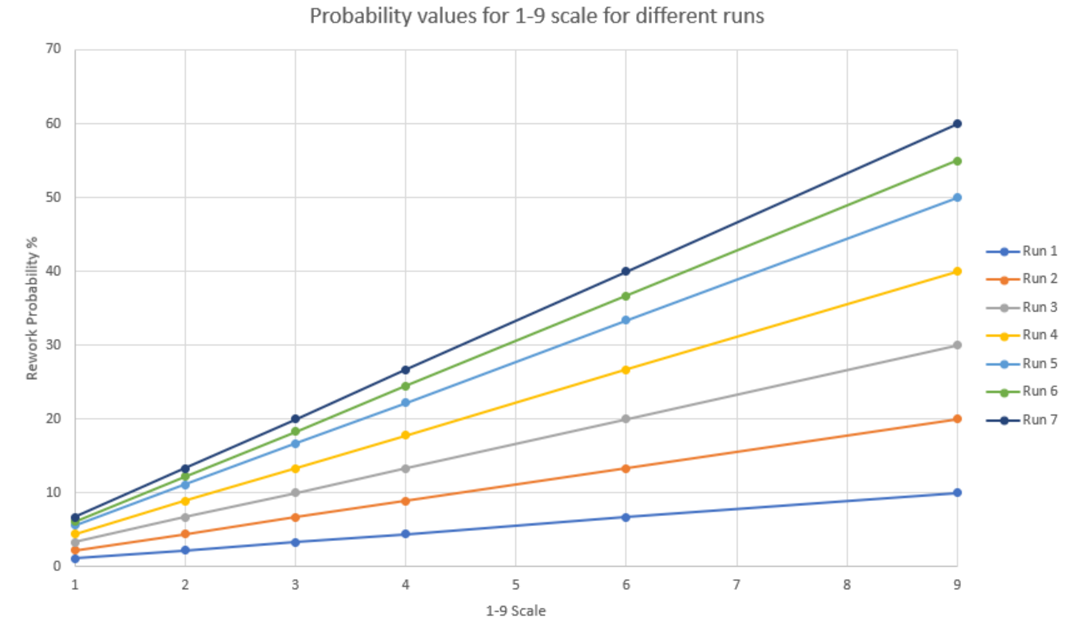


Figure A.2: Illustration of calculation of rework probabilities for TV values for different P values

If it is possible to determine a value for P, all TV values can be linked to a probability value. A process with known duration or cost is simulated for different values of P to make this correlation. After each run, the average duration or cost is noted. Values of P will be increased after each run until the outcome of the simulation matches with the average duration or cost as defined at the beginning. Such a procedure is visualised in Figure A.1. The P-value that results in the known duration or cost is a P-value that represents the system. From this P-value, rework probabilities of the TV values can be computed. To give an example for this: Figure A.1, used seven simulation runs to find a P-value (maximum rework probability) that leads to an average duration of 1000 days. Figure A.2, shows the calculated rework probabilities for the TV values for all these seven runs. The actual rework probabilities lay between run five and run six. Yassine et al. (2001) argues that using a linear relation for computing rework probabilities for other TV values results in more conservative values, then when non-linear methods are used. The difference is especially seen in the middle TV values, as non-linear techniques use a less rapid increase of the rework probability. This difference is shown in Figure A.3.

This manner of scaling is needed because rework probabilities are sensitive to the process of interest. It is not possible to have a mapping procedure without information on the specific system as this will lead to rework probabilities that do not give a good representation of the system in reality. Even more will this result in deviating values for process duration and cost, as those are highly dependent on rework.

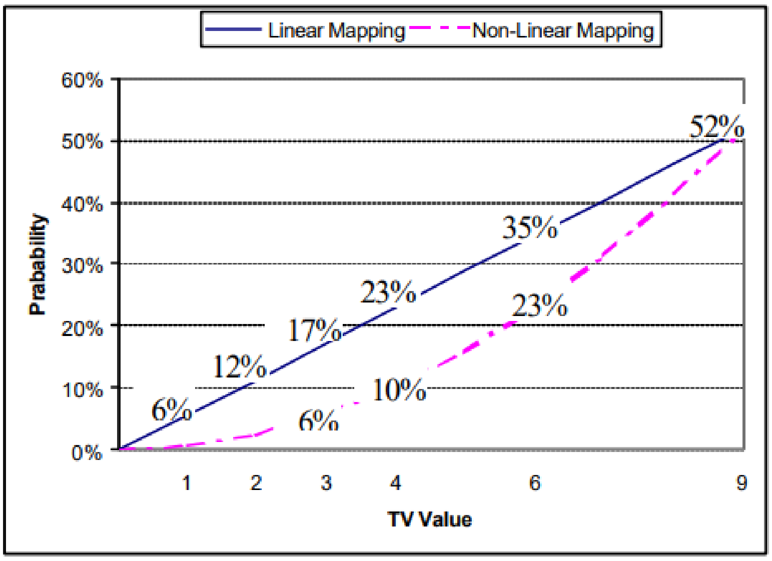


Figure A.3: Linear and Non-Linear mapping as defined by Yassine et al. (2001)

A.2.3. Stage 3: Validation

The last stage of this process is to validate the determined rework probabilities. This is done by interviews with people inside the process. In these interviews, the found rework probabilities are discussed according to the flow chart for the validation process defined by Yassine et al. (2001). The main lines of these interviews are to see if can agreed to the identified rework probabilities. If not, it is wanted to know why the identified rework probabilities are not representative of the process.

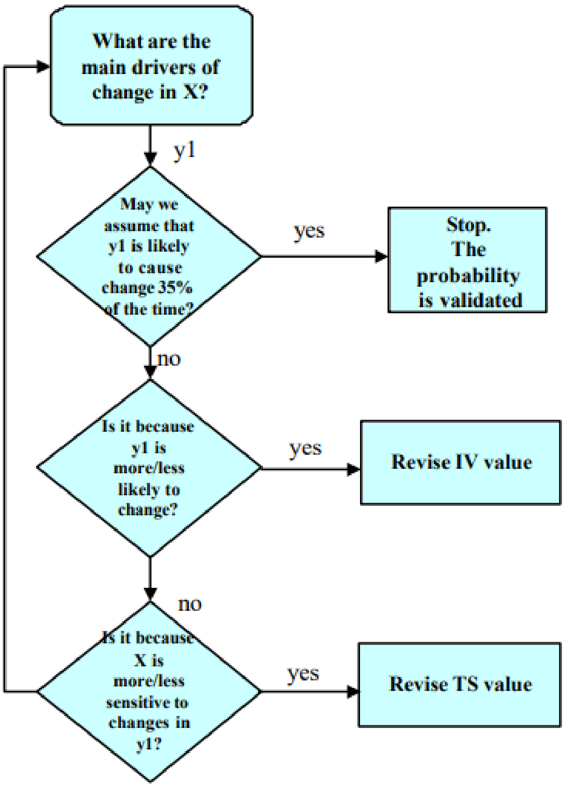


Figure A.4: Flowchart for the validation stage of the rework probabilities (Yassine et al., 2001)

B

MATLAB SCRIPTS

B.1. Generate project-specific DSM structure

Script to generate DSM structure

10-8-2020

A. Reumer

To run the DSM based discrete-event Monte Carlo simulation

```
clc
close all
```

Input

Number of DSM structure building blocks

```
% FILL-IN FOR SPECIFIC PROEJCT
n_DC = 1;       % Design Check ALWAYS 1
n_BE = 1;       % Number of Basic Engineering batches ALWAYS 1
n_DE_f = 1;     % Detailed Engineering that does not come in multiple batches, ALWAYS  1
n_DE_CP = 1;    % Number of construction and piping batches. It is assumed number of constructi
n_sec = 1;      % Number of sections
n_bl = 1;       % Number of blocks
n_zone = 1;     % Number of zones
n_ha = 1;       % Number of hull assembly actions
n_COM = 1;      % Number of Commissioning blocks ALWAYS 1
```

Load Standard DSM Matrices

Binary DSM

```
DSM_STR = readmatrix('DSM_BASIC_S33.xlsx', 'Sheet', 'DSM', 'Range', 'J2:GH182');   % Load stand
REP = isnan(DSM_STR);                                % Replace NaN by 0
DSM_STR(REP) = 0;                                    % Replace NaN by 0
DSM_STR(DSM_STR>1) = 0;                              % Replace values > 1 by 0, (only needed to load i
txtData  = readcell('DSM_BASIC_S33.xlsx', 'Sheet', 'DSM', 'Range', 'I2:I182');     % Store task
ST = arrayfun(@num2str,DSM_STR,'UniformOutput',false);    % Convert matrix to string to store i
TXT_STR = cell(length(DSM_STR)+1,length(DSM_STR)+1);      % Create cell array
TXT_STR(1,:) = [ '-'; txtData];                           % Fill cell array
TXT_STR(:,1) = [ '-'; txtData];                           % Fill cell array
TXT_STR(2:length(DSM_STR)+1, 2:length(DSM_STR)+1) = ST;   % Fill cell array
```

Rework Probabilities

```
DSM_STR_1 = readmatrix('DSM_BASIC_S33.xlsx', 'Sheet', 'DSM_1_TV', 'Range', 'J2:GH182'); % Load
REP = isnan(DSM_STR_1);                              % Replace NaN by 0
DSM_STR_1(REP) = 0;                                  % Replace NaN by 0
txtData_1 = txtData;                                 % Store tasks names
ST_1 = arrayfun(@num2str,DSM_STR_1,'UniformOutput',false);      % Convert matrix to string to s
TXT_STR_1 = cell(length(DSM_STR_1)+1,length(DSM_STR_1)+1);      % Create cell array
TXT_STR_1(1,:) = [ '-'; txtData_1];                             % Fill cell array
TXT_STR_1(:,1) = [ '-'; txtData_1];                             % Fill cell array
TXT_STR_1(2:length(DSM_STR_1)+1, 2:length(DSM_STR_1)+1) = ST_1; % Fill cell array
```

Rework Impact

```
DSM_STR_2 = readmatrix('DSM_BASIC_S33.xlsx', 'Sheet', 'DSM_2', 'Range', 'J2:GH182');    % Load
REP = isnan(DSM_STR_2);                  % Replace NaN by 0
DSM_STR_2(REP) = 0;                      % Replace NaN by 0
DSM_STR_2 = DSM_STR_2./100 ;             % Convert rework impact to values between 0-1
txtData_2 = txtData;                     % Store tasks names
ST_2 = arrayfun(@num2str,DSM_STR_2,'UniformOutput',false);      % Convert matrix to string to s
TXT_STR_2 = cell(length(DSM_STR_2)+1,length(DSM_STR_2)+1);      % Create cell array
TXT_STR_2(1,:) = [ '-'; txtData_2];                             % Fill cell array
TXT_STR_2(:,1) = [ '-'; txtData_2];                             % Fill cell array
TXT_STR_2(2:length(DSM_STR_2)+1, 2:length(DSM_STR_2)+1) = ST_2; % Fill cell array
```

Break Up in Separate Blocks

```
% Define breake points for predefined tasks
V = [0; 10; 25; 31; 37; 42; 61; 63; 93; 123; 148; 158; 168; 175; 177; 181];
v = V+1;

% Break up standard DSM matrix in smaller matrices and store names of tasks
MAT = cell(length(V));
MAT_1 = cell(length(V));
MAT_2 = cell(length(V));
NAMES = cell(1,length(V));
NAMES_1 = cell(1,length(V));
NAMES_2 = cell(1,length(V));
for W = 1:1:length(V)-1
    for Q =1:1:length(V)-1
        MAT{Q+1, W+1} = DSM_STR(V(Q)+1:V(Q+1), V(W)+1:V(W+1));
        MAT_1{Q+1, W+1} = DSM_STR_1(V(Q)+1:V(Q+1), V(W)+1:V(W+1));
        MAT_2{Q+1, W+1} = DSM_STR_2(V(Q)+1:V(Q+1), V(W)+1:V(W+1));
        NAMES{Q+1} = TXT_STR(v(Q)+1:v(Q+1),1);
        NAMES_1{Q+1} = TXT_STR_1(v(Q)+1:v(Q+1),1);
        NAMES_2{Q+1} = TXT_STR_2(v(Q)+1:v(Q+1),1);
    end
end

% Stadard matrix for Binary DSM
MAT(1,:)=[];
MAT(:,1) =[];
NAMES(:,1)=[];

%Standard matrix for Rework Probabilities
MAT_1(1,:)=[];
MAT_1(:,1) =[];
NAMES_1(:,1)=[];

% Standard matrix for Rework Impact
MAT_2(1,:)=[];
MAT_2(:,1) =[];
NAMES_2(:,1)=[];
```

```
Names = {'Design Check', 'Basic Engineering', 'RFQ Sections', 'RFQ Blocks', 'RFQ Zones','DE not

% Block Matrix Binary DSM
BLOCK_DSM = cell(length(V),length(V));   % Store the DSM blocks and their names in a cell array
BLOCK_DSM(2:end,1) = Names;              % Fill cell array
BLOCK_DSM(1, 2:end) = Names;             % Fill cell array
BLOCK_DSM(2:end,2:end) = MAT;            % Fill cell array
```

```
%Block Matrix Rework Probabilities
BLOCK_DSM_1 = cell(length(V),length(V)); % Store the DSM blocks and their names in a cell array
BLOCK_DSM_1(2:end,1) = Names;            % Fill cell array
BLOCK_DSM_1(1, 2:end) = Names;           % Fill cell array
BLOCK_DSM_1(2:end,2:end) = MAT_1;        % Fill cell array

% Block Matrix Rework Impact
BLOCK_DSM_2 = cell(length(V),length(V)); % Store the DSM blocks and their names in a cell array
BLOCK_DSM_2(2:end,1) = Names;            % Fill cell array
BLOCK_DSM_2(1, 2:end) = Names;           % Fill cell array
BLOCK_DSM_2(2:end,2:end) = MAT_2;        % Fill cell array
```

Build Up DSM Structure Based on batches

```
N_INPUT = [n_DC; n_BE; n_sec; n_bl; n_zone; n_DE_f; n_DE_CP; n_sec; n_bl; n_zone; n_sec; n_bl;

ADD_BL = N_INPUT - ones(length(N_INPUT),1);                       % Additional blocks with resp
ADD = sum(ADD_BL);                                                % Number of cells that needs
ADD_str = arrayfun(@num2str,ADD_BL,'UniformOutput',false);        % Convert data to string valu

DSM_NEW = cell(length(BLOCK_DSM)+ADD,length(BLOCK_DSM)+ADD);      % Pre - Allocation
ADD_V = cell(length(BLOCK_DSM)-1,2);                              % Fill cell array
ADD_V(:,1) = BLOCK_DSM(2:end,1);                                  % Fill cell array
ADD_V(:,2) = ADD_str;                                             % Fill cell array

DSM_NEW_1 = cell(length(BLOCK_DSM_1)+ADD,length(BLOCK_DSM_1)+ADD); % Pre - Allocation
ADD_V_1 = cell(length(BLOCK_DSM_1)-1,2);                           % Fill cell array
ADD_V_1(:,1) = BLOCK_DSM_1(2:end,1);                               % Fill cell array
ADD_V_1(:,2) = ADD_str;                                            % Fill cell array

DSM_NEW_2 = cell(length(BLOCK_DSM_2)+ADD,length(BLOCK_DSM_2)+ADD); % Pre - Allocation
ADD_V_2 = cell(length(BLOCK_DSM_2)-1,2);                           % Fill cell array
ADD_V_2(:,1) = BLOCK_DSM_2(2:end,1);                               % Fill cell array
ADD_V_2(:,2) = ADD_str;                                            % Fill cell array
```

Fill new DSM

```
% Define how often certain blocks need to be printed
DSM_FIL = zeros(length(ADD_BL),4);           % Pre-allocation for speed
DSM_FIL_1 = zeros(length(ADD_BL),4);         % Pre-allocation for speed
DSM_FIL_2 = zeros(length(ADD_BL),4);         % Pre-allocation for speed
for L = 1:1:length(ADD_BL)
    DSM_FIL(L,1)= L;
```

```
    DSM_FIL_1(L,1)= L;
    DSM_FIL_2(L,1)= L;
    if any(ADD_BL(L))
        for Z = 1:1:ADD_BL(L)
            DSM_FIL(L,Z+1) = L;
            DSM_FIL_1(L,Z+1) = L;
            DSM_FIL_2(L,Z+1) = L;
        end
    end
end
DSM_FILLED = DSM_FIL(:);
DSM_FILLED = sort(DSM_FILLED);
DSM_FILLED(DSM_FILLED == 0)=[];

DSM_FILLED_1 = DSM_FIL_1(:);
DSM_FILLED_1 = sort(DSM_FILLED_1);
DSM_FILLED_1(DSM_FILLED_1 == 0)=[];

DSM_FILLED_2 = DSM_FIL_2(:);
DSM_FILLED_2 = sort(DSM_FILLED_2);
DSM_FILLED_2(DSM_FILLED_2 == 0)=[];

% Print new DSM in a BLOCK structure || no dependency among blocks jet,
% they are just printed multiple times
NAMES_NEW = cell(length(DSM_FILLED),1);          % Pre-allocation for speed
NAMES_NEW_1 = cell(length(DSM_FILLED_1),1);      % Pre-allocation for speed
NAMES_NEW_2 = cell(length(DSM_FILLED_2),1);      % Pre-allocation for speed

% Set-Up Project Specific Matrices
for G = 1:1:length(DSM_FILLED)
NAMES_NEW(G) = NAMES(DSM_FILLED(G));
NAMES_NEW_1(G) = NAMES(DSM_FILLED_1(G));
NAMES_NEW_2(G) = NAMES(DSM_FILLED_2(G));
    for P = 1:1:length(DSM_FILLED)
        % Project Specific Binary DSM
        DSM_NEW(P+1,G+1) = BLOCK_DSM((DSM_FILLED(P)+1),(DSM_FILLED(G)+1) );
        DSM_NEW(P+1, 1) = BLOCK_DSM(DSM_FILLED(P)+1,1);
        DSM_NEW(1, G+1) = BLOCK_DSM(1,(DSM_FILLED(G)+1));
        % Project Specific Rework Probabilities
        DSM_NEW_1(P+1,G+1) = BLOCK_DSM_1((DSM_FILLED_1(P)+1),(DSM_FILLED_1(G)+1) );
        DSM_NEW_1(P+1, 1) = BLOCK_DSM_1(DSM_FILLED_1(P)+1,1);
        DSM_NEW_1(1, G+1) = BLOCK_DSM_1(1,(DSM_FILLED_1(G)+1));
        % Project Specific Rework Impact
        DSM_NEW_2(P+1,G+1) = BLOCK_DSM_2((DSM_FILLED_2(P)+1),(DSM_FILLED_2(G)+1) );
        DSM_NEW_2(P+1, 1) = BLOCK_DSM_2(DSM_FILLED_2(P)+1,1);
        DSM_NEW_2(1, G+1) = BLOCK_DSM_2(1,(DSM_FILLED_2(G)+1));
    end
end
```

Add Dependencies among batches

```
% Add dependencies among batches (No dependencies between batches /
% building blocks of the same type) For example no dependency between MC1
% and MC2
```

```
B = unique(DSM_FILLED);
Ncount = histc(DSM_FILLED, B);
for U = 1:1: length(Ncount)
        if Ncount(U) > 1
            for Y = (sum(Ncount(1:U-1))+2):((sum(Ncount(1:U-1))+1) + Ncount(U))
                for R = (sum(Ncount(1:U-1))+2):((sum(Ncount(1:U-1))+1) + Ncount(U))
                    if Y == R
                        continue;
                    else
                        % Dependencies Binary DSM
                        DSM_NEW{Y,R} = zeros(length(DSM_NEW{Y,R}));
                        % Dependencies Rework Probabilities
                        DSM_NEW_1{Y,R} = zeros(length(DSM_NEW_1{Y,R}));
                        % Dependencies Rework Impact
                        DSM_NEW_2{Y,R} = zeros(length(DSM_NEW_2{Y,R}));
                    end
                end
            end
        end
end
```

Add Building Strategy

To add the building strategy the link between the different blocks in the DSM_NEW matrix is defined in a seperate Excel Sheet 'LB' (Link Batches). With '0' and '1' dependencies are made inactive or active.

```
% Load dependency information from Excel file
DEP_mat = readmatrix('DSM_BASIC_S33.xlsx', 'Sheet', 'LB', 'Range', 'C2:ET149' );
DEP_REP = isnan(DEP_mat);                    % Replace NaN values with 0
DEP_mat(DEP_REP)=0;                          % Replace NaN values with 0
DEP_mat(DEP_mat>1) =0;                       % Replace values > 1 with 0 (required to load entire matri
txtDataDEP = readcell('DSM_BASIC_S33.xlsx', 'Sheet', 'LB',  'Range', 'B2:B149'); % Load text
ST_DEP = arrayfun(@num2str,DEP_mat,'UniformOutput',false);      % Convert matrix to string to s
TXT_DEP = cell(length(DEP_mat)+1,length(DEP_mat)+1);            % Create cell array
TXT_DEP(1,:) = [ '-'; txtDataDEP];                              % Fill cell array
TXT_DEP(:,1) = [ '-'; txtDataDEP];                              % Fill cell array
TXT_DEP(2:length(DEP_mat)+1, 2:length(DEP_mat)+1) = ST_DEP;     % Fill cell array
clearvars k W                                                   % Clear variables

% Binary DSM
FILL_cell = cell(length(DEP_mat), length(DEP_mat));     % Pre-allocation for speed
DSM_UPD = DSM_NEW;
DSM_UPD(1,:) = [];  % Activity names are not relevant for this step
DSM_UPD(:,1) = [];  % Activity names are not relevant for this step

% Rework Probabilities
FILL_cell_1 = cell(length(DEP_mat), length(DEP_mat));   % Pre-allocation for speed
DSM_UPD_1 = DSM_NEW_1;
DSM_UPD_1(1,:) = [];    % Activity names are not relevant for this step
DSM_UPD_1(:,1) = [];    % Activity names are not relevant for this step

% Rework Impact
FILL_cell_2 = cell(length(DEP_mat), length(DEP_mat));   % Pre-allocation for speed
```

```
DSM_UPD_2 = DSM_NEW_2;
DSM_UPD_2(1,:) = [];    % Activity names are not relevant for this step
DSM_UPD_2(:,1) = [];    % Activity names are not relevant for this step

% The DSM_UPD cell array is filled with matrices of ones or zeros based on
% a present or absence link
for k = 1:1:length(DEP_mat)
    for w = 1:1:length(DEP_mat)
        if any(DEP_mat(k,w))
        % Binary DSM
        FILL_cell{k,w} =   ones(length(DSM_UPD{k,w}(:,1)), length(DSM_UPD{k,w}(1,:)));
        % Rework Probabilities
        FILL_cell_1{k,w} = ones(length(DSM_UPD_1{k,w}(:,1)), length(DSM_UPD_1{k,w}(1,:)));
        % Rework Impact
        FILL_cell_2{k,w} = ones(length(DSM_UPD_2{k,w}(:,1)), length(DSM_UPD_2{k,w}(1,:)));
        else
        % Binary DSM
        FILL_cell{k,w} =   zeros(length(DSM_UPD{k,w}(:,1)), length(DSM_UPD{k,w}(1,:)));
        % Rework Probabilities
        FILL_cell_1{k,w} = zeros(length(DSM_UPD_1{k,w}(:,1)), length(DSM_UPD_1{k,w}(1,:)));
        % Rework Impact
        FILL_cell_2{k,w} = zeros(length(DSM_UPD_2{k,w}(:,1)), length(DSM_UPD_2{k,w}(1,:)));
        end
    end
end

% Multiply the cell arrays elementswise to get a Binary DSM matrix with dependencies among batc
DSM_NEW_DEP = cell(length(DSM_NEW));    % Pre-allocation for speed
DSM_NEW_DEP(1,:) = DSM_NEW(1,:);        % Reprint Activity names
DSM_NEW_DEP(:,1) = DSM_NEW(:,1);        % Reprint Activity names
DSM_NEW_DEP(2:end,2:end) = cellfun(@(x,y) x.*y, DSM_UPD, FILL_cell, 'UniformOutput', false);

 % Multiply the cell arrays elementswise to get a Rework Probability matrix with dependencies a
DSM_NEW_DEP_1 = cell(length(DSM_NEW_1));    % Pre-allocation for speed
DSM_NEW_DEP_1(1,:) = DSM_NEW_1(1,:);        % Reprint Activity names
DSM_NEW_DEP_1(:,1) = DSM_NEW_1(:,1);        % Reprint Activity names
DSM_NEW_DEP_1(2:end,2:end) = cellfun(@(x,y) x.*y, DSM_UPD_1, FILL_cell_1, 'UniformOutput', fals

% % Multiply the cell arrays elementswise to get a Rework Impact matrix with dependencies among
DSM_NEW_DEP_2 = cell(length(DSM_NEW_2));    % Pre-allocation for speed
DSM_NEW_DEP_2(1,:) = DSM_NEW_2(1,:);        % Reprint Activity names
DSM_NEW_DEP_2(:,1) = DSM_NEW_2(:,1);        % Reprint Activity names
DSM_NEW_DEP_2(2:end,2:end) = cellfun(@(x,y) x.*y, DSM_UPD_2, FILL_cell_2, 'UniformOutput', fals
```

Print Project-Specific Matrices

Project-Specific Binary DSM

```
% Convert DSM_NEW in 'Block' structure to a matrix structure
DSM_C = DSM_NEW_DEP ;
DSM_C(1,:) = [];
DSM_C(:,1) = [];
PRINT_DSM = cell2mat(DSM_C);    %% PROJECT SPECIFIC DSM [double array]
```

Project-Specific Rework Probabilities

```
% Convert DSM_NEW in 'Block' structure to a matrix structure
DSM_C_1 = DSM_NEW_DEP_1 ;
DSM_C_1(1,:) = [];
DSM_C_1(:,1) = [];
PRINT_DSM_1= cell2mat(DSM_C_1);    %% PROJECT SPECIFIC DSM1 double array
```

Project-Specific Rework Impact

```
% % Convert DSM_NEW in 'Block' structure to a matrix structure
DSM_C_2 = DSM_NEW_DEP_2 ;
DSM_C_2(1,:) = [];
DSM_C_2(:,1) = [];
PRINT_DSM_2= cell2mat(DSM_C_2);    %% PROJECT SPECIFIC DSM2 double array
```

B.2. DSM-based discrete-event Monte Carlo simulation

DSM based Discrete-Event Monte Carlo Simulation

2-7-2020

A. Reumer

```
clc
close all
```

Simulation Input

```

%Load binary DSM, rework probabilities and impact from Excel file
DSM = xlsread('InputV5',1,'B2:O15');
DSM_1 = xlsread('InputV5',2,'B2:O15');
DSM_2 = xlsread('InputV5',3,'B2:O15');

%Duration input
BCV_d = xlsread('InputV5',4,'B3:B16');
MLV_d = xlsread('InputV5',4,'C3:C16');
WCV_d = xlsread('InputV5',4,'D3:D16');
%Cost input
BCV_c = xlsread('InputV5',4,'E3:E16');
MLV_c = xlsread('InputV5',4,'F3:F16');
WCV_c = xlsread('InputV5',4,'G3:G16');

IC = xlsread('InputV5',4,'H3:H16');       %Learning Curve
m = 14;
run("sequence_vect.mlx")                     % Run script to specify sequence of activities

DSM = CREATE_DSM;
DSM_1 = CREATE_DSM_1;
DSM_2 = CREATE_DSM_2;
BCV_d = CREATE_BCV_d.*1;
MLV_d = CREATE_MLV_d.*1;
WCV_d = CREATE_WCV_d.*1;

BCV_c = CREATE_BCV_c;
MLV_c = CREATE_MLV_c;
WCV_c = CREATE_WCV_c;

IC = CREATE_IC;
```

```

% Sequence Vector
%Other input
m = 14;                                 %number of tasks
n = 1100;                               %numbers of sample LHS
COR = [1 0.9;0.9 1];                    %Correlation between duration and cost
W = ones(m,1);                          %Work vector, '1' stands for 100% of work is remaining
WNi = zeros(m,1);                       %Work Now vector, boolean entities TRUE = '1' , FALSE =
S = 0;                                  %Cumulative duration for a given run
C = 0;                                  %Cumulative cost for a given run
```

```
T_s = 130;                                % Target for duration in [days]
T_c = 630;                                % Target for cost in [$k]
```

Uncertainty in Duration and Cost

PDF Generation

```
rng('default');
s = rng;
pd_d=cell(1,2);
pd_c=cell(1,2);
for k = 1:1:m
pd_d{k} = makedist('Triangular',BCV_d(k), MLV_d(k),WCV_d(k)); %Construct PDF for all durations
pd_c{k} = makedist('Triangular',BCV_c(k), MLV_c(k),WCV_c(k)); %Construct PDF for all cost
end
PD = [pd_d' pd_c'];                                           % Merge duration and cost PDF in
```

Sample duration and cost from PDF functions with Latin Hypercube Sampling (LHS)

```
S_D = zeros(n, 2*m);        %Pre-allocation for speed
S_C = zeros(n, 2*m);        %Pre-allocation for speed
for P = 1:1:m
CorSamp = lhsgeneral(PD(P,:), COR, n);
CorSamp_D = CorSamp(:,1);
CorSamp_C = CorSamp(:,2);
S_D(:,P) = CorSamp_D;
S_C(:,P) = CorSamp_C;
end
S_D = S_D(1:n,1:m);         % Sampled initial task durations
S_C = S_C(1:n,1:m);         % Sampled initial task cost
```

Simulation Algorithm

```
S_TOT = zeros(n,1);             % Pre-Allocation for speed
C_TOT = zeros(n,1);             % Pre-Allocation for speed

% Loop to execute the process n times
for N = 1:1:n
W = ones(m,1);                  % Reset Work vector, '1' stands for 100% of work is remaining
loopcnt = 0;                    % Reset loop count
S=0;                            % Reset duration S [days]
C=0;                            % Reset cost C [$k]
WorkDone=[];                    % Pre-Allocation
Cost_ACT=[];                    % Pre-Allocation
T = [];                         % Pre-Allocation
myCoordList=[];                 % Pre-Allocation

    % The process is finnished if all work is done
    while any(W)
    Act_A = W>0;                %determine active activities for this stage
    WN = WNi;                   %reset work now vector

    %Find most upstream activity with unfinished work
```

```
    for j = 1:1:m
    if Act_A(j) == 0
        continue
    end
        WN(j) = 1;
        break
    end
%Check if there are other possible tasks with remaining work
    for g = j+1:1:m
        if Act_A(g) == 1 && all((Act_A(1:g-1).*DSM(g,1:g-1)') == 0)
            WN(g) = 1;
        else
            continue
        end
    end
%%Calculate Time until the next event (shortest activity ends), t
    t_v = WN.*W .* S_D(N,:)';        %duration of active activities
    t = min(t_v(t_v>0));             %find miminal duration of active activities
    I = find(t_v == t);              %Element of minimal duration
    T = [T; [t]];                    % Store t for all the loops in the while loop

%%Work on active activities

%Decrement W
    t_w = WN.*t;                             % Time worked on activities [days]
    W_d = t_w./ S_D(N,:)';                   % Work done in values between 0 - 1
    W = W - W_d;                             % Update work vector
    W(W<0)=0;                                % Adjust for values <0
    WorkDone=[WorkDone; [t_w], [W_d], [W]]; % Store values for all the loops in the while loop

%Increment S and C
    S = S + t;                               % Cumulative duration [days]
    C_act = W_d.*S_C(N,:)';                  % Calculate cost for active activities [$k]
    C = C + sum(C_act);                      % Cumulative cost [$k]
    Cost_ACT = [Cost_ACT; [C_act]];          % Store values for all the loops in the while loop
    myCoordList=[myCoordList; [S, C]];       % Store values for all the loops in the while loop

%%Check for rework Generated by completed activity (only possible from step 2)
    if I>1                                   % First activity cannot have first order rework
        CH_RW_FO = DSM_1(1:I-1,I);           % Select Rework Probabilities
        RD_FO = rand(I-1,1);                 % Create array with random numbers
        CHECK_FO = CH_RW_FO > RD_FO;         % Check if first-order rework occurs
        if any(CHECK_FO)                     % If first order rework is present,  work vector (W
            EL_FO = find(CHECK_FO);                  % Find element (task) with first-order rewo
            W_A_FO = DSM_2(EL_FO,I).*IC(EL_FO);      % Calculate additional work through rework.
            W(EL_FO) = W(EL_FO) + W_A_FO;            % Update Work Vector
            W(W>1) = 0.9;                            % Adjust Work Vector for values exceeding 1
            SO_RW = tril(DSM_1(EL_FO(1):I,EL_FO(1):I));
            SO_RW_D = DSM_1(I+1:m, EL_FO(1):I);
            SO_RW_T = [SO_RW; SO_RW_D];
            SO_DSM_2 = DSM_2(EL_FO(1):m,EL_FO(1):I);
            SO_IC = IC(EL_FO(1):m);
            RD_SO1 = rand(m-EL_FO(1)+1,I-EL_FO(1)+1);
            CH_SO1 = zeros(m-EL_FO(1)+1, I-EL_FO(1)+1);      % Pre-Allocation for speed
```

```
            WA_SO1_ST = zeros(m-EL_FO(1)+1, I-EL_FO(1)+1);   % Pre-Allocation for speed
            AT = W(EL_FO(1):m) < 1;                          % Rework can only occur when some
            % Check if second order rework occurs
            for U = 1:1:length(SO_RW_T(1,:))
                CHECK_SO1 = SO_RW_T(:,U) > RD_SO1(:,U);
                CH_SO1(:,U) = CHECK_SO1;
                WA_SO1 = AT.*CH_SO1(:,U).*SO_DSM_2(:,U).*SO_IC;
                WA_SO1_ST(:,U) = WA_SO1;
            end
            if length(EL_FO)>1
                if any(CH_SO1(:,(EL_FO+1)-EL_FO))                %Then there is second order rew
                    [row,col] = find(CH_SO1(:,(EL_FO+1)-EL_FO));
                    AK = unique(col);
                    AR = ones(length(SO_RW_T(1,:)),1);
                    for R = AK(1)+1:1:length(SO_RW_T(1,:))
                        if any(CH_SO1(R, 1:R-1))
                            AR(R) = 0;
                        else
                            CH_SO1(:,R) =0;
                        end
                    end
                    ARR = find(AR);
                    WA_SO1_ST(:,ARR ) = [];
                    WA_SO_TOT = sum(WA_SO1_ST,2);
                    W(EL_FO(1):m) = W(EL_FO(1):m) + WA_SO_TOT;  % Update Work Vector
                    W(W>1) = 0.9;                                % Adjust Work Vector for values
                end
            else
                if any(SO_RW_T(:,EL_FO-EL_FO+1) > rand(m-EL_FO+1,1))
                    if any(CH_SO1(:,1))
                        AR2 = ones(length(SO_RW_T(1,:)),1);
                        AR2(1) =0 ;
                        for V = 2:1:length(SO_RW_T(1,:))
                            if any(CH_SO1(V, 1:V-1))
                                AR2(V) = 0;
                            else
                                CH_SO1(:,V) =0;
                            end
                        end
                        ARR2 = find(AR2);
                        WA_SO1_ST(:,ARR2 ) = [];
                        WA_SO_TOT = sum(WA_SO1_ST,2);
                        W(EL_FO:m) = W(EL_FO:m) + WA_SO_TOT;    % Update Work Vector
                        W(W>1) = 0.9;                            % Adjust Work Vector for values
                    end
                end
            end
        end
    end
    loopcnt = loopcnt +1;                            % Count loops in a while loop. So number of loo
    S_tot = myCoordList(loopcnt,1);                  % Total Duration in this process run in [days]
    C_tot = myCoordList(loopcnt,2);                  % Total Cost in this process run in [$k]
    WORK = zeros(m, length(t));                                % Pre-allocation for speed
    WORKD = zeros(m, length(t));                               % Pre-allocation for speed
```

```
    T_W = zeros(m, length(t));                                       % Pre-allocation for speed
    COSTACT = zeros(m, length(t));                                    % Pre-allocation for speed
    % Store values in seperate matices
    for p = m:m:length(WorkDone)
        T_W(:, p/m) = WorkDone(p-(m-1):p,1);
        T_W_tot = sum(T_W,2);
        WORK(:,p/m) = WorkDone(p-(m-1):p,3);
        WORKD(:,p/m) = WorkDone(p-(m-1):p,2);
        WORKD_tot = sum(WORKD,2);
        COSTACT(:,p/m) = Cost_ACT(p-(m-1):p);
    end
    end
    S_TOT(N) = S_tot;                    % Store total duration of run n in a vector
    C_TOT(N) = C_tot;                    % Store total cost of run n in a vector
    T_TOT(:,N) = T_W_tot;                % Store time steps in a vector
    PS = T_TOT(:,N);
    dev = sum(T_TOT(:,N));
    PS_TEST2(:,N) = PS ./ dev;
    WORKD_TOT(:,N) = WORKD_tot;
    WORKD_TOT(WORKD_TOT<1) = 1;
    RW_TOT = WORKD_TOT - ones(m,N);
    PS_RW(:,N) = RW_TOT(:,N)./sum(RW_TOT(:,N));
end
```

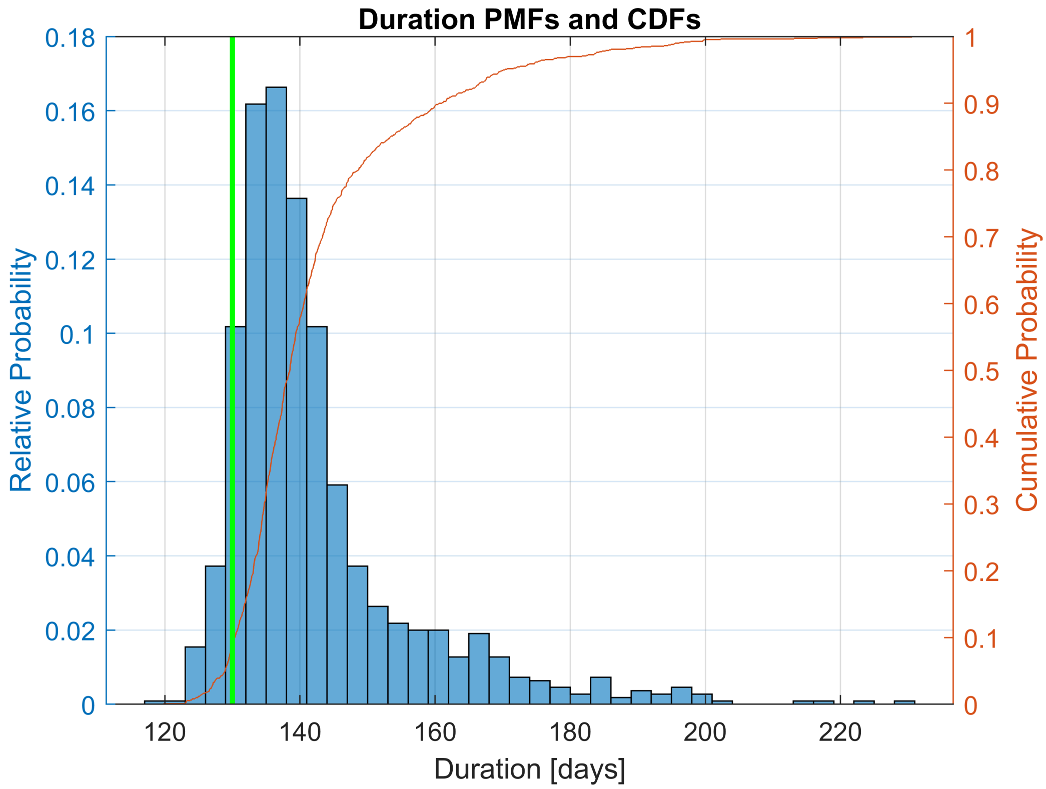
Simulation Output

```
OUTPUT = [C_TOT S_TOT];
writematrix( OUTPUT ,'OUTPUTV2.xlsx','Sheet',1,'Range','B2:C1102');
writematrix( S_D ,'OUTPUTV2.xlsx','Sheet',2,'Range','B2:O1102');
writematrix( S_C ,'OUTPUTV2.xlsx','Sheet',3,'Range','B2:O1102');

PS = T_TOT(:,2);
dev = sum(T_TOT(:,2));
PS_TEST = PS ./ dev;

% Test stability of simulation output
Cr = mean(C_TOT);
Crb = mean(C_TOT(1:end-100));
STAB_C = (Cr - Crb)/ Crb;
STAB_TEST_C = STAB_C < 0.01 ;

Sr = mean(S_TOT);
Srb = mean(S_TOT(1:end-100));
STAB_S = abs((Sr - Srb))/ Srb;
STAB_TEST_S = STAB_S < 0.01 ;

C_RW = RW_TOT.*S_C';
frac_rw_C = bsxfun(@rdivide, C_RW, C_TOT');
frac_rw_C (isnan(frac_rw_C)) = 0;
frac_rw_C = frac_rw_C.*100;

S_RW = RW_TOT.*S_D';
frac_rw_S = bsxfun(@rdivide, S_RW, S_TOT');
frac_rw_S (isnan(frac_rw_S)) = 0;
```

```
frac_rw_S = frac_rw_S.*100;

pmf_S = hist(S_TOT, max(S_TOT))' ./ numel(S_TOT);
```

RISK

```
% Calculate Schedule and Budget Risk
S_Risk = S_TOT > T_s;
S_r_sorty = S_TOT .* S_Risk;
RISK_S = ((1/n).*S_Risk).*((S_r_sorty - T_s).^2);
RR_S = sum(RISK_S);

C_Risk = C_TOT > T_c;
C_r_sorty = C_TOT .* C_Risk;
RISK_C = ((1/n).*C_Risk).*((C_r_sorty - T_c).^2);
RR_C = sum(RISK_C);

% Generate Simulation Output Plots
figure(1)
yyaxis left
histogram(S_TOT, 'BinWidth',3, 'Normalization','probability')
yyaxis right
cdfplot(S_TOT)
yyaxis left
title('Duration PMFs and CDFs')
xlabel('Duration [days]')
ylabel('Relative Probability')

yyaxis right
ylabel('Cumulative Probability')
hold on;
line([T_s, T_s], ylim, 'LineWidth', 2, 'Color', 'g');
```



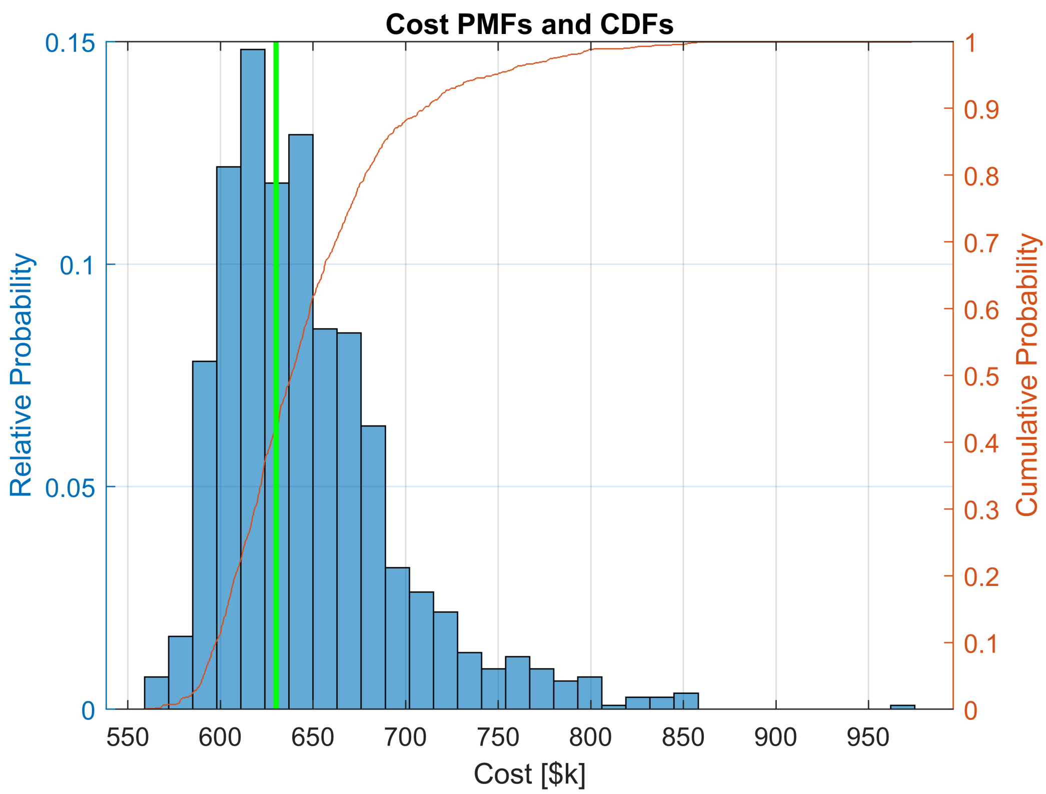

```
[f_S,x_S] = ecdf(S_TOT);
P_acc_S= intersections(f_S, x_S, [0 1] ,[T_s T_s]);
P_unacc_S = 1 - P_acc_S;
MS = mean(S_TOT)  % Mean duration [days] -- Browning 2002 found 138 days
```

```
MS = 142.3252
```

```
std_S = std(S_TOT) % Standard deviation [days] -- Browning 2002 found 14 days
```

```
std_S = 14.0132
```

```
figure(2)
yyaxis left
histogram(C_TOT,'BinWidth',13,'Normalization','probability')
yyaxis right
cdfplot(C_TOT)
yyaxis left
title('Cost PMFs and CDFs')
xlabel('Cost [$k]')
ylabel('Relative Probability')

%xlim([540 870])
yyaxis right
ylabel('Cumulative Probability')
hold on;
line([T_c, T_c], ylim, 'LineWidth', 2, 'Color', 'g');
```



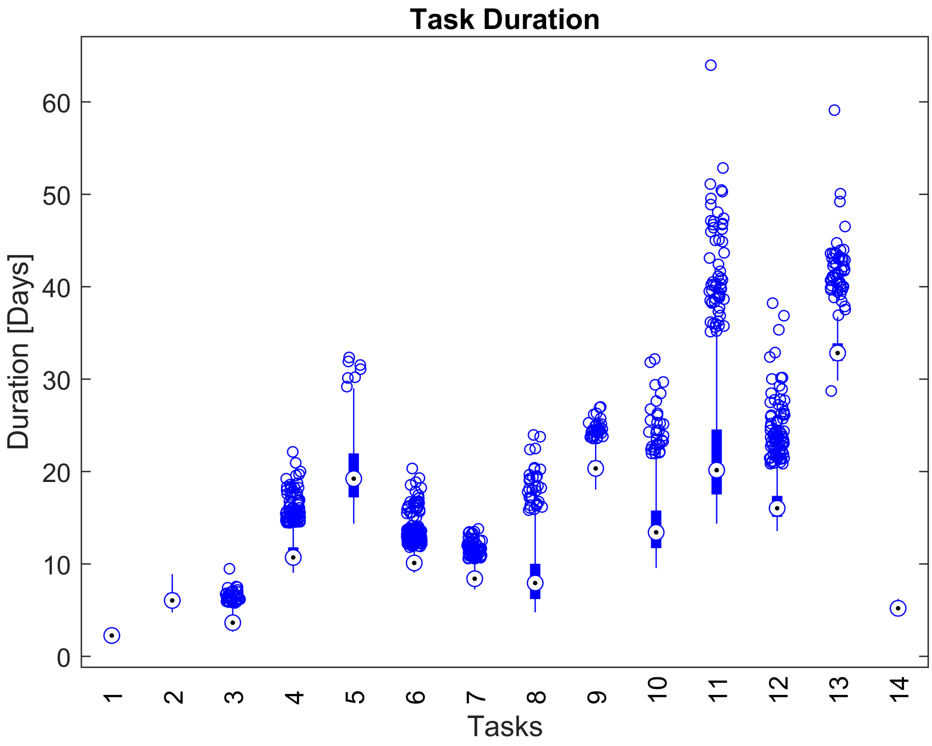

```
MC = mean(C_TOT) % Mean cost [$k] -- Browning 2002 found 637 [$k]
```

```
MC = 647.6500
```

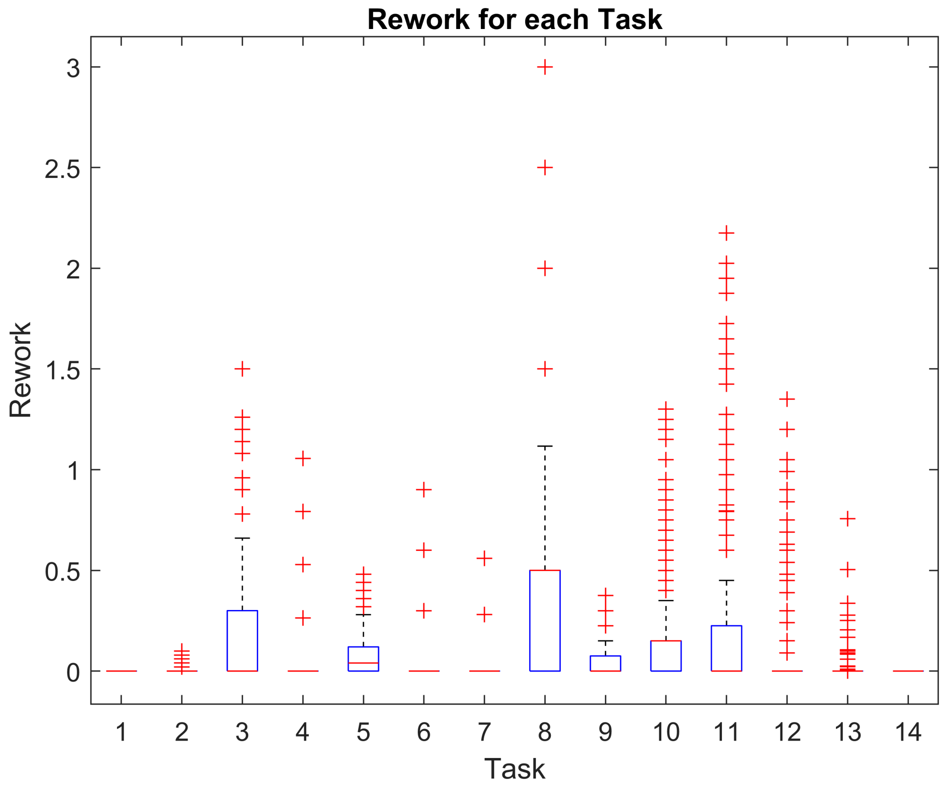
```
std_C = std(C_TOT) % Standard deviation [$k] -- Browning 2002 found 63 [$k]
```

```
std_C = 49.0086
```

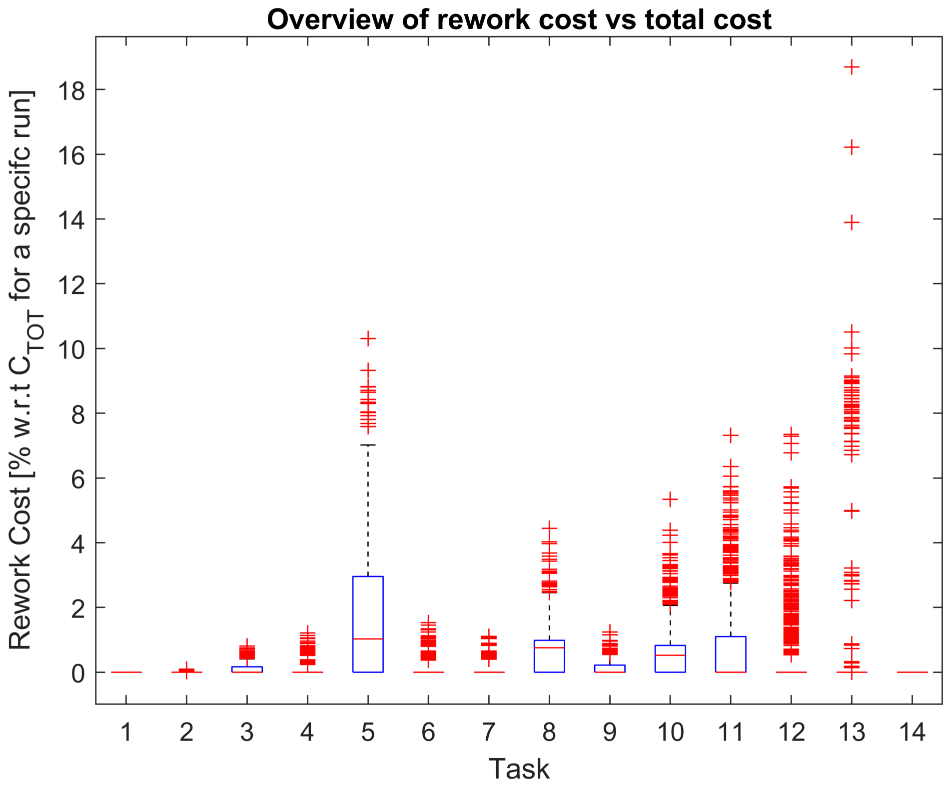
```
[f_C,x_C] = ecdf(C_TOT);
P_acc_C= intersections(f_C, x_C, [0 1] ,[T_c T_c]);
P_unacc_C = 1 - P_acc_C;

figure(3)
boxplot(T_TOT', 'PlotStyle','compact')
title('Task Duration')
xlabel('Tasks')
ylabel('Duration [Days]')
```



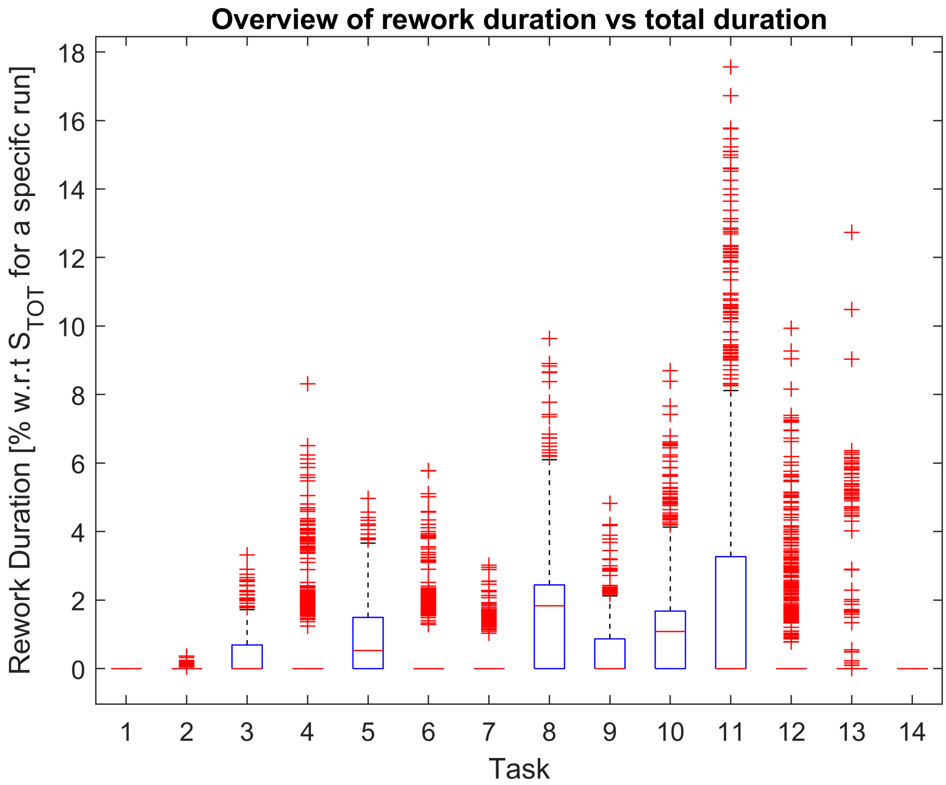

```
figure(4)
boxplot(RW_TOT')
title('Rework for each Task')
xlabel('Task')
ylabel('Rework')
```



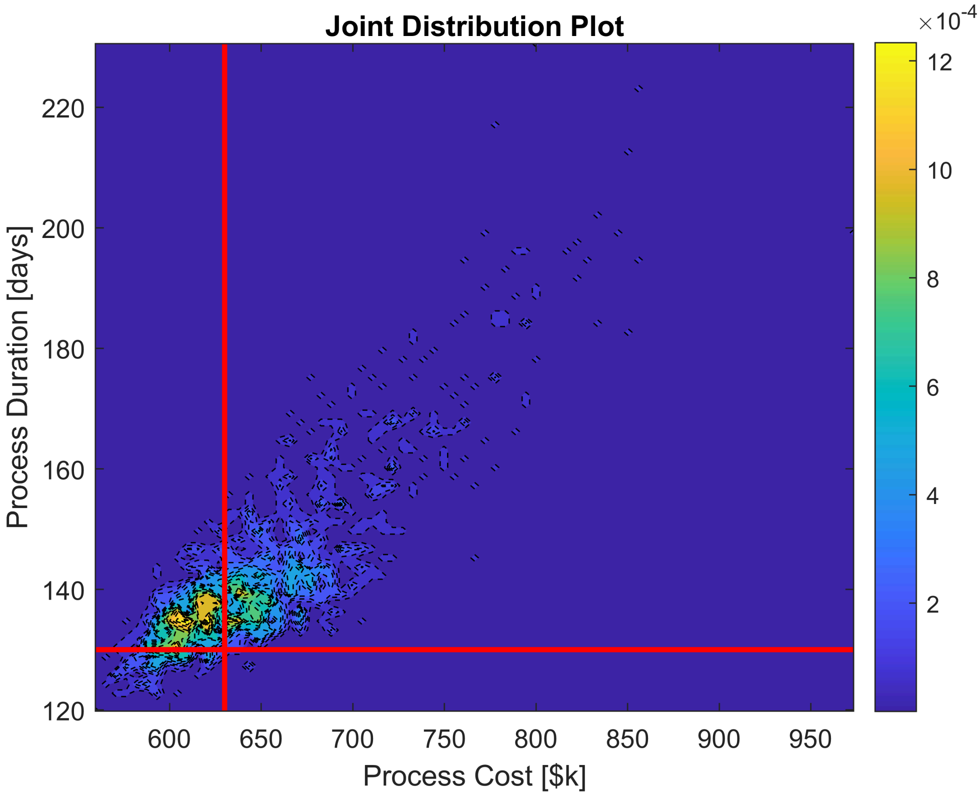

```
figure(5)
boxplot(frac_rw_C')
title('Overview of rework cost vs total cost')
xlabel('Task')
ylabel('Rework Cost [% w.r.t C_{TOT} for a specifc run]')
```



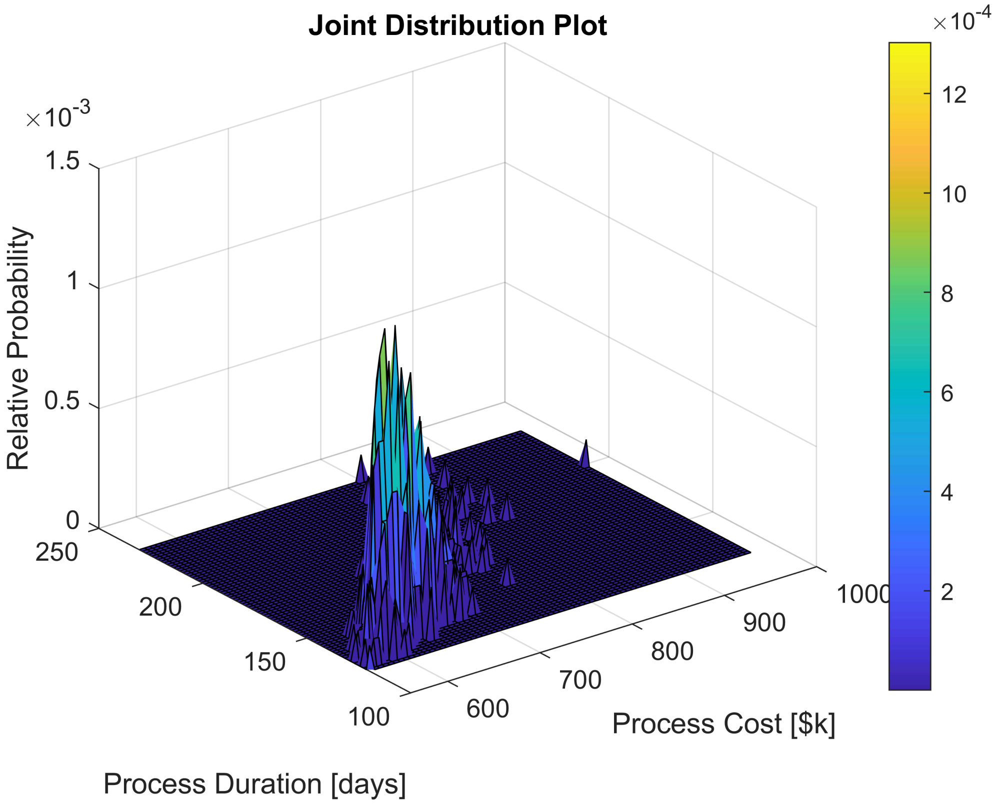

```
figure(6)
boxplot(frac_rw_S')
title('Overview of rework duration vs total duration')
xlabel('Task')
ylabel('Rework Duration [% w.r.t S_{TOT} for a specifc run]')
```




```
CALC_BIN = bin(S_TOT);
[rho, xvec, yvec] = density(C_TOT, S_TOT);
figure(7)
contourf(xvec, yvec, rho, 18, '--') %':'
colorbar
hold on
line([T_c, T_c], ylim, 'LineWidth', 2, 'Color', 'r')
hold on;
line(xlim, [T_s, T_s], 'LineWidth', 2, 'Color', 'r')
title('Joint Distribution Plot')
xlabel('Process Cost [$k]')
ylabel('Process Duration [days]')
hold off
```




```
[rho, xvec, yvec] = density(C_TOT, S_TOT);

figure(8)
surf(xvec, yvec, rho)
title('Joint Distribution Plot')
xlabel('Process Cost [$k]')
ylabel('Process Duration [days]')
zlabel('Relative Probability')
colorbar
```


Joint Distribution Plot
×10^-3
1.5
1
0.5
0
Relative Probability
250
200
150
100
Process Duration [days]
600
700
800
900
1000
Process Cost [$k]
×10^-4
12
10
8
6
4
2

B.3. Analysis of the effect of used assumptions for input duration uncertainty

Scenarios Output

```
clc
close all
b = 100;
T_s = 100;
P_val = 0.55;
```

Case Study Test Assumptions for Uncertainty in Initial Duration

```
S_TOT_CS_in = importdata("S_TOT_CS_A1500.mat");
S_TOT_CS = (S_TOT_CS_in ./1291) .*100;

STAB_m = abs(mean(S_TOT_CS(1:end)) - mean(S_TOT_CS(1:end-b)))/mean(S_TOT_CS(1:end-b))
```

```
STAB_m = 3.5636e-04
```

```
STAB_v = abs(var(S_TOT_CS(1:end)) - var(S_TOT_CS(1:end-b)))/var(S_TOT_CS(1:end-b))
```

```
STAB_v = 0.0035
```

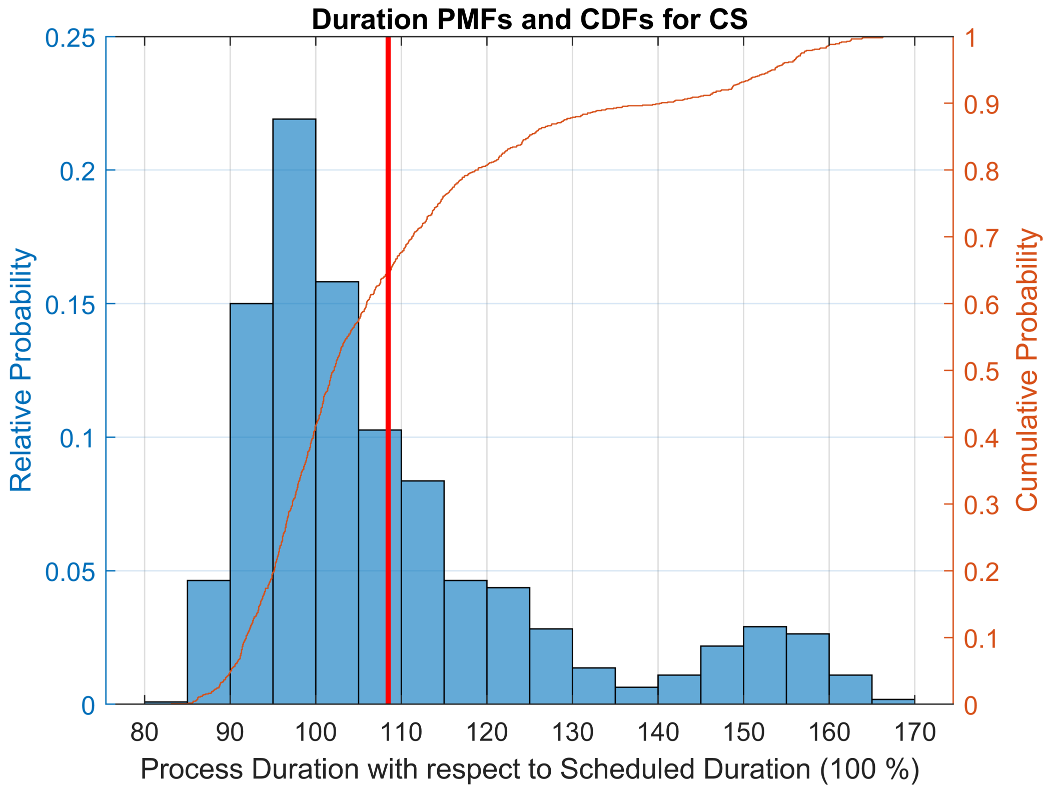
```
CS_Risk = S_TOT_CS_in > 903;
CS_r_sorty = S_TOT_CS_in .* CS_Risk;
RISK_CS = ((1/length(S_TOT_CS)).*CS_Risk).*((CS_r_sorty - 903));
RR_CS = sum(RISK_CS)
```

```
RR_CS = 497.2718
```

```
figure(1)
yyaxis left
histogram(S_TOT_CS, 'Normalization','probability') % 'BinWidth', 25,
yyaxis right
cdfplot(S_TOT_CS)
yyaxis left
title('Duration PMFs and CDFs for CS')
xlabel('Process Duration with respect to Scheduled Duration (100 %)')
ylabel('Relative Probability')
MS_CS = mean(S_TOT_CS)  % Mean duration [days] -- Browning 2002 found 138 days
```

```
MS_CS = 108.4641
```

```
yyaxis right
ylabel('Cumulative Probability')
hold on;
line([MS_CS, MS_CS], ylim, 'LineWidth', 2, 'Color', 'r');
```




```
[f_CS,x_CS] = ecdf(S_TOT_CS);
P_acc_CS= intersections(f_CS, x_CS, [0 1] ,[T_s T_s]);
P_unacc_CS = 1 - P_acc_CS
```

```
P_unacc_CS = 0.5818
```

```
RWF = MS_CS / T_s;
std_CS = std(S_TOT_CS) % Standard deviation [days] -- Browning 2002 found 14 days
```

```
std_CS = 18.2307
```

Scenario 1: Waiting Test Assumptions for Uncertainty in Initial Duration

```
b = 100 ;
T_s = 100;
S_TOT_S1_in = importdata("S_TOT_W_1300.mat");
S_TOT_S1_in(1301:end) = [];
S_TOT_S1 = (S_TOT_S1_in ./1291).*100;

STAB_S1_m = abs(mean(S_TOT_S1(1:1300)) - mean(S_TOT_S1(1:1300-b)))/mean(S_TOT_S1(1:1300-b))
```

```
STAB_S1_m = 0.0017
```

```
STAB_S1_v = abs(var(S_TOT_S1(1:1300)) - var(S_TOT_S1(1:1300-b)))/var(S_TOT_S1(1:1300-b))
```

```
STAB_S1_v = 1.2004e-04
```

```
S1_Risk = S_TOT_S1_in > 903;
S1_r_sorty = S_TOT_S1_in .* S1_Risk;
RISK_S1 = ((1/length(S_TOT_S1)).*S1_Risk).*((S1_r_sorty - 903));
RR_S1 = sum(RISK_S1)
```

```
RR_S1 = 494.0000
```

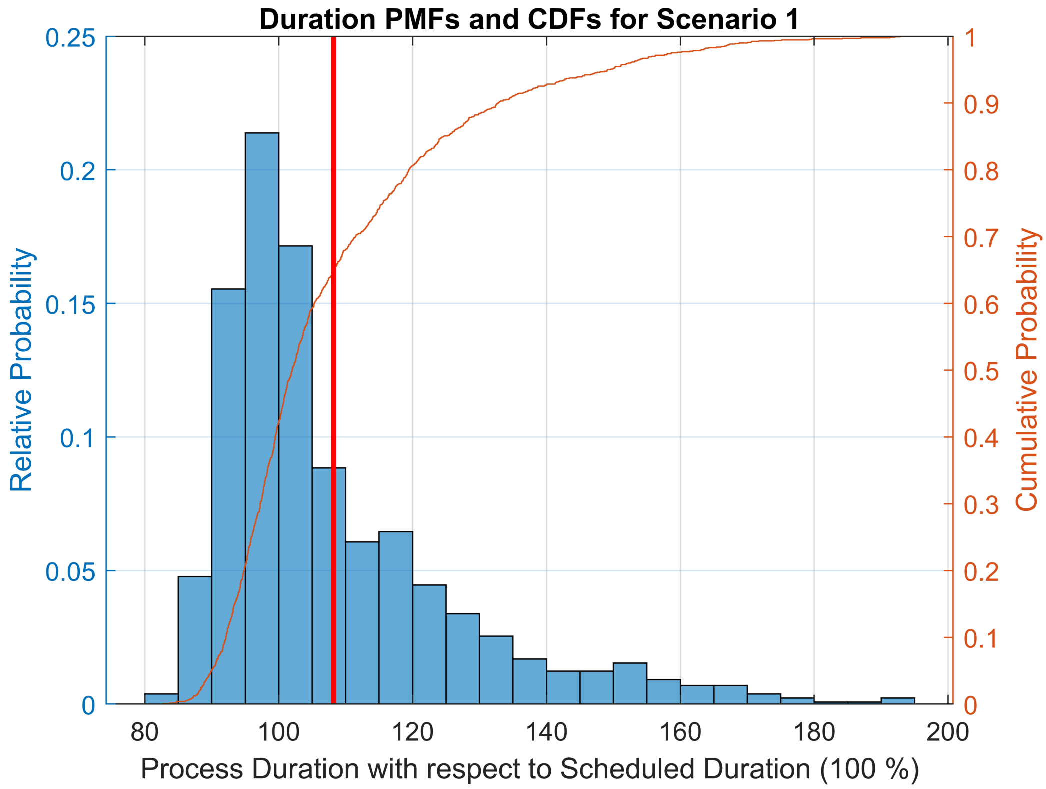
```
S1_Risk_INC = S_TOT_S1 > 70;
S1_r_sorty_INC = S_TOT_S1 .* S1_Risk_INC;
RISK_S1_INC = ((1/length(S_TOT_S1)).*S1_Risk_INC).*((S1_r_sorty_INC - 70));
RR_S1_INC = sum(RISK_S1_INC)
```

```
RR_S1_INC = 38.2107
```

```
figure(2)
yyaxis left
histogram(S_TOT_S1,'Normalization','probability') % 'BinWidth', 25,
yyaxis right
cdfplot(S_TOT_S1)
yyaxis left
title('Duration PMFs and CDFs for Scenario 1')
xlabel('Process Duration with respect to Scheduled Duration (100 %)')
ylabel('Relative Probability')
MS_S1 = mean(S_TOT_S1)  % Mean duration [days] -- Browning 2002 found 138 days
```

```
MS_S1 = 108.2107
```

```
yyaxis right
ylabel('Cumulative Probability')
hold on;
line([MS_S1, MS_S1], ylim, 'LineWidth', 2, 'Color', 'r');
```




```
[f_S1,x_S1] = ecdf(S_TOT_S1);
P_acc_S1= intersections(f_S1, x_S1, [0 1] ,[T_s T_s]);
P_unacc_S1 = 1 - P_acc_S1
```

```
P_unacc_S1 = 0.5762
```

```
RWF = MS_S1 / T_s;
std_S = std(S_TOT_S1) % Standard deviation [days] -- Browning 2002 found 14 days
```

```
std_S = 18.1489
```

Scenario 2: Assumptions Test Assumptions for Uncertainty in Initial Duration

```
b = 100 ;
T_s = 100;
S_TOT_S2_in = importdata("S_TOT_S2_A.mat");
S_TOT_S2_in(1101:end) = [];
S_TOT_S2 = (S_TOT_S2_in ./1291).*100;

STAB_S2_m = abs(mean(S_TOT_S2(1:1100)) - mean(S_TOT_S2(1:1100-b)))/mean(S_TOT_S2(1:1100-b))
```

```
STAB_S2_m = 0.0020
```

```
STAB_S2_v = abs(var(S_TOT_S2(1:1100)) - var(S_TOT_S2(1:1100-b)))/var(S_TOT_S2(1:1100-b))
```

```
STAB_S2_v = 0.0055
```

```
S2_Risk = S_TOT_S2_in > 903;
S2_r_sorty = S_TOT_S2_in .* S2_Risk;
RISK_S2 = ((1/length(S_TOT_S2)).*S2_Risk).*((S2_r_sorty - 903));
RR_S2 = sum(RISK_S2)
```

```
RR_S2 = 481.6109
```

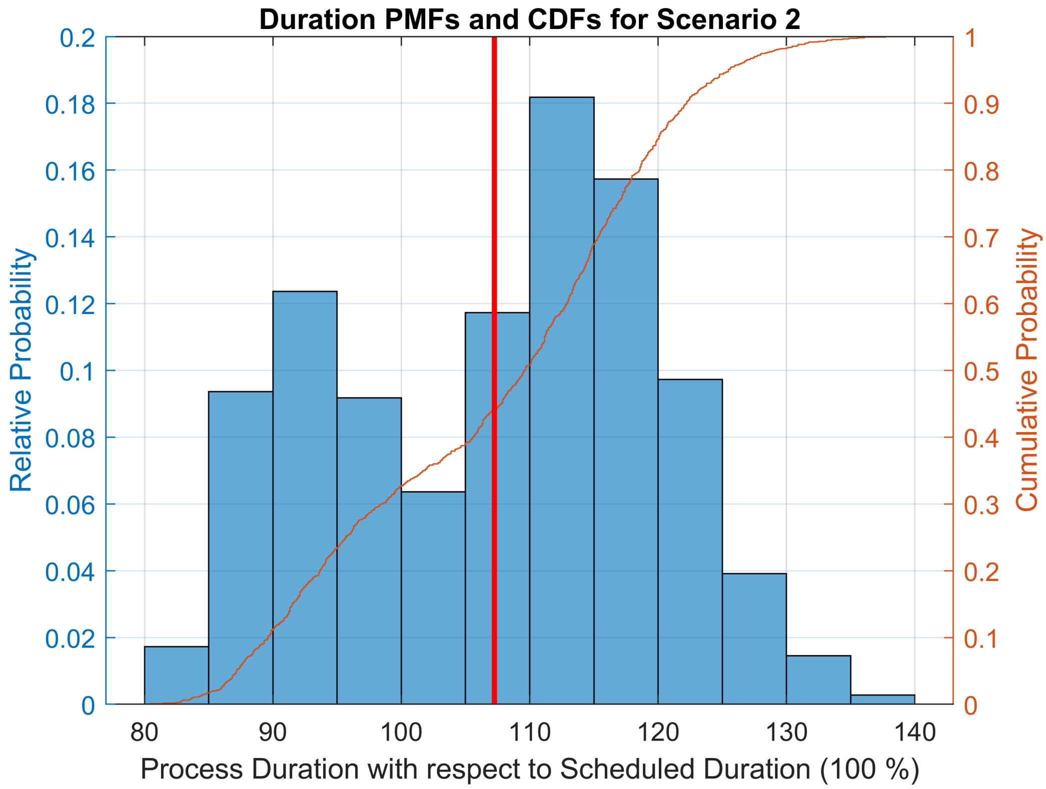
```
S2_Risk_INC = S_TOT_S2 > 70;
S2_r_sorty_INC = S_TOT_S2 .* S2_Risk_INC;
RISK_S2_INC = ((1/length(S_TOT_S2)).*S2_Risk_INC).*((S2_r_sorty_INC - 70));
RR_S2_INC = sum(RISK_S2_INC)
```

```
RR_S2_INC = 37.2510
```

```
figure(3)
yyaxis left
histogram(S_TOT_S2,'Normalization','probability') % 'BinWidth', 25,
yyaxis right
cdfplot(S_TOT_S2)
yyaxis left
title('Duration PMFs and CDFs for Scenario 2')
xlabel('Process Duration with respect to Scheduled Duration (100 %)')
ylabel('Relative Probability')
MS_S2 = mean(S_TOT_S2)  % Mean duration [days] -- Browning 2002 found 138 days
```

```
MS_S2 = 107.2510
```

```
yyaxis right
ylabel('Cumulative Probability')
hold on;
line([MS_S2, MS_S2], ylim, 'LineWidth', 2, 'Color', 'r');
```




```
[f_S2,x_S2] = ecdf(S_TOT_S2);
P_acc_S2= intersections(f_S2, x_S2, [0 1] ,[T_s T_s]);
P_unacc_S2 = 1 - P_acc_S2
```

```
P_unacc_S2 = 0.6727
```

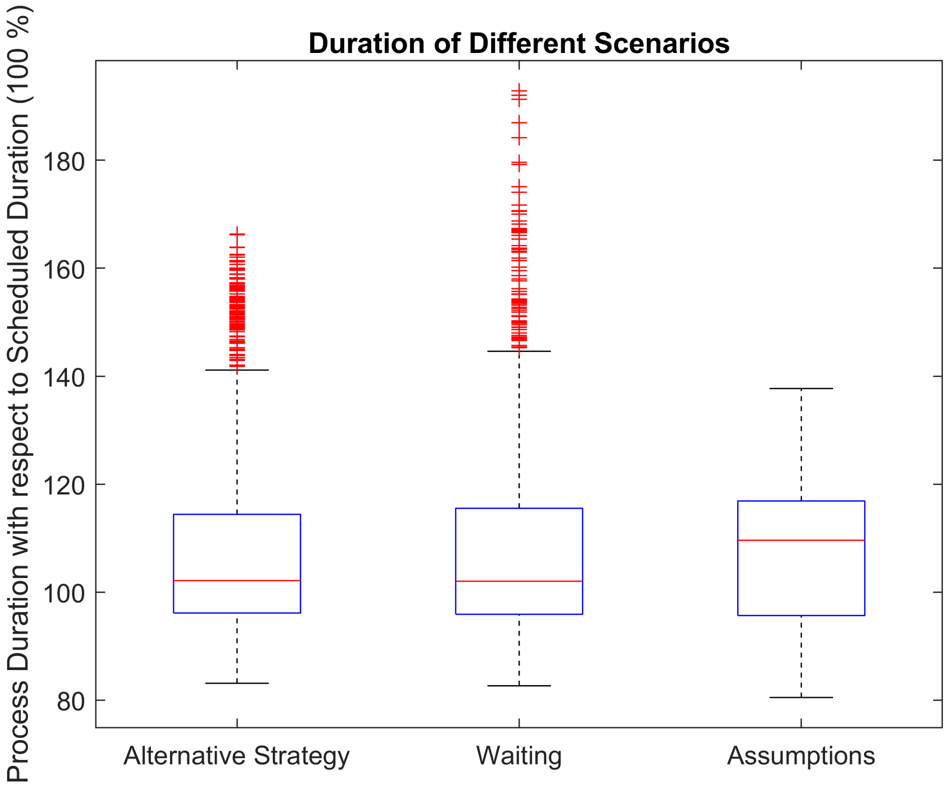
```
RWF = MS_S2 / T_s;
std_S = std(S_TOT_S2) % Standard deviation [days] -- Browning 2002 found 14 days
```

```
std_S = 12.4141
```

Compare Outcome of 3 Approaches

```
figure(4)
%rng('default')  % For reproducibility
x1 = S_TOT_CS;
x2 = S_TOT_S1;
x3 = S_TOT_S2;
x = [x1; x2; x3];

g1 = repmat({'Alternative Strategy'},1100,1);
g2 = repmat({'Waiting'},1300,1);
g3 = repmat({'Assumptions'},1100,1);
g = [g1; g2; g3];
```

```
boxplot(x,g)
title('Duration of Different Scenarios')
ylabel('Process Duration with respect to Scheduled Duration (100 %)')
```



C

An algorithm to generate project-specific DSM matrices for documentation flow simulation in complex shipbuilding projects

An Algorithm to Generate Project-Specific DSM Matrices for Documentation Flow Simulation in Complex Shipbuilding Projects

Anne Reumer, *Delft University of Technology*, annereumer@gmail.com
Jeroen Pruyn, *Delft University of Technology*, J.F.J.Pruyn@tudelft.nl
Mathijs van Schaik, Damen Shipyards, mathijs.van.schaik@damen.com

***Abstract*—Shipbuilding projects are characterised by a large number of (inter)dependant tasks. Today, many companies are involved in the shipbuilding process, which need to share large amounts of data with each other. The large amounts and the inter dependencies make shipbuilding project complex. All parties strive to process integration, which requires process understanding. Information flow modelling is a helpful tool to gain process understanding. In shipbuilding information is stored in project documentation, which makes documentation flow analysis relevant for shipbuilding. A method for such analysis is the design structure matrix (DSM), which is widely used to study the complex dependency network in processes. However, the required input for such analysis, three squared matrices, are hard to obtain. This difficulty is the current most considerable drawback for application of the DSM method. These martices also are project-specific, which makes this method less suitable for the comparison of various processes. This paper proposes an algorithm to generate project-specific DSM matrices for various projects of the same type. The algorithm is successful applied in a case study on a complex shipbuilding project.**

***Index Terms*—DSM, documentation flow modelling, modular, non project-specific**

I. Introduction

A drastic change is seen in shipbuilding processes of European companies over the past decades. Traditionally one company was producing a vessel, today the number of involved companies has increased significantly [1], [2]. This change resulted in a separation between engineering and production. Also, activities are executed in parallel. All combined created new and unique challenges for shipbuilding. Under these challenges, the complexity of the information flow and the difficulty to manage an engineer-to-order project increased.

Many studies have been conducted to understand the impact of these challenges in shipbuilding. Held argues that 60-80% of the value of a vessel is outsourced [3]. According to, Andritos are shipbuilding operations characterised by the generation and manipulation of large amounts of information [4]. Andritos also states that *'even vessels from the same series differ somewhat from each other'* [4]. Mello and Strandhagen observed that building existing designs at different yards creates new issues [5]. Also, the lack of proper Information and Communication Technology (ICT) systems is blamed for these issues [4]–[7].

In complex projects, it is difficult to manage interactions, because the large number of tasks and people, it is hard to predict the impact of a single change [5], [8]. Besides, rework is seen as the most critical factor affecting project success [9]. Elimination of this adverse effect requires process integration [10], [11]. Subsequently, process integration requires process understanding. With the increased complexity in processes, process understanding became a challenge. Information flow modelling is a valuable tool to gain process understanding [10], [12], [13]. Baldwin et al. [14] claim that only a better understanding of the information flow among participants can improve management. Austin et al. [15] discovered that a lack of knowledge of information flow and dependencies lead to disturbed process flows. For projects, information is stored on project documentation. Thus the documentation flow needs to be modelled.

Over the years, various documentation flow modelling methods are implemented. However, many researchers claim that most models are not suitable for project processes [8], [13], [16]–[18]. These models often have problems representing interdependencies and feedback loops. Eppinger et al. [8] and Browning [19] demonstrates that the Design Structure Matrix is a suitable method to model documentation flow in complex projects [8]. Although, the collection of the required input matrices is a time-consuming task. These matrices are project-specific which increases data collection effort when multiple projects are analysed. This paper aims to find a method to generate multiple project-specific DSM matrices for various projects while keeping data collection effort limited.

First previous work on methods that reduce data collection effort is analysed, and a gap is identified. Next, a method is proposed to fill this gap. Then data is collected to use this method for a shipbuilding project. Finally, potential extensions to this method and conclusions are discussed.

II. Literature

The Design Structure Matrix (DSM) is a method to capture dependencies among activities for processes. Figure 1 illustrates the DSM method, all activities are placed on both the rows and the columns, and marks identify dependencies. Various advanced DSM analysing methods exists to study this complex network of dependencies. The DSM-based discrete-event Monte Carlo is one of them. Similar to other DSM analysing methods three matrices are required: the Binary DSM, rework probability DSM and rework impact DSM. The construction of these matrices is a time-consuming and iterative process [20]. Both the data collection and the creation of the matrices are a burden on resources. These arising challenges motivated researches to develop methods to make creating DSM matrices more efficient and accurate. An overview of the proposed methods is given, and a gap is identified.

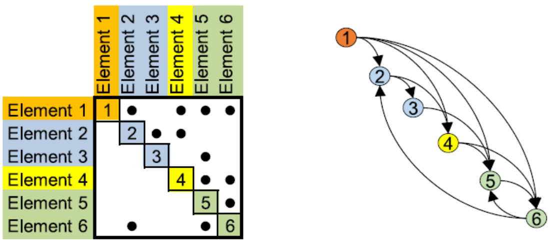


Fig. 1. Illustration of the DSM method [20]

Some see potential in the use of existing documentation for data collection, for instance, the supplier-input-process-output-customer (SIPOC) diagram. Browning [21] shows that the SIPOC diagram can be used to construct a binary DSM matrix. The benefits of the SIPOC diagram for DSM matrix creation are also demonstrated in a role-play to open participants eyes to the challenges of and potentials of proper process integration [13].

Others argue to facilitate data collection through workshops. Kasperek et al. [22] strive for reproducibility of process models by a framework. The use of a documentation template also is suggested to capture information generated within modelling activities. Moon et al. [23] propose a group decision framework for the dependency structure of a complex system based on expert surveys. The use of the decision framework must eliminate influencing factors on decision-making [23]. A DSM Tool, DiMo, is developed to generate DSM matrices during workshops [24]. The tool is incredibly valuable for applications with limited time and resources [24].

The application of the quality function deployment (QFD) in the area of DSM construction is also seen. Chen and Huang [24] use the QFD approach to quantify dependencies among design tasks in a systematic way. This approach makes it also possible to estimate the coupling strength of tasks by using the full information contained in the QFD [24]. Others use the QFD approach to reduce the number of engineering-related elements in the DSM [25]. Kaldate et al. [25] argue that only engineering elements with maximum impact should gain attention.

Others argue that the challenges are caused by the used models to plan and manage work in the process. So argues Browning [26] that models for product design are more sophisticated than models for planning and managing technical work. Process flowcharts, Gantt Charts and work-break-down structures are examples of models used in technical work [26]. The use of multiple models causes challenges when constructing DSM matrices. However, an architecture framework is proposed to manage multiple process views in one model [26]. Later, this work was extended by additional views [28]. The work aims to provide project managers and other project users with complete and straightforward process data. Another study developed an algorithm for a bidirectional transformation between a product-project system and its corresponding DSM [29].

Over the years, different methods and tools are proposed aiming to simplify the data collection and the creation of DSM matrices. Despite the effort, these tasks are still a challenging task. A gap is identified in the proposed methods and tools, as all methods and tools are project-specific. Proposed tools are not helpful for DSM-based analysis of various projects of the same type. Since each project requires its DSM matrices, and this is still a demanding job. This paper aims to find a method to generate project-specific DSM matrices for similar project types while keeping data collection and creation effort limited.

III. Model Built up

Although organisational differences are present at different yards, all yards build vessels. These vessels are constructed from multiple sections, blocks and zones, which comes with reoccurring tasks. Based on this characteristic, it is expected to find a general set of tasks that describe shipbuilding projects. A general set of tasks helps to reduce data collection effect, because a task decomposition for specific projects is no longer needed. However, it is only useful if the general set can be transformed to a project-specific set. This section describes the construction of a model, which uses a general set of reoccurring tasks to generate project-specific DSM matrices for shipbuilding projects.

A. Model Requirements

A set of requirements is defined to ensure the development of a useful algorithm. In this context, a useful algorithm is an algorithm that makes it possible to use collected data for the creation of DSM matrices for various projects. Next, the algorithm must result in a straight forward way to determine input values. Also, must the algorithm must be user friendly for users that are not familiar with the entire process. Next, the algorithm must be able to generate DSM matrices for different shipbuilding projects. Different can refer to building location or design. However, engineer-to-order shipbuilding projects are the scope of this paper.

B. Model Algorithm

Essential input for the DSM-based discrete event Monte Carlo simulation are three project-specific matrices: *Binary*

DSM (DSM), *Matrix with Rework Probabilities (DSM_1)* and *Matrix with Rework Impact (DSM_2)*. The demanding task of the establishment of the DSM matrices is one of the largest drawbacks to this method. However, a method is developed to construct multiple project-specific DSM matrices in an easy manner. The only requirement for this method is a group of reoccurring tasks.

The first step is to categorise (reoccurring)project tasks in groups. Note that not all tasks need to be reoccurring; it is also possible to have a group of tasks that is only used once. It is recommended to use task groups for with their occurrence number is known at the beginning of a project. DSM matrices for these (reoccurring) groups are established, Basic DSM Matrices. These Basic DSM Matrices are the building blocks for the final DSM matrices constructed by the described algorithm. To discribe various projects of the same type it is essential that all vital activities are included in this Basic DSM Matrices. All Basic DSM matrices are put together in three matrices. One matrix for dependencies, one matrix for rework probabilities and one matrix for rework impact.

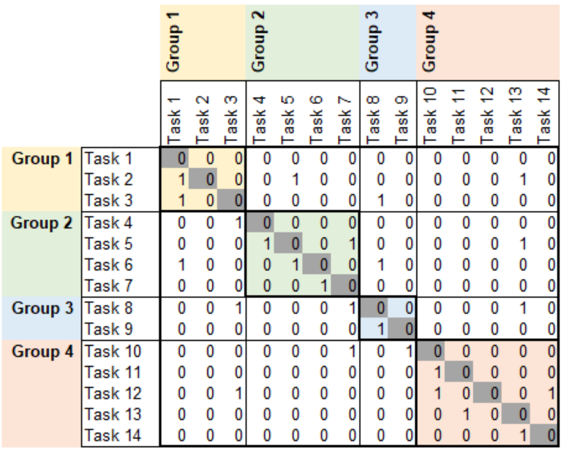


Fig. 2. Step 1: Standard DSM with DSM 'building blocks'

Subsequently, the number of DSM building blocks needs to be defined. This input is essential for the creation of the DSM matrices. It is recommended to use groups for which their number is known at the beginning of the project. When it is chosen for less straight forward grouping, this step takes more effort. It is recommended to avoid such groups. Table I gives an example of re occurrence numbers for four groups.

TABLE I
NUMBER OF DSM BUILDING BLOCKS AS AN EXAMPLE

	Number of DSM Building Blocks
Group 1	1
Group 2	3
Group 3	4
Group 4	1

Based on the size of the Basic DSM Matrices, shown in Figure 2 and the re-occurrence number of the various groups,

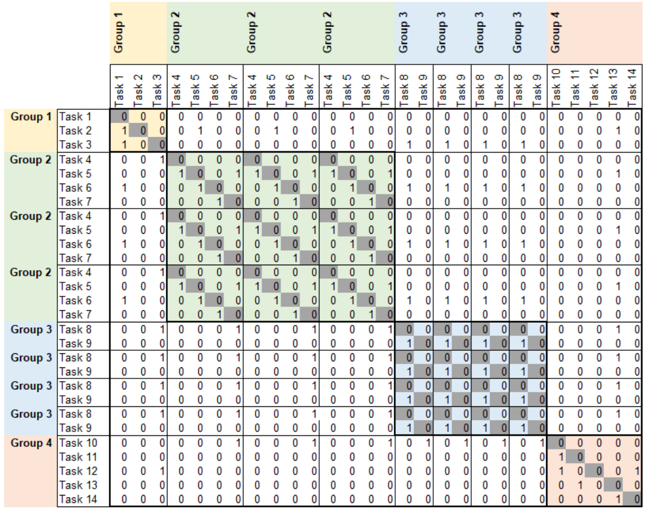


Fig. 3. Step 2: Empty DSM Matix filled with DSM Building Blocks from BLOCK-DSM, no dependencies are included jet

the size of the final DSM matrices is calculated. First three empty matrices of equal size are created. Basic DSM Matrices are printed into the empty matrices based on the re occurrence numbers. Figure 3 displays printed Basic DSM matrices at predefined places. At this moment no dependencies among groups are present. For instance, all marked green sub-matrices are equal in Figure 3. The lack of dependencies does not only concern marked sub-matrices but also concerns sub-matrices that connect groups. For example, all sub-matrices that connect group 3 to group 4 are equal in Figure 3.

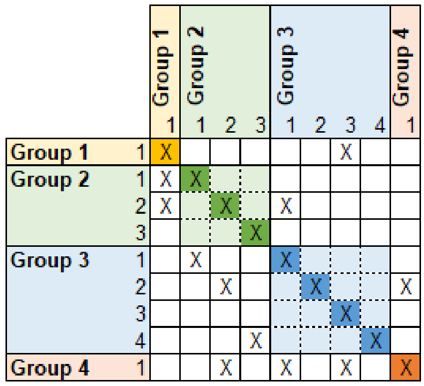


Fig. 4. Step 3: Dependencies among DSM Building Groups Stored in a Separate Matrix

Eventually, dependencies among DSM building blocks are implemented. Information among these dependencies is stored in a separate matrix, shown by Figure 4. It is important that one noticed that this matrix does not only stores links among different groups but also has marks on the diagonal. Diagonal marks are essential in this algorithm, since omitting these links results in empty colour-marked sub-matrices.

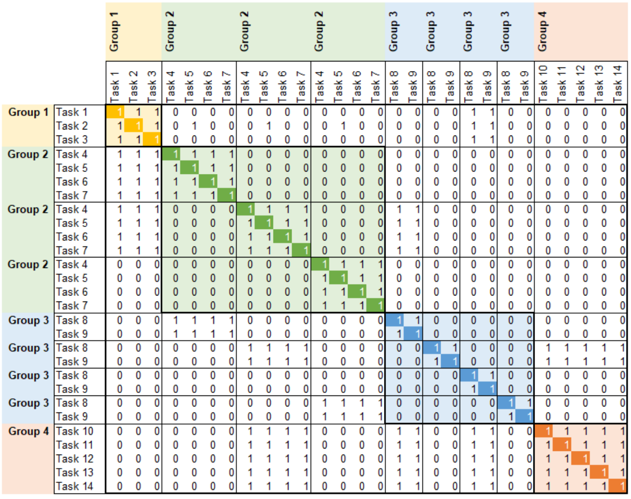


Fig. 5. Step 4: Dependencies among DSM Building Groups Converted to a mxm Matrix

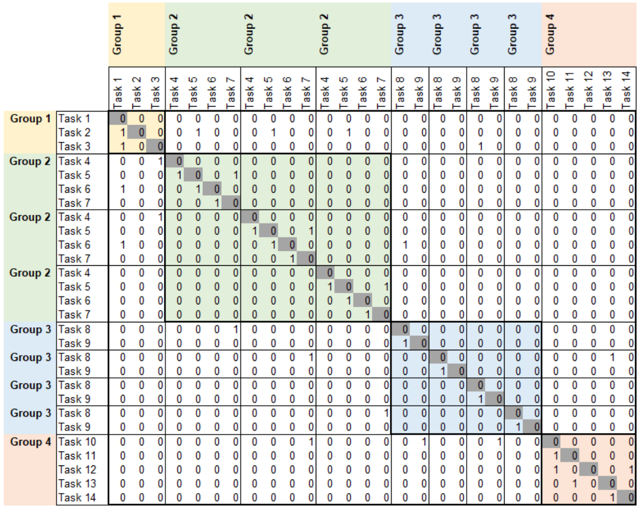


Fig. 6. Step 5: Final DSM Matrix

In a subsequent step, the links of Figure 4 are converted to an mxm matrix. Entities containing an 'X' are transformed into sub-matrices with ones. Empty entities are transformed into sub-matrices of zeros. Figure 3 displays this conversion. This transformed matrix is used to included dependency information in the matrix shown by Figure 3. Equation 1 demonstrates how a final DSM matrix is obtained. It should be mentioned that element-wise multiplication is used.

$$DSM_{final} = DSM_{step2} .* DSM_{step4} \tag{1}$$

Equation 1 results in an mxm DSM matrix, including dependency data among the used DSM building blocks. Figure 6 illustrates this result. The algorithm can be used for all three DSM matrices. The only difference is the input in the Standard DSM Matrix given in Figure 2.

C. Required Input

The described algorithm makes it possible to construct DSM Matrices in a relatively simple way for processes which can be defined by several reoccurring tasks. Before this algorithm can be used for a shipbuilding project, particular input is required.

First of all, a task decomposition is the most important. Without a well-defined decomposition, the algorithm is not helpful. As mentioned before it is vital to use reoccurring groups when decomposing a process. This requirement implies that this algorithm is less suitable for the inclusion of a high detailed task decomposition. From experience is learned that the higher the used detail, the harder it becomes to find reoccurring groups to describe the process.

Besides, dependencies must be identified. Dependencies serve two functions in this context and are, therefore, crucial input. First dependencies define which sub-matrices are printed in the final DSM matrices. Second, dependencies are an input value for the first DSM matrix. Thus dependencies do not only affect one DSM matrix but also the construction of all other DSM matrices. Next to the multiple functions of dependencies, are two dependency types addressed for this application: dependencies among activities and dependencies among activity groups. Because dependencies have both influences on the construction of the DSM matrices by the algorithm and serve as one of the input values of the DSM matrix dependencies are a crucial input value.

Rework probability, *the likelihood an activity has rework due to a typical change in its input*, is defined by Equation 2 Browning2002. Further is rework affected by the likelihood for input changes and the probability that such changes affect the activity. Equation 3 describes this relationship Browning2002.

$$\begin{aligned}&\text{P(rework for activity i caused by a typical}\\&\text{change in its input from activity j)} = \boldsymbol{DSM}_{ij1}\end{aligned} \tag{2}$$

$$\begin{aligned}&\text{P(rework for an activity caused by change in one}\\&\text{of its inputs)} = \text{P(change in the input)}\\&\cdot\text{P(the change affecting the activity)}\end{aligned} \tag{3}$$

Rework probability does not influence the construction of the DSM matrices. However, it is still vital input for a DSM-based discrete-event Monte Carlo simulations as it defined the frequency rework is generated. For such analysis, rework probabilities have a significant influence on simulation outcome.

The consequence of rework can differ from task to task. Robust activities experience fewer consequences of changes, than sensitive activities. Rework impact is defined as *'the percentage of the activity that must be reworked'*, Equation 4, Browning2002. Each dependency link among activities has its rework impact value.

$$\begin{aligned}&\text{\%(rework for activity i caused by a typical}\\&\text{change in input from activity j)} = \boldsymbol{DSM_{ij2}}\end{aligned} \tag{4}$$

Also, rework impact does not influence the construction of the DSM matrices. Nevertheless, it is critical input for a DSM-based discrete-event Monte Carlo simulations or equivalent. Rework impact defines the consequence of rework and therefore influences the analysis.

IV. Data Collection

Different data collection methods are used to collect the required input to use the algorithm in a study on the effect of the quality of project documentation on project execution of complex shipbuilding projects. Used methods are discussed for each input value.

TABLE II
PARTICIPANTS FOR INTERVIEWS FOR EACH DEPARTMENT

Department	**Function**
Project Management	former Project Manager
Engineering	Manager Engineering, Project Manager Engineering
Material Coordination	Logistics Process Manager, Team Leader Material Coordination
Procurement	Coordinator Project Procurement
Production	Site Managers working in: Vietnam, Israel and South Africa

A. Standard Binary DSM Matrix

Task decomposition and dependency identification are required to construct the binary DSM matrix. As shipbuilding concerns, different projects at multiple yards, a non-yard and non-project specific decomposition is needed. Although organisational differences are present at different yards, all yards build vessels. So it is expected to find a non-specific decomposition for projects within the scope of this paper.

First, an initial high-level task decomposition is made from building schedules, project plans and other process documentation. In this high-level decomposition, the process is decomposed into departments, shown by Figure 7.

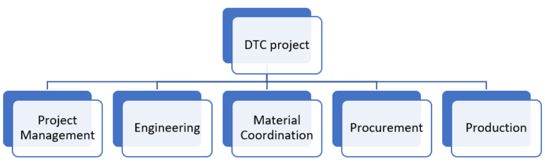


Fig. 7. Decomposition based on departments

The next step is to define tasks and dependencies at a more detailed level through interviews. Interviews are based on shared question forms to keep the conversations structured and efficient. Interviews are performed using Microsoft Teams and have an approximated duration of 90 minutes. Sometimes more time is needed, and a new moment is scheduled. *Experience learned that long (>90 min) Microsoft Teams meetings are less effective than two shorter (<90 min) meetings*. During the interviews, send-in answers are discussed, remaining questions are handled, and notes are taken of relevant relating issues.

After each interview, a DSM structure for each department is created. Links with other departments or the yard are mentioned. Eventually, all department DSMs are collected, and a standard DSM structure is created for the binary DSM. In this way, the DSM structure is not project-specific. The DSM structure is made project-specific by the algorithm. Figure 8 gives an overview of the task decomposition at a high level.

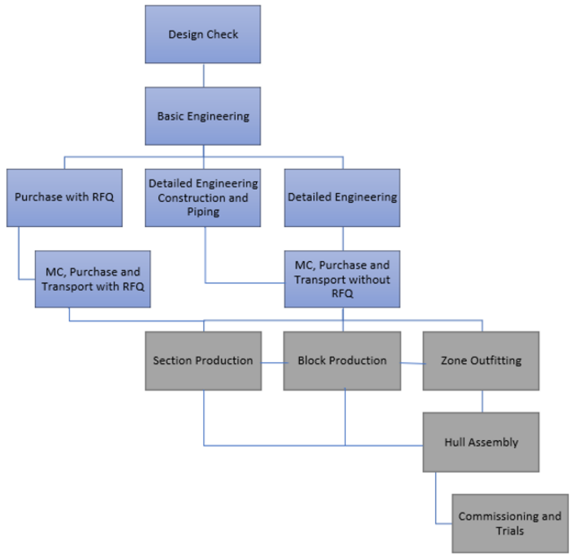


Fig. 8. High-level task decomposition of a case study project, grey blocks represent yard activities

B. Standard Rework Probability DSM Matrix

For each dependency link in the binary DSM, a rework probability value exist. These values are crucial simulation input. Even though it is crucial input, assigning rework probabilities is challenging. Yassine et al. define two reasons why it is challenging to define accurate values for rework probabilities: 1) engineers and designers are generally uncomfortable providing direct probabilities, and 2) provided estimates vary widely [30] . Input quality defines simulation output quality. Thus a reliable method to determine rework probabilities is essential. A secure method is found in the *Rework Probability Assessment* [30].

With the assessment, direct rework probability values are no longer required. Rework probability is described by *Information Variability (IV)* and *Task Sensitivity (TS)*. *IV stands for the likelihood that information will change and TS stands for the sensitivity of a task for such change*. IV and TS values are assigned from a predefined scale presented by Tables III and IV. Multiplication of IV and TS results in TV values, Equation 5. *TV is not a probability; it represents relative probability to other tasks on a scale from 1 to 9*. A DSM-based simulation is required to convert TV values to the probability domain.

$$TV = IV \times TS \tag{5}$$

TABLE III
LEVELS OF INFORMATION VARIABILITY (IV) AS MENTIONED IN [30]

Value	Description	Likelihood of Change
1	Stable	Low
2	Unknown	Medium to high
3	Unstable	Very high

TABLE IV
LEVELS OF TASK SENSITIVITY (TS) AS MENTIONED IN [30]

Value	Description	Dependent Task is:
1	Low	Insensitive to most information changes
2	Medium	Sensitive to major information changes
3	High	Sensitive to most information changes

C. Standard Rework Impact DSM Matrix

Rework impact represents the consequence of rework, which can differ from task to task. Each link in the binary DSM has its rework impact. Thus rework impact on a specific task depends on the task that causes rework. *It is assumed that rework impact is a fixed number*. So it does not change with the number of iterations. For shipbuilding projects, it is hard to assign a rework impact value. Rework consequences are influenced by time and the severity of rework. Compared to literature, rework impact values for shipbuilding project tasks are small. Rework on a section will not lead to 30%-50% of the initial work. From interviews is concluded that values around 1%-5% are common.

A data collection survey asking for values between 0% and 100% is not suitable for the required accuracy. It is not precise due to small values and the fact that a percentage is a relative value. Nonetheless, rework impact is an essential input. Misplaced rework impact values affect process duration and cost. So assigning rework impact values demands attention. These characteristics make the data collection process for rework impact difficult and timely.

Similar activities are grouped to reduce time. For each group, a rework impact value in hours is proposed. For example, one value is assigned to all activities of Basic Engineering. Sometimes not enough information is available to argue a proposed value and the field is kept empty. Trough interviews proposed impact is discussed. Remaining values are identified by the (in general) required hours for typical rework on an activity. Also, this method does not result in exact numbers suitable for all relevant shipbuilding projects. However, *it is essential that order of magnitude matches*. Minor deviations have a low impact on simulation outcome due to the scaling process of rework probabilities.

V. CASE STUDY

The algorithm and the general data is used in a study of the effect of the quality of project documentation on project execution of complex shipbuilding projects. Generating project-specific DSM matrices is part of this case study. Required actions to obtain these matrices are highlighted in this paper.

A. Project-Specific Data Collection

Project specific data collection includes dependencies among DSM Building Groups, Figure 4. Links are identified in project schedules, project data or interviews. Experience learned that it is easier to identify links among groups than among specific activities.

B. Scale Rework Probability

Rework probabilities differ from project to project, thus scaling is required. Earlier rework is described by Task Volatility (TV). However, TV values are not probability values, but relative values on a scale from 1 to 9. So, TV values need to be transferred to matching probability values for the shipbuilding project. Scaling is done by the procedure described by [30]. Figure 9 illustrates the procedure.

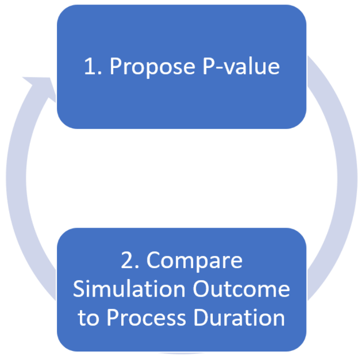


Fig. 9. Procedure for Rework Probability Scaling with a Known Mean Process Duration

1) **Run simulation with small P-value**
   A P-value of 10% is used as starting point
2) **Compare simulated duration to mean process duration**
   Mean process duration for the shipbuilding project is scheduled duration.
3) **Adjust P-value if simulated duration does not match mean process duration**
   A matching P-value is a value that results in a mean process duration that deviates less than 1%.

C. Project Specific DSM Matrices

In the case study the algorithm is successfully used to generate the DSM matrices. Although only one project is used for this case study it is demonstrated that the algorithm can generate DSM matrices for other projects in a simple manner. Data collection effort is reduced as only project-specific data is required to generate project-specific DSM matrices for various projects.

VI. RECOMMENDATIONS FOR FUTURE WORK

This paper describes the development of an algorithm for the generation of project-specific DSM matrices for documentation flow simulation using a DSM-based discrete-event Monte Carlo simulation. The algorithm achieves the set requirements.

However, the algorithm can be extended to ensure a wider applicability.

A. Impact on Performance

The creation of the algorithm to generate project-specific DSM matrices for documentation flow simulation in complex shipbuilding projects was a time-consuming task. Fortunately, the approach reduces data collection effort for the simulation of other shipbuilding projects. In general projects that can be described by the same set of activities benefit from the algorithm. The algorithm is the most beneficial for applications where it is desired to compare multiple projects to each other. Because then it becomes possible to use the collected data for more then one project.

B. Extend the Algorithm

The current algorithm does not include dependencies among DSM building blocks of the same group. From interviews is concluded that including links within groups is not necessary to study the shipbuilding project. However, allowing dependencies of the same group can extend the applicability and detail level of the DSM-based discrete-event Monte Simulation. Thus with the potential extension, the simulation can be used for projects where links within groups play a crucial role.

However, the inclusion requires addition dependency information. Since collecting dependency information is a time-consuming task, extending the algorithm with dependencies within groups is only wanted if the benefits outweigh the effort. Links within groups can be beneficial for a more detailed study for several similar projects.

C. Algorithm Application

In this paper, the algorithm is applied to one project. Therefore it is required to scale TV-values to rework probabilities to create project-specific input. However, the use ability of the algorithm is higher when scaling is not required as it eliminates an additional step. It should be mentioned that having a rough idea of rework probability values at the beginning of a project is valuable for early risk analysis. Thus, the possibility to eliminate scaling is beneficial even though it has a limited effect on the accuracy of the rework probability values.

A database can be created once the algorithm is used on a variety of projects. In this database, projects can be grouped in categories. Each category can be linked to a specific transformation from TV values to rework probabilities. However, this requires additional research on how to categorise projects.

VII. Conclusion

The generation of DSM matrices is a time-consuming and demanding task. Matrices also are project-specific. Thus, the effort can not be spread over multiple projects. However, DSM matrices are crucial input for advanced DSM analysis such as the DSM-based discrete-event Monte Carlo simulation. Documentation flow simulation allows rework-analysis to study the effect of documentation quality on project execution. With a straightforward manner to construct DSM matrices for multiple projects, it becomes possible to spread data collection effort. The approach also makes DSM-based discrete-event Monte Carlo simulations useful for industrial applications.

A simple method is found to construct DSM matrices by defining shipbuilding activities in (reoccurring)groups. However, to benefit from the simple method, groups must not be project-specific. It must be able to determine the number of reoccurring groups at the project start. Dependencies among activities within the same group are not included in this version. Nevertheless, this can be included by small adjustments.

The algorithm is developed for engineer-to-order projects in shipbuilding. However, the method can also be used for projects with similar characteristics. The identification of reoccurring groups is seen as the essential factor of this method.

To conclude, the algorithm described in this paper provides a simple way to construct project-specific DSM matrices for complex shipbuilding projects. The method is successful applied in a case study. Application on multiple projects is a next step in the research on the application of the algorithm. Potential is seen for the application on early risk analysis in the future.

References

[1] R. Levering, R. Ligthart, N.Noorderhaven and L. Oerlemans "Continuity and change in interorganizational project practices: The Dutch shipbuilding industry, 1950–2010", International Journal of Project Management, vol. 31, pp735-747, july, 2013.

[2] M.H. Mello, J.O. Strandhagen, and E. Alfnes "Analyzing the factors affecting coordination in engineer-to-order supply chain", International Journal of Operations and Production Management, vol. 35, issue 7, pp1005-1031, july, 2015

[3] T. Held, "Supplier Integration as an Improvement Driver – An Analysis of Some Recent Approaches in the Shipbuilding Industry" in Supply Chain Network Management, C. Engelhardt-Nowitzki, O. Nowitzki, and H. Zsifkovits, Eds. Gabler, 2010, pp369-384.

[4] F. Andritos and J. Perez-Prat, "The Automation and Integration of Production Processes in Shipbuilding" European Commision Joint Research Centre, June, 2000.

[5] M.H. Mello and J.O. Strandhagen "Supply chain management in the shipbuilding industry: challenges and perspectives", Engineering for the Maritime Environment, vol. 225, issue 3, pp261-270, aug, 2011.

[6] Z. Sladoljev "Search for a Model of Effective Ship Production Management", Journal of Ship Production, vol. 12, issue 4, pp220-229, nov, 1996.

[7] M. Fleischer, R. Kohler,T. Lamb, and H. Bongiorni, "Shipbuilding Supply Chain Integration Project", Environmental Research Institute of Michigan, Interim Report, nov, 1998.

[8] S.D. Eppinger, D.E. Whitney, R.P Smith, and D.A. Gebala "A model-based method for organizing tasks in product development", Research in Engineering Design, vol. 6, pp1-13, march, 1994.

[9] S.D. Eppinger and T.R. Browning Design Structure Matrix Methods and Applications. MIT Press, ISBN: 978-0-262-01752-7, 2012.

[10] V.D. Hunt, Process Mapping. John Wiley and Sons, Inc., ISBN: 0471-13281-0, 1996.

[11] S.D. Eppinger and T.R. Browning "Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development", Transactions on Engineering Management, vol. 49, issue 4, pp428-442, nov, 2002.

[12] S. W. Ambler, The object primer : agile model-driven development with Uml 2.0, 3rd ed. London : Cambridge University Press, 2004.

[13] T. R. Browning, "A Role-Playing Game for Teaching about Enterprise Process Integration," Journal of Enterprise Transformation, vol. 4, no. 3, pp. 226–250, Aug. 2014, doi: https://doi.org/10.1080/19488289.2013.879682.

[14] A. N. Baldwin, S. A. Austin, T. M. Hassan, and A. Thorpe, "Modelling information flow during the conceptual and schematic stages of building design," Construction Management and Economics, vol. 17, no. 2, pp. 155–167, Mar. 1999, doi: https://doi.org/10.1080/014461999371655.

[15] S. Austin, A. Baldwin, and A. Newton, "A Data Flow Model to Plan and Manage the Building Design Process," Journal of Engeering Design, vol. 7, no. 1, pp. 3–25, 1996, doi: 10.1080/09544829608907924.

[16] S. Cho and S. D. Eppinger, Product Development Process Modeling Using Advanced Simulation. DETC'01, 2001.

[17] A. Yassine and D. Braha, "Complex Concurrent Engineering and the Design Structure Matrix Method," Concurrent Engineering, vol. 11, no. 3, pp. 165–176, Sep. 2003, doi: 10.1177/106329303034503.

[18] A. A. Oloufa, Y. A. Hosni, M. Fayez, and P. Axelsson, "Using DSM for modeling information flow in construction design projects," Civil Engineering and Environmental Systems, vol. 21, no. 2, pp. 105–125, Jun. 2004, doi: https://doi-org.tudelft.idm.oclc.org/10.1080/10286600310001638474.

[19] T. R. Browning, "Modeling and Analyzing Cost, Schedule, and Performancein Complex System Product Development," phdthesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1998.

[20] Browning, "Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities," IEEE Transactions on Engineering Management, vol. 63, no. 1, pp. 27–52, Feb. 2016, doi: 10.1109/TEM.2015.2491283.

[21] Browning, "Process Integration Using the Design Structure Matrix," Systems Engineering, vol. 5, no. 3, pp. 180–193, Jun. 2002, doi: https://doi-org.tudelft.idm.oclc.org/10.1002/sys.10023.

[22] D. Kasperek, S. Maisenbacher, A. Kohn, U. Lindemann, and M. Mauer, "Increasing the reproducibility of structural modelling," Journal of Engineering Design, vol. 26, no. 7-9, pp. 259–281, May 2015, doi: https://doi.org/10.1080/09544828.2015.1026883.

[23] J. Moon et al., "Group Decision Procedure to Model the Dependency Structure of Complex Systems: Framework and Case Study for Critical Infrastructures," Systems Engineering, vol. 18, no. 4, pp. 323–338, Jul. 2015, doi: https://doi.org/10.1002/sys.21306.

[24] N. Halonen, T. Juuti, and Luostarinen, "DSM MADE EASY," in International Design Conference, Dubrovnik, Croatia, 2012, pp. 1445–1454.

[25] Chen and Huang, "A systematic approach for supply chain improvement using design structure matrix," Journal of Intelligent Manufacturing, vol. 18, no. 2, pp. 285–299, Apr. 2007, doi: https://doi.org/10.1007/s10845-007-0022-z.

[26] A. Kaldate, D. Thurston, H. Emamipour, and M. Rood, "Engineering parameter selection for design optimization during preliminary design," Journal of Engineering Design, vol. 17, no. 4, pp. 219–310, 2006, doi: https://doi.org/10.1080/09544820500274027.

[27] T. R. Browning, "The many views of a process: Toward a process architecture framework for product development processes," Systems Engineering, vol. 12, no. 1, pp. 69–90, Aug. 2009, doi: https://doi.org/10.1002/sys.20109.

[28] T. R. Browning, "Managing Complex Project Process Models with a Process Architecture Framework," International Journal of Project Management, vol. 32, no. 2, pp. 229–241, May 2013, doi: https://doi.org/10.1016/j.ijproman.2013.05.008.

[29] A. Sharon, O. L. de Weck, and D. Dori, "Improving Project–Product Lifecycle Management with Model–Based Design Structure Matrix: A joint project management and systems engineering approach," Sytems Engineering, vol. 16, no. 4, pp. 413–426, Oct. 2013, doi: 10.1002/sys.21240.

[30] Yassine, Whitney, and Zambito, Assessment of Rework Probabilities for Simulating Product Development Processes Using the Design Structure Matrix (DSM). Computers and Information in Engineering Conference, 2001.

D

Detailed Rework Analysis

In this appendix the rework is analysed at a more detailed level. For this analysis, data is divided into four equal groups:

1. Minimum - 25th percentile
2. 25th percentile - median (50th percentile)
3. median (50th percentile) - 75th percentile
4. 75th percentile - maximum

Activities from the below-listed project phases are analysed.

1. Detailed engineering
2. Section production
3. Block production
4. Zone outfitting

The analysis uses Violin plots to visualise the rework distribution in the various groups and phases. Compared to box plots give Violin plots addition statistical information, since the probability distribution is vertical projected. Figure D.1 gives an example of an Violin plot. In this figure, a median value similar to the box plot is shown. The interquartile range (IQR) also is present. The Violin plots of the detailed analysis show median values by a red cross and the IQR by a green box. The vertical axis presents rework by the numbers 0-6; ‘1’ means that a tasks is completely reworked. It should be mentioned that this refers to the pure activity durations. Activities and their names are placed on the horizontal axis.

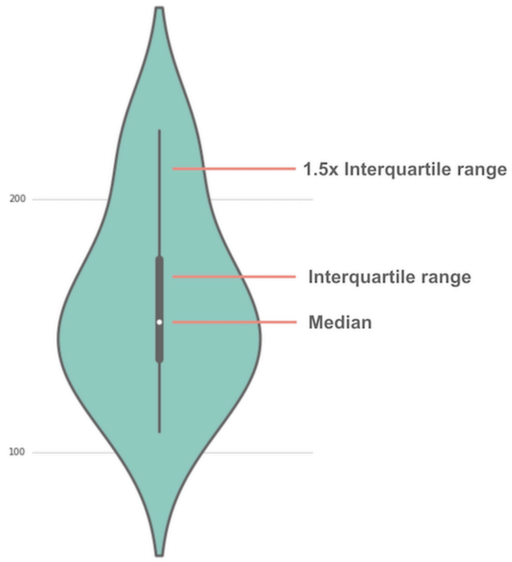


Figure D.1: Example of a Violin plot retrieved from https://mode.com/blog/violin-plot-examples/

D.1. Scenario 1: Delayed documentation - Waiting

Waiting on delayed deliverables scored second on the standard deviation. Thus this scenario comes with a relatively widespread in duration distribution. As mentioned before, this can relate to the circumstances of delayed activities. So different set of runs from Figure 4.13 are studied in detail. The aim is to identify circumstances or activities that cause negative effect on project execution when tasks are forced to wait for deliverables.

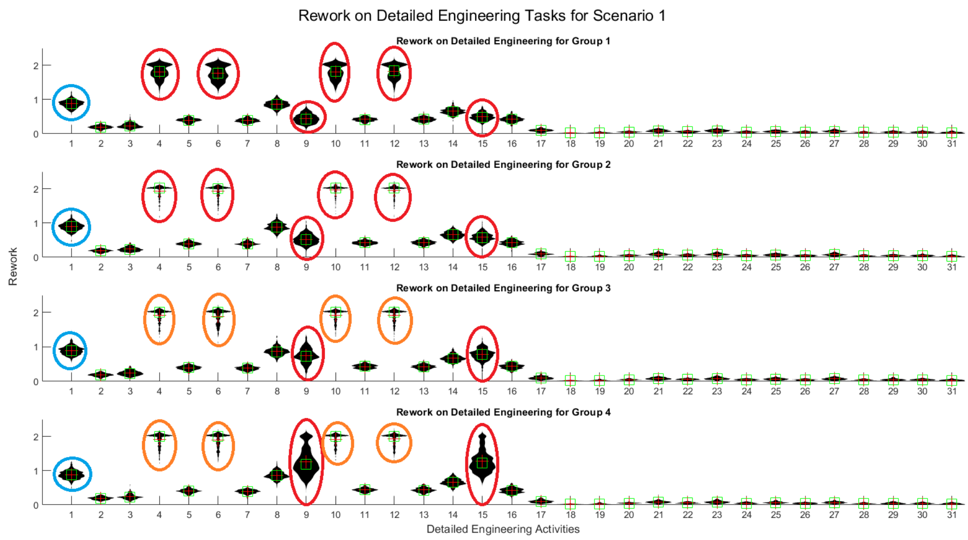


Figure D.2: Rework on Detailed Engineering Tasks for Scenario 1: Delayed Deliverables - Waiting

Figure D.2 shows the distribution of rework for the multiple engineering activities in the different groups. This figure is used to determine the impact of waiting on rework on Detailed Engineering activities. For this approach is stated that *activities are sensitive for waiting on delayed deliverables if rework on (detailed engineering) activities changes for the groups*. Activities that contribute to the outcome differences are emphasized with red ovals in Figure D.2. For some activities, rework is reported in Figure D.2, but this does not change for the group. Such tasks do not cause a difference in the project outcome. An example is task 1; almost no difference is present in the blue marked violin plots. Orange ovals point at a combination; activities contrite to output differences but not in the orange marked groups.

As the aim of this analysis is to identify activities that cause negative effect on project execution is focussed on marked red activities. Tasks 9 and 15 are red for all four groups and has the most impact on outcome differences of the detailed engineering tasks. This concerns:

- **Construction drawings**
- **Mechanical Diagrams**

Then four tasks contribute to output differences but have less effect than the earlier named activities. Included activities, 4,6,10 an 12, are marked with orange ovals.

- **Hull Outfitting Drawings**
- **Superstructure Outfitting Drawings**

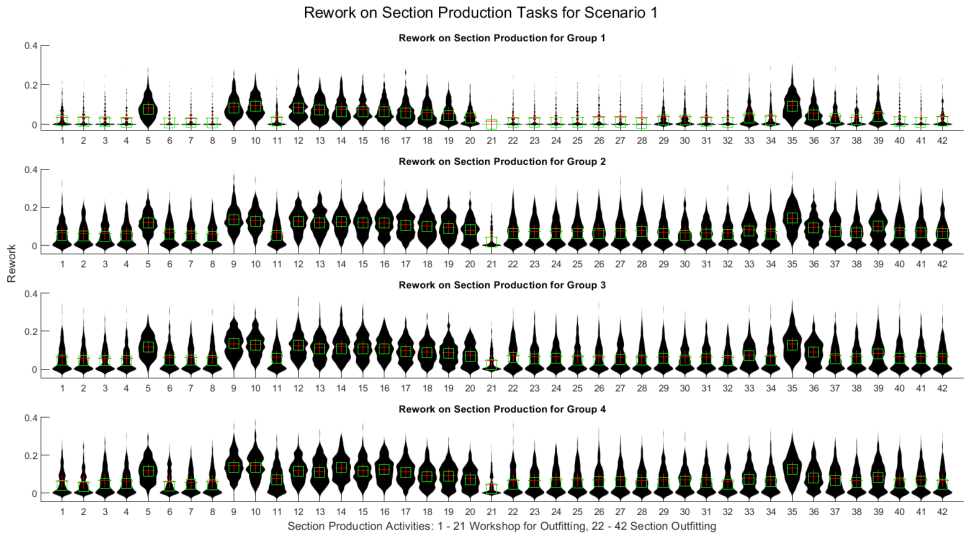


Figure D.3: Rework on section production tasks for scenario 1: Delayed documentation - Waiting

In contrast to rework on *Detailed Engineering* activities is rework on *Section Production* activities limited. Figure D.3 gives an overview of rework distribution on the activities for the various groups. It can be seen that reported rework is in the range 0-0.4, a large difference compared to rework ranges in Figures D.2, D.4 and D.5. Thus *it is not expected that rework on Section Production has large impact on outcome differences for a situation where activities are forced to wait on delayed deliverables*. However, in general more rework is reported for **Workshop for Section Outfitting** than for **Section Outfitting**.

Figure D.4 reports most rework for *Block Production* activities. However, rework distributions for *Workshop for Block Outfitting* shows a little change over the groups. Thus is concluded that rework on Workshop for Block Outfitting does not have an impact on output differences for this scenario.

Differences are noticed for *Block Outfitting Equipment* and *Block Outfitting Hot works*. Since no clear difference exists for rework on the different sections, only one section is highlighted with a red oval. Rework on tasks 8 and 15 is increasing with the groups. Thus *it is concluded that rework on Block Outfitting Equipment and Block Outfitting Hot works have influence on outcome difference in this situation*.

Figure D.5 shows rework distribution for **Zone Outfitting** activities. For all activities, an upward shift is reported with increasing group numbers. Since no big differences are seen between zones, only one zone is highlighted.

Of all outfitting tasks, most rework is reported for hot works. Also, rework on hot works increases most over the groups. Thus, hot works have a notable impact on heavy process overruns. Outfitting

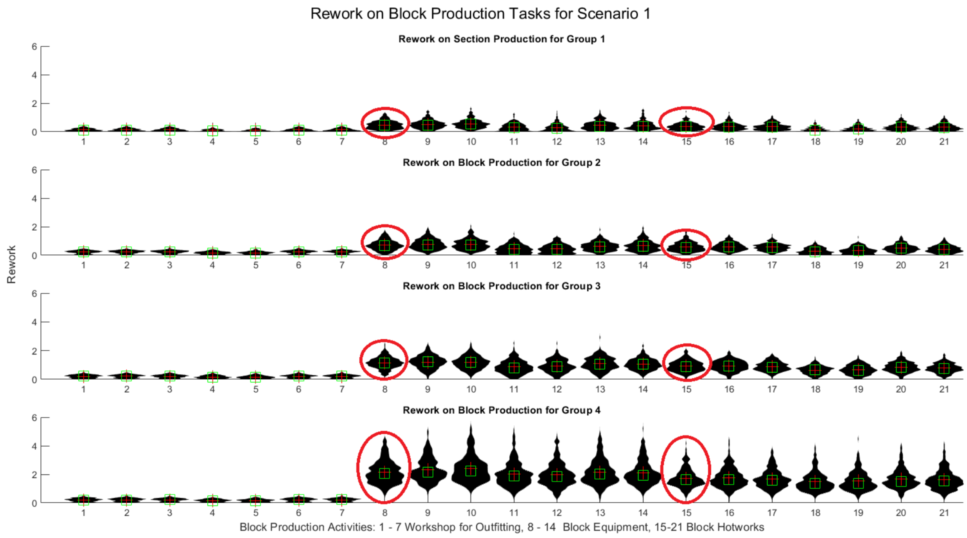


Figure D.4: Rework on block documentation for scenario 1: Delayed documentation - Waiting

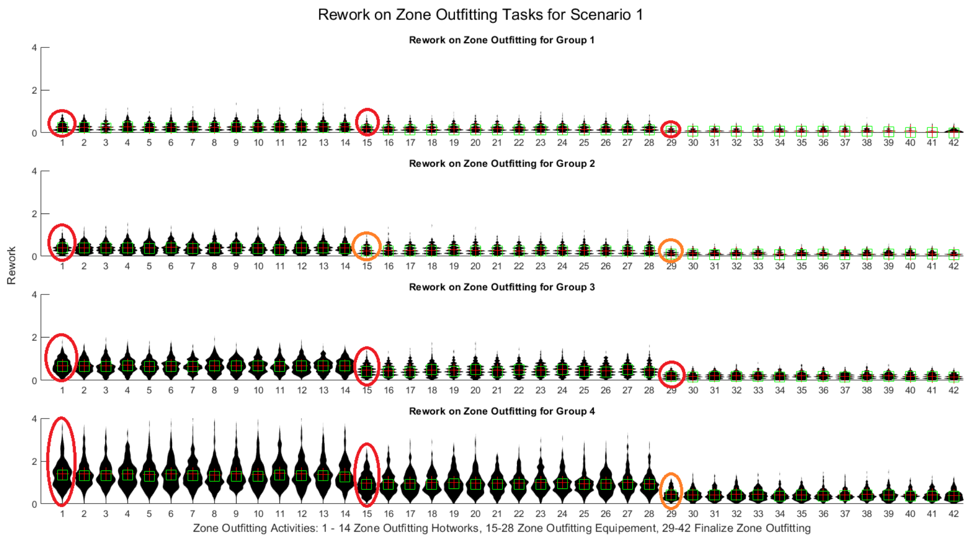


Figure D.5: Rework on zone outfitting tasks for scenario 1: Delayed documentation - Waiting

equipment and finalizing zone outfitting are highlighted with orange in group 2 as no apparent difference between group 1 and 2 is seen. Similarity implies that process outcomes in group 2 are not caused by an increase in rework on these tasks.

D.2. Scenario 2: Delayed documentation - Assumptions

From Figure 4.14 is concluded that using assumptions to start activities early has a various impact on project duration. Rework on Detailed Engineering, Section Production, Block Production and Zone Outfitting are analysed to get insight into the occurrence of severe effect on project duration.

The violin plots in Figure D.6 show the distribution of rework for the multiple engineering activities in the different groups. This figure is used to determine the impact of using assumptions to start delayed activities early on Detailed Engineering activities. For this approach is stated that *activities are sensitive for using assumptions to begin activities early if rework on (detailed engineering) activities changes for the groups*.

For analysis purposes, ovals with three colours are used. The blue ovals are activities that have rework, but rework does not change for the groups. These activities do not cause differences in process duration. Not all exercises are highlighted, but an example is given for two activities. Next are orange and red ovals seen in Figure D.6. Red ovals highlight activities with increasing rework for the groups. Orange ovals mark a stop in rework increase for that activity.

Carpentry Drawings, activity 2, is the only activity marked red for all groups. Carpentry Drawings includes 2D drawings of the accommodations, and floor and wall plans. This activity has the most impact of all *Detailed Engineering* activities on project duration outcome. Other red marked activities are similar to sensitive activities as named for scenario 1.

- **Hull Outfitting Drawings**
- **Superstructure Outfitting Drawings**

It should be noted that rework on these activities does not contribute to project outcomes in the fourth group.

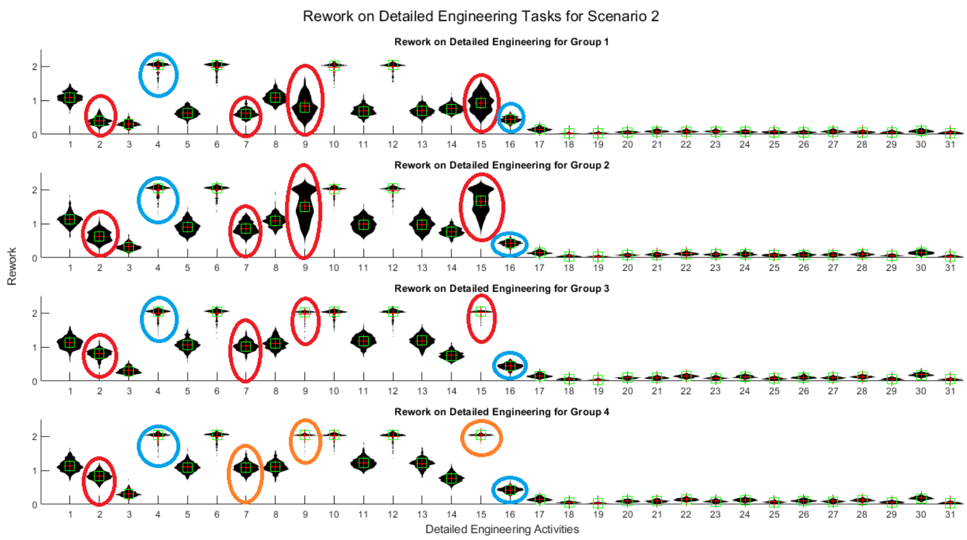


Figure D.6: Rework on detailed engineer tasks for scenario 2: Delayed documentation - Assumptions

Not all activities are affected by this scenario. Almost no rework is reported for activities 17 - 31, which contain construction and piping information for production. Thus, *assumptions have a limit effect on construction and piping information for production*.

Figure D.7 gives violin plots of the rework distribution on section production tasks. Similar to the situation where activities are forced to wait on delayed deliverables is the impact on section production rework limited. The range of rework is even smaller, 0-0.3 compared to 0-0.4. Rework occurrence differs for sections. However, differences among groups are narrow. Therefore it is stated that output deviations are not caused by rework on section production tasks.

In contrast to the limited effect on section production task, are block production tasks affected more by assumptions. Figure D.8 illustrates block production rework for various groups. ***Impact on Workshop for Block Outfitting** is limited*. The impact on **Block Outfitting: Equipment** is limited. But

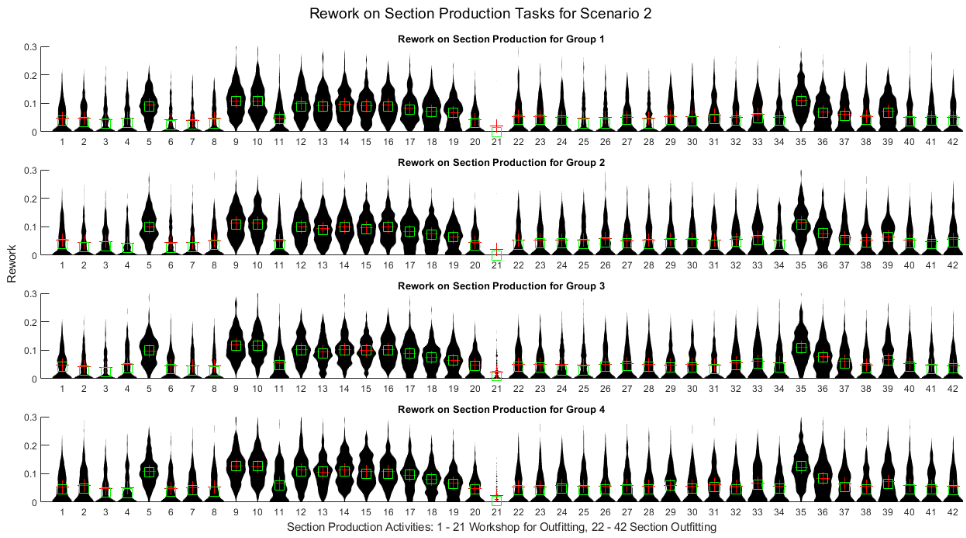


Figure D.7: Rework on section production tasks for scenario 2: Delayed documentation - Assumptions

Block Outfitting: Hot works also suffers from assumptions. For both **Block Outfitting: Equipment** and **Block Outfitting: Hot works** an upward shift is reported increasing with the group numbers. Thus, Workshop Outfitting Equipment and Hot works are sensitive to assumptions and create outcome differences.

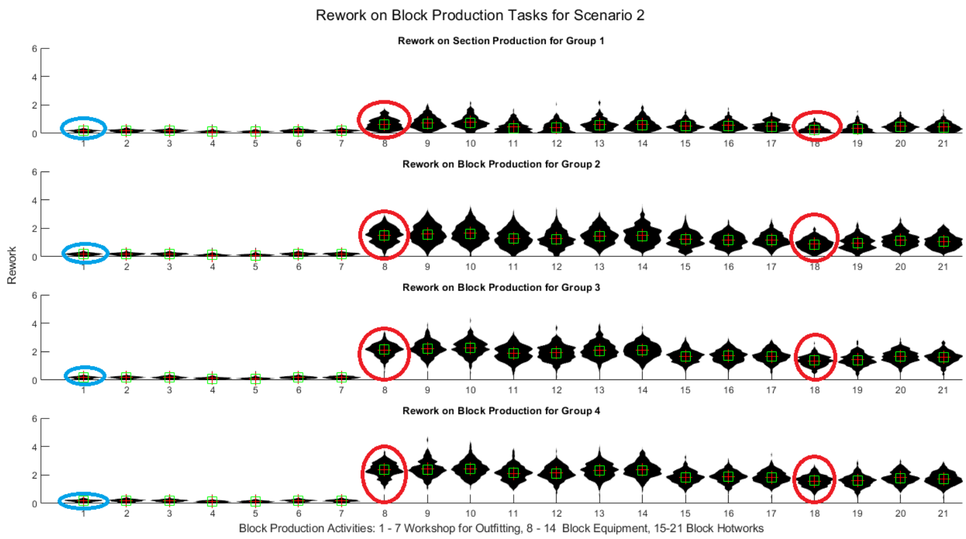


Figure D.8: Rework on block production activities for scenario 2: Delayed documentation - Assumptions

Figure D.9 displays rework distribution of zone outfitting activities. Only the red mark is used, as rework on all activities increases with the groups. Thus *all zone outfitting tasks contribute to the variation in project duration*. As no clear differences among zones are identified, only one zone is highlighted. However, zone outfitting hot works are affected most by the use of assumptions. Also, zone outfitting equipment is affected by the assumptions, although the influence is a little bit less than for the hot works. Least effect in zone outfitting tasks is seen for finalising zone outfitting. This result makes sense as zone outfitting hot works and equipment are linked to both engineering tasks and block production tasks. Thus these tasks have more dependencies, and the likelihood for an adjustment once a change happens is larger.

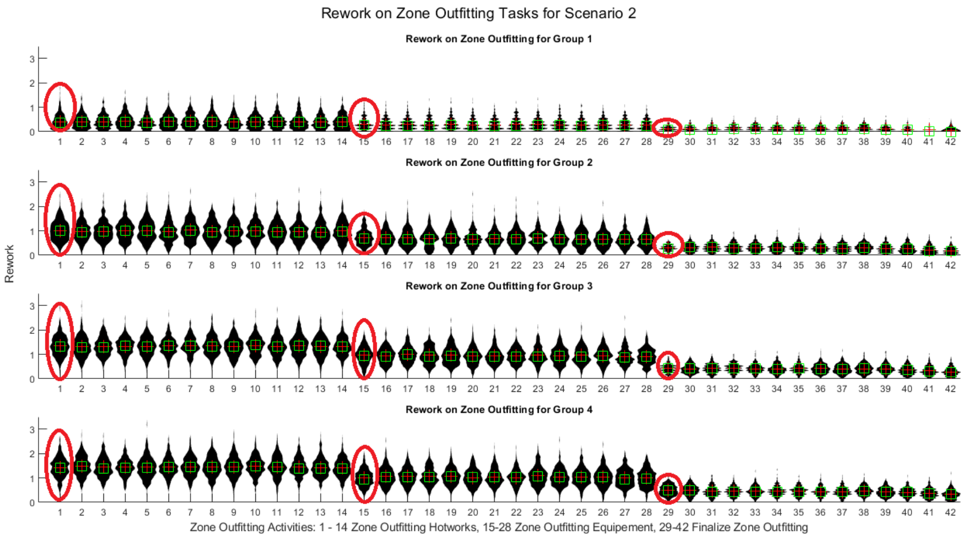


Figure D.9: Rework on zone outfitting tasks for scenario 2: Delayed documentation - Assumptions

D.3. Scenario 3 : Poor communication

Poor communication ranked third on standard deviation in Table 4.7. Since the standard deviation is a measure for the spread of a distribution, comes this ranking with a considerable stretch. The impact of poor communication on the rework of Detailed Engineering, Section Production, Block Production and Zone Outfitting is studied. The aim is to identify sensitive activities for poor communication.

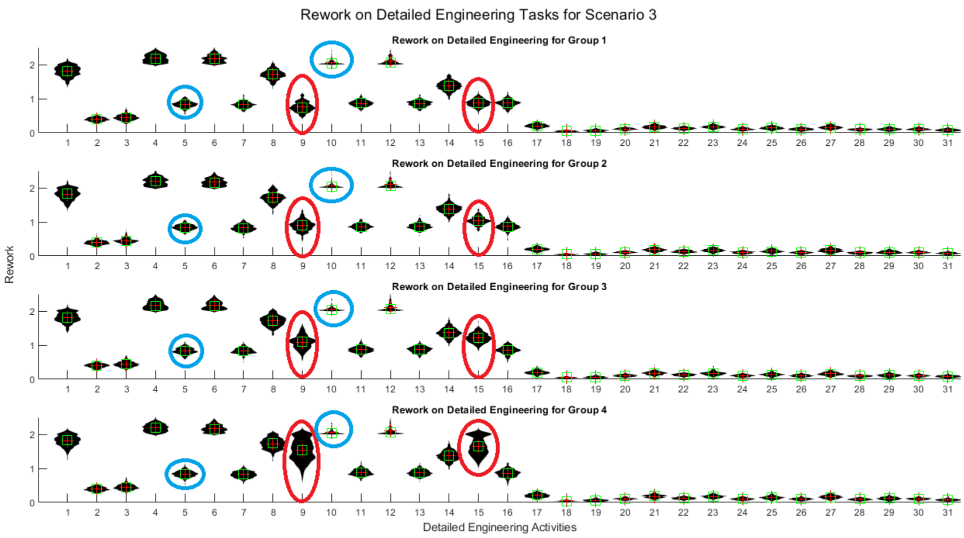


Figure D.10: Rework on detailed engineering tasks for scenario 3: Poor communication

Figure D.10 gives violin plots of the rework distribution for *Detailed Engineering* activities under the poor communication scenarios as defined in 4.4.4. First of all, not all activities are affected by poor communication. Especially the effect on **Construction** and **Piping** Engineering drawings for production is limited. On the other hand, some activities are affected by poor communication but not by the different groups. The blue marked violin plots give examples in Figure D.10. Red marked violin plots indicate activities that are sensitive to poor communication. These activities contribute to the deviation in process outcome, as reported in Figure 4.15. The sensitive activities include:

- **Construction Drawings**
- **Mechanical Diagrams**

From the rework violin plots on *Section Production* activities in Figure D.11 is concluded that *poor communication affects section production little*. The main argument for this statement is the displayed rework range, 0-0.2. This range is even lower than the ranges seen for waiting on delayed deliverables and using assumptions to start activities early. Due to this low range, no activities are marked in Figure D.11. For some activities, rework is reported. However, rework on those activities does not change for the different groups. This behaviour is another argument to state that section production activities do not contribute to the deviation in process outcomes as reported by Figure 4.15.

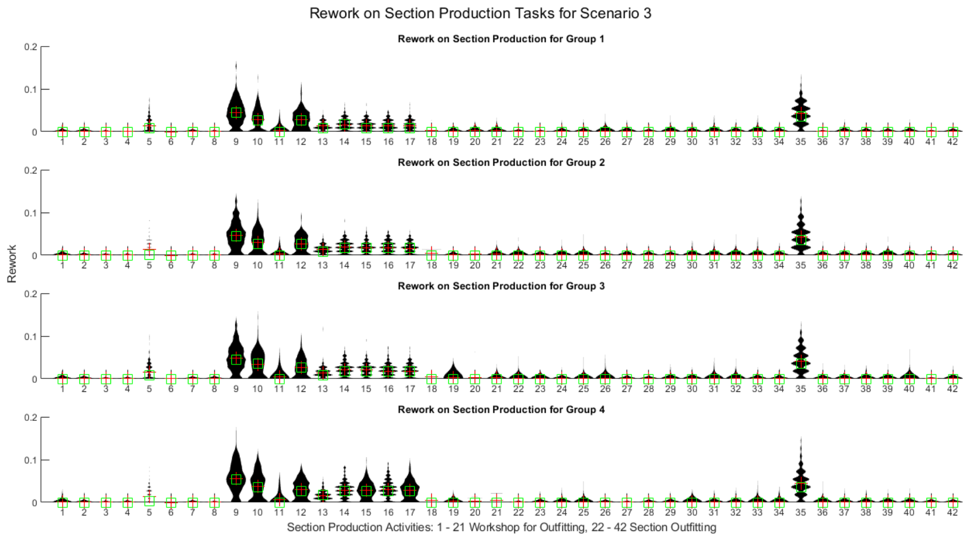


Figure D.11: Rework on section production tasks for scenario 3: Poor communication

Figure D.12 gives violin plots for the rework distribution on *Block Production* tasks for a situation with poor communication. No rework is reported for Workshop for Block Outfitting. The links between detailed engineering activities with no or little rework explain this outcome. On the contrary, most rework is found for Block Outfitting Equipment. Also, substantial rework is reported for Block Outfitting Hot works. This outcome is in line with the dependencies among detailed engineering and block outfitting tasks. Block Outfitting Equipment, and Block Outfitting Hot works depend on Construction Drawings and Mechanical Diagrams, detailed engineering deliverables that are sensitive to poor communication. Both Block Outfitting Equipment and Block Outfitting Hot works are marked red since these activities are sensitive to poor communication. No clear difference is identified among blocks. Thus only one block is marked. To conclude, *rework due to the poor communication of deliverables for Block Outfitting Equipment and Hot works contribute to outcome differences* as reported by Figure 4.15.

In Figure D.13 the violin plots for rework distribution for *Zone Outfitting* tasks in a situation with poor communication are presented. The range of rework is reduced compared to rework on block production activities. Three activities in Figure D.13 are marked red. Other mark colours are not used, since all zone outfitting tasks are affected by poor communication and show an increase in the different groups. As almost no difference is seen for the different zones, only one zone is highlighted as an example.

Figure D.13 reports most rework for Zone Outfitting Hot works. This outcome is in line with obtained insights from interviews with site managers. Site managers name that interruptions in the process affect activities with hot works most. Thus, *all zone outfitting activities named in Figure D.13 contribute to the deviation in project duration reported in Figure 4.15*.

To summarize, ***care is required to communicate the following deliverables:***

- Construction Drawings
- Mechanical Diagrams

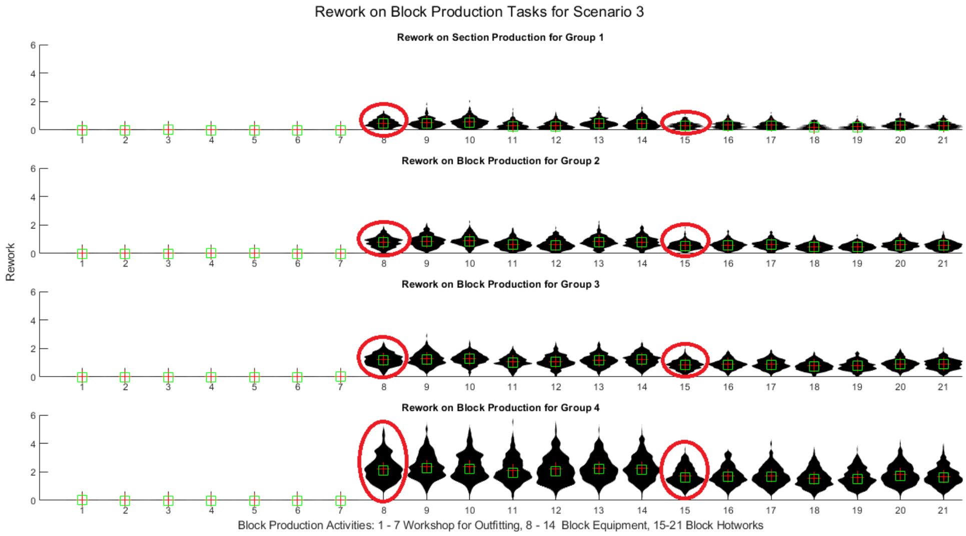


Figure D.12: Rework on block production tasks for scenario 3: Poor communication

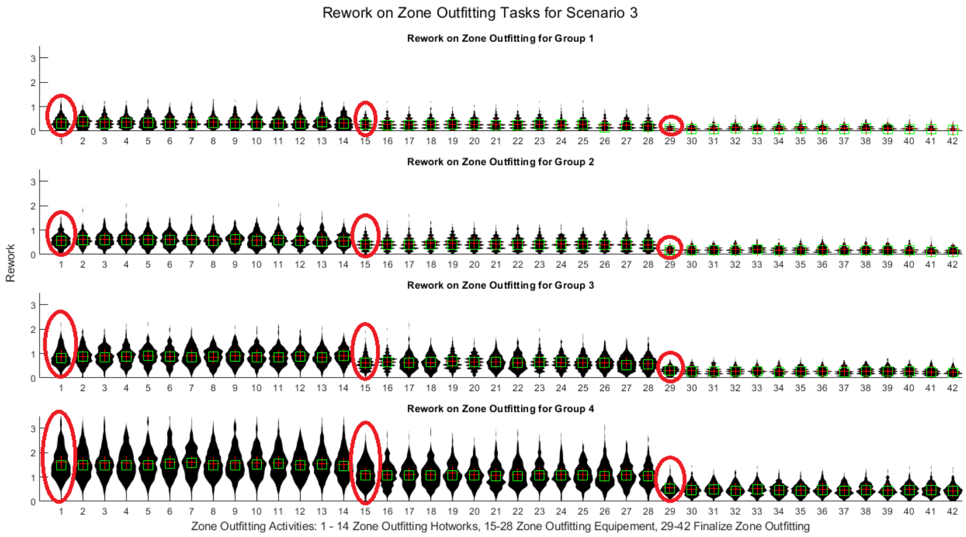


Figure D.13: Rework on zone outfitting tasks for scenario 3: Poor communication

- Block Outfitting Equipment
- Block Outfitting Hot works
- Zone Outfit Hot works
- Zone outfit Equipment
- Zone Outfitting Finalize space

Especially, mechanical diagrams and construction drawings require attention, due to the relative massive impact (10%) on block outfitting. Also, zone outfitting Hot works have a feedback loop with a relatively large effect (6%) to block equipment.

D.4. Scenario 4: Input changes

Input changes ranked second for mean value and schedule risk in Table 4.7. Also, Figure 4.20 reports large outliers for this scenario compared to the other scenarios. Violin plots of rework on project activities are used to understand which activities are affected most by input changes and harm project execution. For this approach is stated that *activities are sensitive for input changes if rework on project activities changes for the groups*. However, it should be noted that results are affected by the defined input changes in 4.4.5.

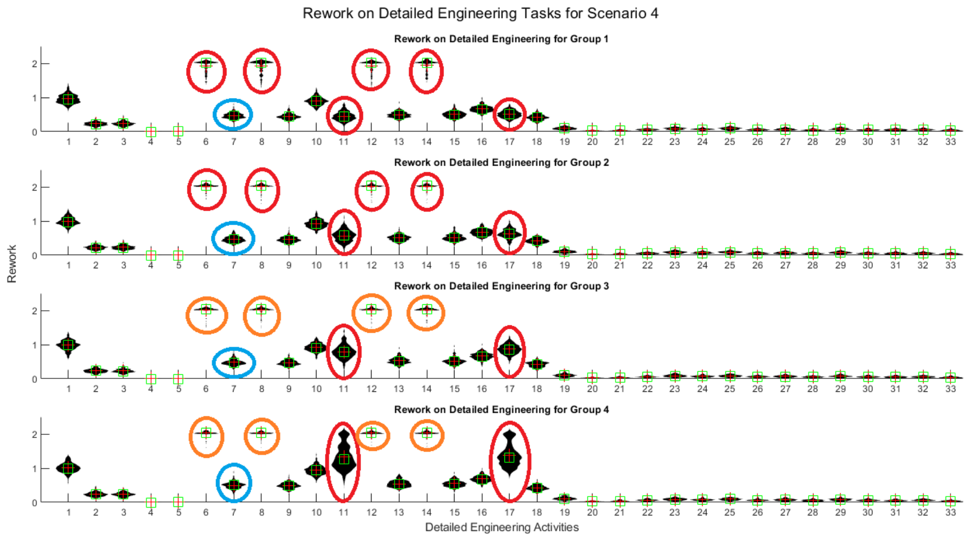


Figure D.14: Rework on detailed engineering tasks for scenario 4: Input changes

Figure D.14 provides violin plots of rework on *Detailed Engineering* tasks. Red marked represent activities that influence project execution and contribute to outcome differences. Blue marked represent activities that have rework but do not contribute to project outcome differences. Orange marked the point at a stop of the influence of specific activity on project outcome differences.

Two activities are red marked for all groups; activity 11 and 17. These activities include:

- **Superstructure outfitting drawings**
- **Mechanical arrangement drawings**

Four activities are red and orange marked. These activities influence outcome differences, but their influence has a limit. The limit means that rework on other activities causes an outcome in group 3 or 4. This concerns:

- **Construction drawings**
- **Mechanical diagrams**

Figure D.15 shows violin plots for rework distribution on section production tasks as a consequence of input changes. Values and differences among section production tasks are small. The differences are mainly seen in the presence of outliers. Again rework values are small, thus is stated that *rework on section production activities due to input changes is not the primary driver of project duration deviations* as reported in Figure 4.16.

Figure D.16 provides rework distribution plots of block production tasks due to input changes. Again, most rework is reported for block production activities. For all tasks rework is reported, but ***Block Outfitting Equipment** and **Block Outfitting Hot Works** are affected most by input changes*. Differences among the displayed blocks are small; thus, only one block is marked as an example. Only one task is blue marked, workshop for block outfitting. This blue mark states that rework is reported but that this rework does not contribute to the outcome differences, as shown in Figure 4.16. Block outfitting equipment, and hot works are marked red. These activities contribute to the outcome differences in Figure

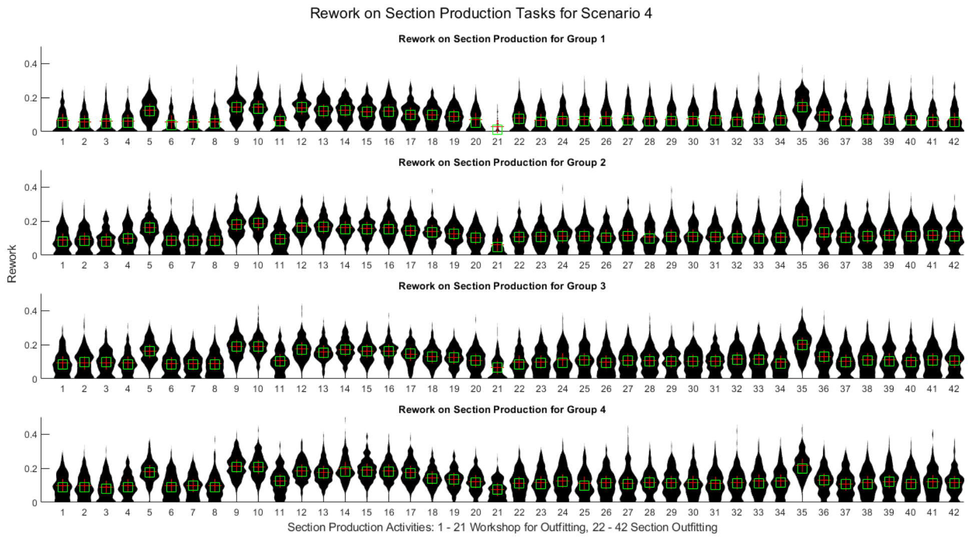


Figure D.15: Rework on section production tasks for scenario 4: Input changes

4.16. The other required documentation explains the difference in rework between workshop for block outfitting and block outfitting activities for the tasks. There is more rework on outfitting documentation than is present on other documentation. That results in more rework on outfitting tasks. Also are block outfitting tasks more affected by rework for subsequent jobs than workshop for block outfitting.

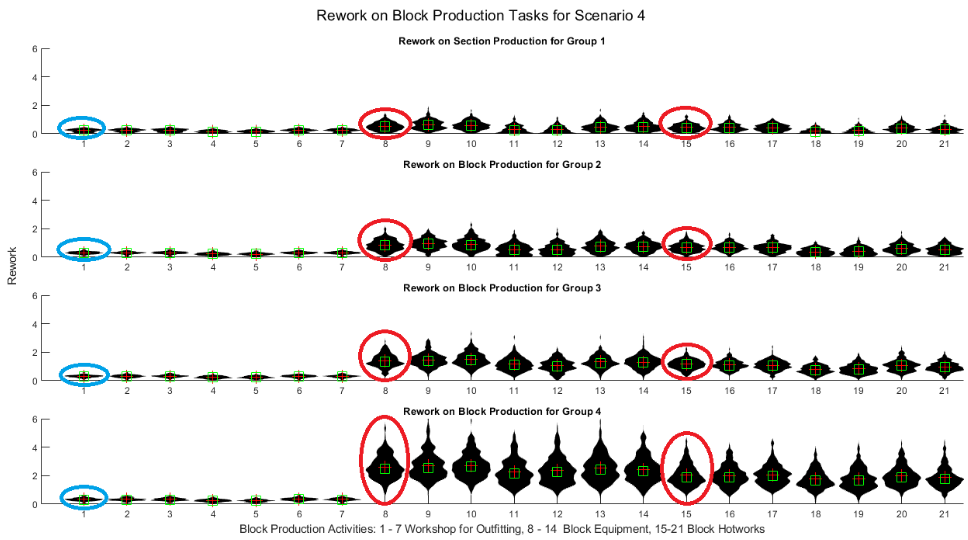


Figure D.16: Rework on block production tasks for scenario 4: Input changes

Figure D.17 illustrates rework distribution violin plots for zone outfitting tasks as a consequence of input changes. Again large values are reported. Since all activities have increasing rework with the groups, only the red mark is used. Also, differences among zones are small; thus, only one zone is highlighted.

The most impact is seen for **Zone Outfitting Hot works**. This rank is inline with gained insights from interviews with site managers. Interviewed site managers state that input changes have a severe impact on hot work activities and related tasks. Rework on **Zone Outfitting Equipment** also has a considerable contribution to outcome differences. The location of zone outfitting in the entire project explains the enormous impact on project outcome. Zone outfitting is one of the last stages in the

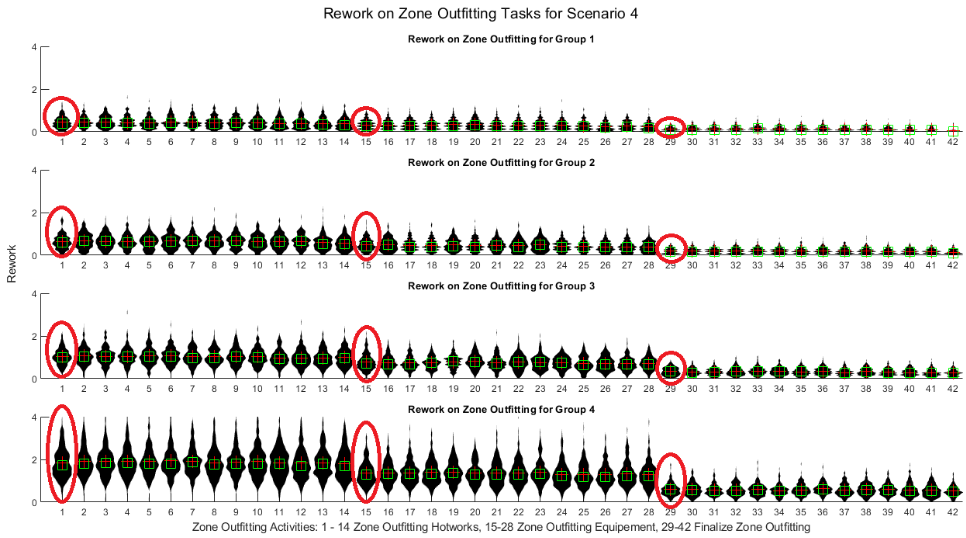


Figure D.17: Rework on zone outfitting tasks for scenario 4: Input changes

production process; thus, input changes affect a large number of tasks.

*Input changes have a severe impact on project execution, especially when it concerns **Block Outfitting Equipment**, **Block Outfitting Hot works**, **Zone Outfitting Hot works** and **Zone Outfitting Equipment***.

D.5. Scenario 5: Mistakes

Mistakes ranked highest in section 4.6. However, a spread in data is reported in Figure 4.17. Violin plots are used to study the impact of mistakes on project execution in more detail. The aim is to identify activities that are sensitive for mistakes. For this approach is stated that *activities are sensitive for input changes if rework on project activities changes for the groups*.

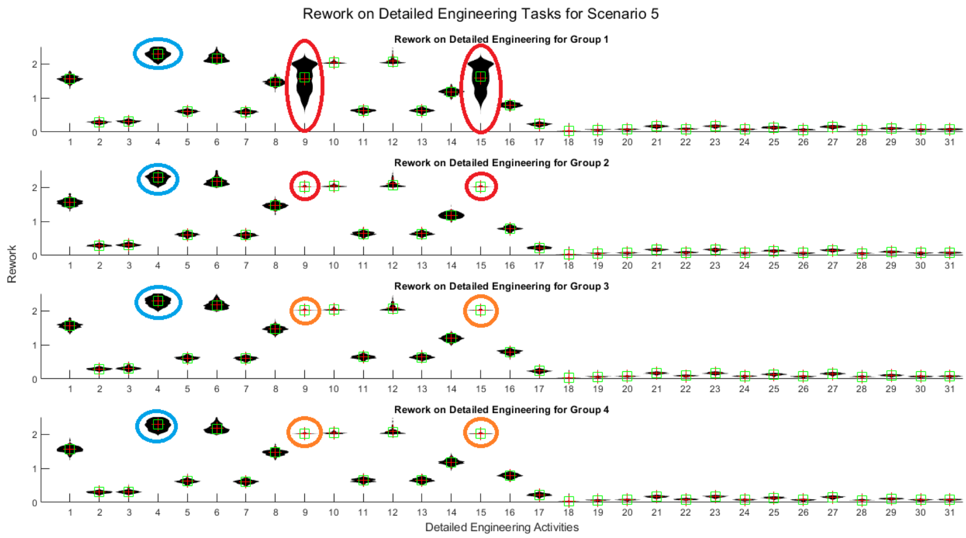


Figure D.18: [Rework on detailed engineering tasks for scenario 5: Mistakes

Figure D.18 provides rework distribution violin plots for a situation with mistakes as defined in subsection 4.4.6. Impact on activities differs. For some activities rework distribution does not change for the groups. An example of such an activity is given by the blue marked task. These activities are not seen as sensitive for mistakes as they do not contribute to a difference in project duration. For other activities rework distribution changes with the groups. These activities are named as sensitive activities to mistakes and marked red. However there impact on project distribution is limited.

At a given moment differences between groups are no longer reported, these points are orange marked. Activity 9, Construction Drawings , forms as an example. Rework distributions between in group 1 and group 2 differs significantly. However, no difference is seen between group 2 and 3. Thus rework on activity 9 can affect project outcome, it determines whether it is or is not possible to find a project duration in group 1. However, rework on activity 9 does not tell whether the project duration ends in group 3 or 4. Other activities (other then Detailed Engineering) determine whether the project duration results in a duration from group 3 or 4.

Detailed Engineering activities that have an influence on project duration are:

- **Construction Drawings**
- **Mechanical Diagrams**

Figure D.19 gives the influence of mistake related rework on section production activities. A difference is noticed among sections. Some sections suffer more from mistake related rework than others. Differences among groups are small. But differences are identified for tasks 13 and 16 (both Workshop for Section Outfitting), which are red marked. It can be seen that for both red marked activities the red cross, representing the median value, is increasing with the groups. However, it should be mentioned that rework range is small, 0-0.2. Further are the differences among the groups also small. Thus, it is more likely that duration output differences are caused by rework on other activities (other than section production). On the other hand there are also activities that report rework but do not change for the groups. The blue marked violin plot in Figure D.19 is an example of such an activity. It is understood that more activities with similar behaviour are present in this figure, only one task is highlighted to give an example. The blue marked activity has rework but does not contribute to outcome differences.

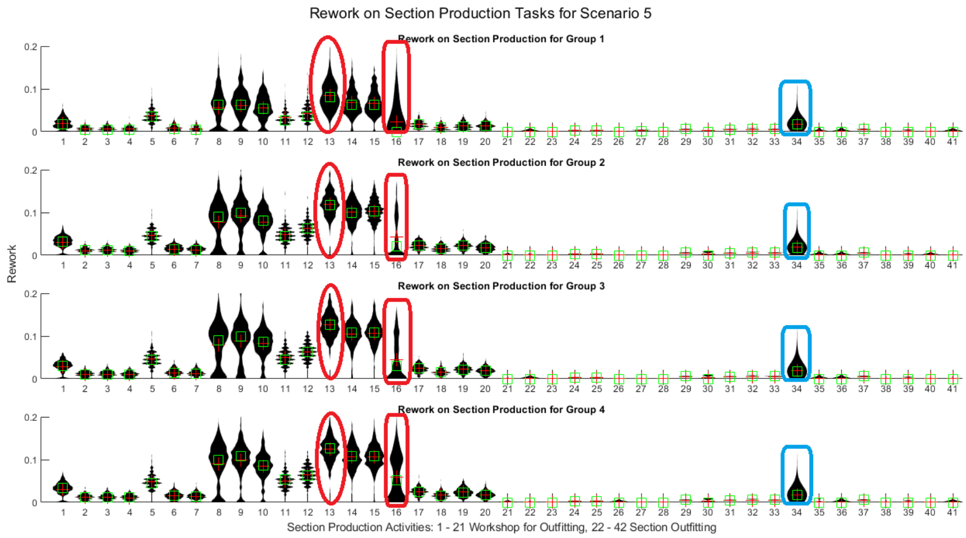


Figure D.19: Rework on section production tasks for scenario 5: Mistakes

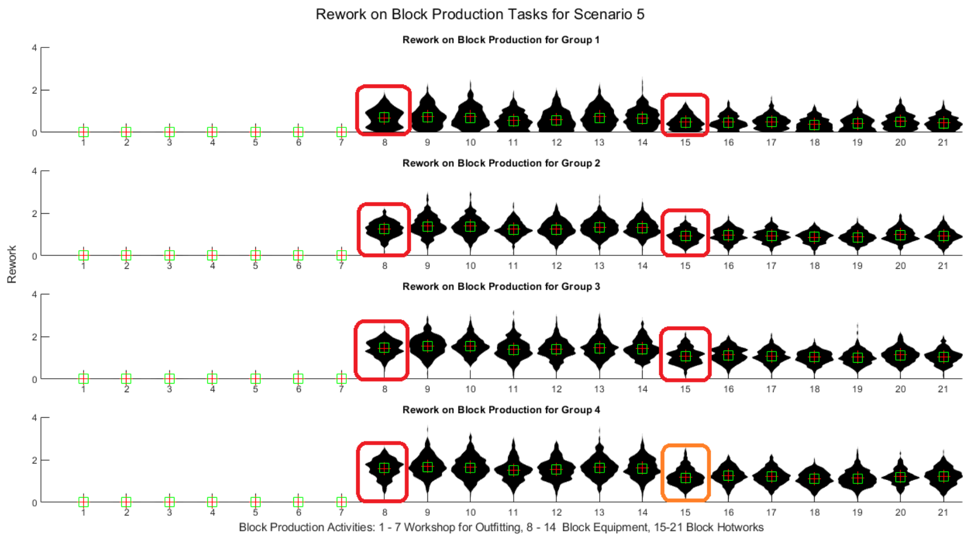


Figure D.20: Rework on block production tasks for scenario 5: Mistakes

Figure D.20 gives violin plots of mistake related rework on block production tasks. The rework distribution looks different than for other the other scenarios as very little rework is reported for the activity Workshop for Block Outfitting. It looks like Figure D.20 does not report any rework at all for these activities. However, rework values are so small given the used scale that they are not visible on the violin plot. Apparently are hot works and equipment outfitting much more affected by mistakes. Also a difference in rework impact can explain the difference.

Rework for Block outfitting equipment and hot works increases with the groups. The increase can be seen for the red marked violin plots in Figure D.20. Since no clear difference is noted among the different blocks, only one block is highlighted. From this increase is concluded that mistake-related rework on these tasks influences severity of overruns. This conclusion applies for all blocks as no clear difference is identified.

Figure D.21 reports mistake-related rework on zone outfitting activities. Again, for most activities an increase is seen. However, for some activities the increase is limited. Red marked activities represent activities with an increase among the groups. These activities have an impact on project overruns. For

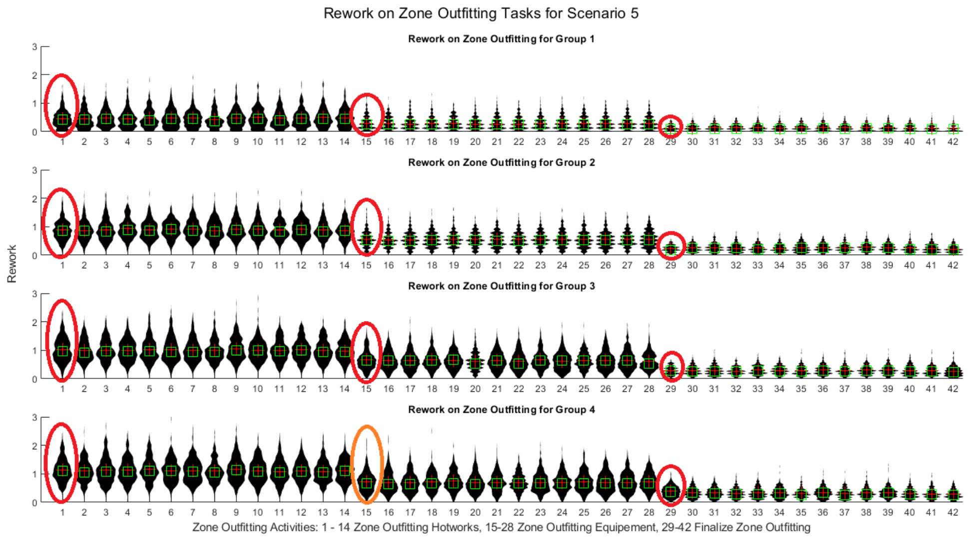


Figure D.21: Rework on zone outfitting tasks for scenario 5: Mistakes

activity 15, zone outfitting equipment, the impact is limited to the third group. The violin plot in group 4 is marked orange to illustrate this limited. No clear differences are identified among zones. For example, the increase for all zone outfitting hot works activities, activities 1-14, is nearly similar. Therefore only one zone is marked.

To summarize, *Construction Drawings, Mechanical Diagrams, and Block and Zone outfitting suffer most from mistakes*. This outcome is reasonable, since outfitting tasks have large number of dependencies and are time-consuming. Outfitting activities are located at the end of the process, which gives less freedom to find creative solutions once a mistake is identified then early in the process. This outcome stresses the importance of correct project documentation used for the outfitting phase.

E

Time Plan

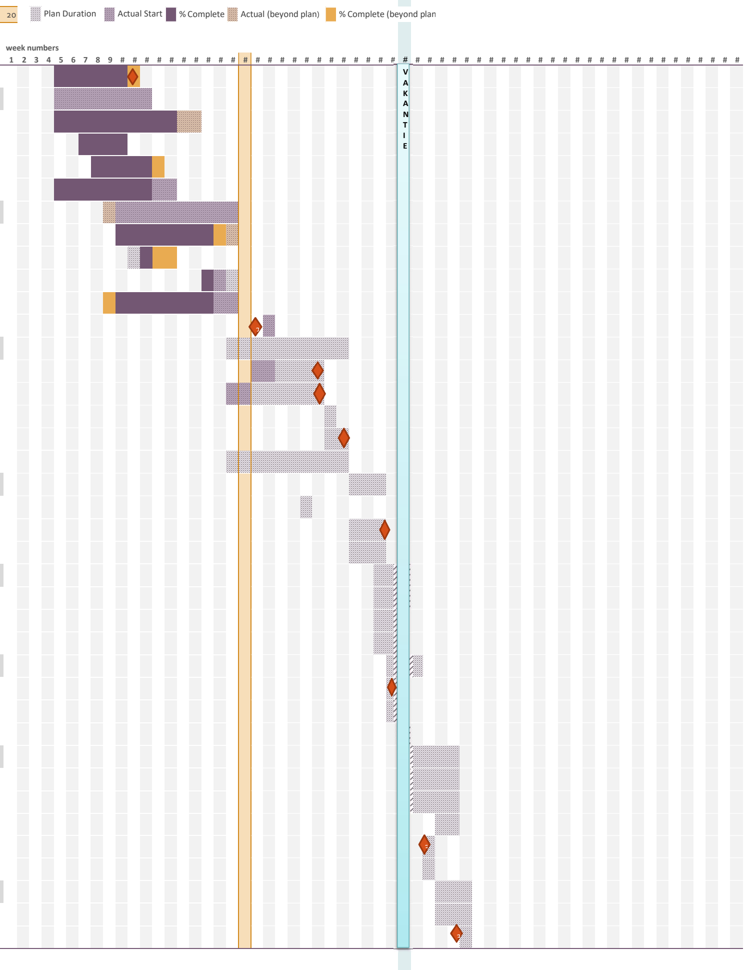
Master Thesis Planner

Select a period to highlight at right. A legend describing the charting follows.

Anne Reumer

Period Highlight: 20

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Plan van Aanpak	5	6	5	7	100%
Preperations	5	8	5	8	
Collect information about DTC	5	10	5	12	90%
Search for literature on success in shipbuiliding	7	4	7	4	100%
Other Industries	8	5	8	6	100%
Report Preperations	5	10	5	10	80%
Analysis	10	10	9	11	
Literature Study	10	8	10	10	90%
Define Research Methodology	11	2	12	3	100%
Test Data Collection Method	17	3	17	2	80%
Report Analysis	10	10	9	11	90%
MID TERM	22	0,5	22	0,5	
Solution Design	19	10			0%
Develop Document Flow Simulation	21	6	22		
Data Collection Process	19	8	19	2	
Determine Impact Priority Quality Attributes	27	1			
Create procedure to develop future sharing strategy	27	2			
Report Solution Design	19	10			
Find Answers	29	3			0%
Select DTC project	25	0,5			
DTC case study	29	3			
Report Find Answers	29	3			
Report	31	3			0%
Paper	31	3			0%
Report Conclusions	31	2			0%
Report Recommandations	31	2			0%
Concept 1 meeting	32	3			0%
Hand in concept 1 (hard copy)	32	1			0%
draft paper 6-10 pages	32	1			0%
meeting concept 1	33	0,5			0%
Concept 2 (Green light meeting)	34	4			0%
Concept 2	34	4			0%
paper	34	4			0%
presentation	36	2			0%
DEADLINE paper	35	0,5			0%
hand in concept 2	35	0,5			0%
Final Thesis and Presenatation	36	3			0%
Revisions from concept 2	36	3			0%
DEADLINE Final Thesis 3 weeks before graduation	38	0,5			0%
	16	28	16	30	0%



Bibliography

M. Adler. An algebra for data flow diagram process decomposition. *IEEE Transactions on Software Engineering*, 14(2):169–183, February 1988. ISSN 0098-5589. doi: 10.1109/32.4636.

P. S. Adler, A. Mandelbaum, V. Nguyen, and E. Schwerer. From project to process management: An empirically-based framework for analyzing product development time. *Management Science*, 41(3): 377–559, March 1995. doi: https://doi.org/10.1287/mnsc.41.3.458.

B. Ahn. Managing the efficiency of foreign engineering contracts: A study of a norwegian and south korean project interface. mathesis, University of Stavanger, June 2015. URL http://hdl.handle.net/11250/297581.

E Akampurira and A. Windapo. Factors influencing the quality of design documentation on south african civil engineering projects. *Journal of the South African Institution of Civil Engineering*, 60(3):41–48, September 2018. ISSN 2309-8775. doi: http://dx.doi.org/10.17159/2309-8775/2018/v60n3a4.

E. Akampurira and A. Windapo. Key quality attributes of design documentation: South african perspective. *Journal of Engineering, Design and Technology*, 17(2):362–382, April 2019. ISSN 1726-0531. doi: https://doi.org/10.1108/JEDT-08-2018-0137. URL https://www.emerald.com/insight/content/doi/10.1108/JEDT-08-2018-0137/full/html.

E.C. Akcay, S. Ergan, and D. Arditi. Modeling information flow in the supply chain of structural steel components. *Journal of Civil Engineering and Management*, 23(6):753–764, 2017. ISSN 1822-3605. doi: https://doi.org/3846/13923730.2017.1281841.

H. Akkermans and J. Bosker. Design guidelines for participative business modelling projects: Lessons from an unsuccessful case study. In *Problem - Solving Methodologies*, pages 15–24. International System Dynamics Conference, 1994.

S.W. Ambler. *The object primer : agile model-driven development with Uml 2.0*. London : Cambridge University Press, 2004, 3rd edition, 2004. ISBN 0521540186 9780521540186. Studiecollectie-2019-2020.

F Andritos and J. Perez-Prat. The automation and integration of production processes in shipbuilding. resreport, European Commision Joint Research Centre, DG Enterprise, unit E.6, June 2000.

F.M. Arain and S.A. Assaf. Consultant's prospects of the sources of design and construction interface problems in large building projects in saudi arabia. *Journal of King Abdulaziz University-Environmental Design Sciences*, 5(2):15–37, January 2007. doi: 0.4197/env.5-1.2.

J. Arcade, M. Godet, F. Meunier, and F. Roubelat. Structural analysis with the micmac method and actor's strategy with mactor method. *Futures Research Methodology, American Council for the United Nations University: The Millennium Project*, 2010, 1999.

D. Arditi, S. Nayak, and A. Damci. Effect of organizational culture on delay in construction. *International Journal of Project Management*, 35(2):136–147, February 2017. ISSN 0263-7863. doi: https://doi.org/10.1016/j.ijproman.2016.10.018.

B. E. Asbjornslett. Project supply chain management: From agile to lean. phdthesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, November 2002. URL https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/240611.

S. Assaf, M.A. Hassanain, and A. Abdallah. Review and assessment of the causes of deficiencies in design documents for large construction projects. *International Journal of Building Pathology and Adaptation*, 36(3):300–317, July 2018. ISSN 2398-4708. doi: 10.1108/IJBPA-10-2017-0050.

S. Austin, A. Baldwin, and A. Newton. A data flow model to plan and manage the building design process. *Journal of Engeering Design*, 7(1):3–25, 1996. ISSN 0954-4828. doi: 10.1080/09544829608907924.

A. N. Baldwin, S. A. Austin, T. M. Hassan, and A. Thorpe. Modelling information flow during the conceptual and schematic stages of building design. *Construction Management and Economics*, 17(2): 155–167, March 1999. ISSN 0144-6193. doi: https://doi.org/10.1080/014461999371655.

P. L. Bannerman. Defining project success: A multilevel framework. *Project Management Institute*, 2008.

H.A. Bashir, K. AlZebdeh, and J. Abdo. An eigenvalue based approach for assessing the decomposability of interdependent design project tasks. *Concurrent Engineering*, 17(1):35–42, March 2009. doi: https://doi.org/10.1177/1063293X09102248.

R.M. Bastos and D.D.A. Ruiz. Towards an approach to model business processes using workflow modeling techniques in production systems. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, Maui, HI, USA, USA, January 2001. IEEE. doi: 10.1109/HICSS.2001.927231. 0-7695-0981-9.

R.M. Bastos and D.D.A. Ruiz. Extending uml activity diagram for workflow modeling in production systems. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, volume 291, Big Island, HI, USA, January 2002. IEEE. doi: 10.1109/HICSS.2002.994510.

M. Beckmann, V.N. Michalke, A. Schlutter, and A. Vogelsang. Removal of redundant elements within uml activity diagrams. In *2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, Austin, TX, USA, November 2017. IEEE. doi: 10.1109/MODELS.2017.7. 978-1-5386-3492-9.

W. Belassi and O.I. Tukel. A new framework for determining critical success/failure factors in projects. *International Journal of Project Management*, 14(3):141–151, June 1996. ISSN 0263-7863. doi: https://doi.org/10.1016/0263-7863(95)00064-X.

L. Benthall, T. Briggs, B. Downie, B. Gischner, B. Kassel, and R. Wood. Step for shipbuilding: A solution for product model data exchange. *Journal of Ship Production*, 19(1):44–52, February 2003. ISSN 8756-1417.

O. Berard. Building information modeling for managing design and construction. phdthesis, Technical University of Denmark, Department of Civil Engineering, July 2012. 1601-2917.

R.W. Bolton. Enabling shipbuilding supply chain virtual enterprises. *Journal of Ship Production*, 17(2): 76–86, May 2001. ISSN 8756-1417.

T. L. Briggs, S. J. Baum, and T. M. Thomas. Interoperability framework. *Journal of Ship Production*, 21 (2):99–107, May 2005. ISSN 8756-1417. URL https://www.onepetro.org/journal-paper/SNAME-JSP-2005-21-2-99.

R.J. Brooks and A.M. Tobia. Choosing the best model: Level of detail, complexity, and model performance. *Mathematical and Computer Modelling*, 24(4):1–14, August 1996. ISSN 0895-7177. doi: https://doi.org/10.1016/0895-7177(96)00103-3.

Browning, editor. *Project Risk Management Using the DSM*, Cambridge, MA, 2000.

Browning. Process integration using the design structure matrix. *Systems Engineering*, 5(3):180–193, June 2002. ISSN 1098-1241. doi: https://doi-org.tudelft.idm.oclc.org/10.1002/sys.10023.

Browning. Systems architecting and program management with the design structure matrix (dsm). INCOSE International Symposium, November 2014a. Tutorial.

Browning. Design structure matrix extensions and innovations: A survey and new opportunities. *IEEE Transactions on Engineering Management*, 63(1):27–52, February 2016. ISSN 1558-0040. doi: 10.1109/TEM.2015.2491283.

T. R. Browning. *Modeling and Analyzing Cost, Schedule, and Performancein Complex System Product Development*. phdthesis, Massachusetts Institute of Technology, Cambridge, MA, USA, December 1998.

T. R. Browning. A role-playing game for teaching about enterprise process integration. *Journal of Enterprise Transformation*, 4(3):226–250, August 2014b. ISSN 1948-8289. doi: https://doi.org/10.1080/19488289.2013.879682.

T.R. Browning and S.D. Eppinger. Modeling impacts of process architecture on cost and schedule risk in product development. *Transactions on Engineering Management*, 49(4):428–442, November 2002. ISSN 1558-0040. doi: 10.1109/TEM.2002.806709. URL https://ieeexplore.ieee.org/abstract/document/1176870.

T.R. Browning and S.D. Eppinger. A simulation-based process model for managing complex design projects. *TRANSACTIONS ON ENGINEERING MANAGEMENT*, 52(3):316–328, August 2005. ISSN 1558-0040. doi: 10.1109/TEM.2005.850722. URL https://ieeexplore.ieee.org/abstract/document/1468402.

Tyson R. Browning. Applying the design structure matrix to system decomposition and integration problems: A review and new directions. *Transactions on Engineering Management*, 48(3):292–306, August 2001. ISSN 1558-0040. doi: 10.1109/17.946528.

Tyson R Browning. On customer value and improvement in product development processes. *Systems engineering*, 6(1):49–61, January 2003. doi: https://doi.org/10.1002/sys.10034.

Tyson R Browning and Ralph D Heath. Reconceptualizing the effects of lean on production costs with evidence from the f-22 program. *Journal of operations management*, 27(1):23–44, January 2009.

D. Bryde, M. Broquates, and J.M. Volm. The project benefits of building information modelling (bim). *International Journal of Project Management*, 31(7):971–980, October 2013. ISSN 0263-7863. doi: https://doi.org/10.1016/j.ijproman.2012.12.001.

M.K. Buckland. What is a " document"? *Journal of the American Society for Information Science*, 48 (9):804–809, 1997.

M. Carrascosa, S. D. Eppinger, and D. E. Whitney. Using the design structure matrix to estimate product development time. In *Proceedings of DETC'98*, pages 1–10, Atlanta, Georgia, USA, September 1998. 1998 ASME Design Engineering Technical Conferences. Markov Chain.

C. Chang and T.L. Jimming. Data flow model of a total service quality management system. *Computers and Industrial Engineering*, 21(1-4):117–121, 1991. ISSN 0360-8352. doi: https://doi.org/10.1016/0360-8352(91)90074-G.

Chen and Huang. A systematic approach for supply chain improvement using design structure matrix. *Journal of Intelligent Manufacturing*, 18(2):285–299, April 2007. ISSN 0956-5515. doi: https://doi.org/10.1007/s10845-007-0022-z.

W. Chen, A.R. Hamzah, A. Faizul, Z. Nurshuhada, and W.C. Wei. Improve documentation quality for engineering: leap forward through project learining. *Journal of Frontiers in Construction Engineering*, 2:83–88, 2013.

Y. Cheng and H. Shaw. Cloud-based, service-oriented and knowledge-sharing architecture: its design and application in shipbuilding. *International Journal of Computer Integrated Manufacturing*, 28(2): 137–154, 2012. ISSN 0951-192X. doi: 10.1080/0951192X.2013.874587. Original Articles.

Cho. An integrated method for managing complex engineering projects using the design structure matrix and advanced simulation. mathesis, MASSACHUSETTS INSTITUTUE OF TECHNOLOGY, Department of Mechanical Engineering, May 2001. URL https://dspace.mit.edu/handle/1721.1/8607.

Cho and Eppinger. A simulation-based process model for managing complex design projects. *TRANSACTIONS ON ENGINEERING MANAGEMENT*, 52(3):316–328, August 2005. ISSN 1558-0040. doi: 10.1109/TEM.2005.850722.

S. Cho and S.D. Eppinger, editors. *Product Development Process Modeling Using Advanced Simulation*, September 2001. ASME 2001 Design Engineering Technical Conference and Computers and Information in Engineering Conference, DETC'01. URL http://hdl.handle.net/1721.1/3807. DETC2001/DTM-21691.

I. C Ciobanu and G. P. Neupane. Phased-based project management at aker yards langsten - a lean shipbuilding perspective. mathesis, Molde University College, Molde, June 2008.

D.L. Clark, Howell. D.M., and C.E. Wilson. Improving naval shipbuilding project efficiency through rework reduction. mathesis, Naval Postgraduate School, Monterey, California, September 2007.

J.M.G. Coenen. *Controlling Engineering-To-Order Processes in Shipbuilding: A Model-Based Approach*. phdthesis, Delft University of Technology, The Netherlands, Delft, June 2008.

T. Cooke-Davies. The "real" success factors on projects. *International Journal of Project Management*, 20(3):185–190, April 2002. ISSN 0263-7863. doi: https://doi.org/10.1016/S0263-7863(01)00067-9.

S.Z. Dogan, D. Arditi, S. Gunhan, and B. Erbasaranoglu. Assessing coordination performance based on centrality in an e-mail communication network. *Journal of Management in Engineering*, 31(3), May 2015. ISSN 0742-597X. doi: 10.1061/(ASCE)ME.1943-5479.0000255.

H. Doloi, A. Sawhney, K.C. Iyer, and S. Rentala. Analysing factors affecting delays in indian construction projects. *International Journal of Project Management*, 30(4):479–489, May 2012. ISSN 0263-7863. doi: https://doi.org/10.1016/j.ijproman.2011.10.004.

Jonas Dorn. (distributionplot.m). In *Violin Plots for plotting multiple distributions*. MATLAB Central File Exchange, February 2017. URL https://nl.mathworks.com/matlabcentral/fileexchange/23661-violin-plots-for-plotting-multiple-distributions-distributionplot-m. MATLAB function.

DTC. Damen technical cooperation : Think global act local. DamenSharePoint, January 2020a. URL https://damen.sharepoint.com/sites/bf_communication/Shared%20Documents/Forms/AllItems.aspx?web=1&FilterField1=InternalCommunication&FilterValue1=Brochures&FilterType1=Choice&id=%2Fsites%2Fbf%5Fcommunication%2FShared%20Documents%2FBRO%2D1%2D006%2D2001%20PMC%20DTC%20brochure%2001%2D2020%20%2Epdf&parent=%2Fsites%2Fbf%5Fcommunication%2FShared%20Documents.

DTC. Dtc value proposition blocks. DTCSharePoint, January 2020b. PowerPointPresentation.

M. Dumas and A.H.M. ter Hofstede. Uml activity diagrams as a workflow specification language. In M. Gogolla and C. Kobryn, editors, *The Unified Modeling Language. Modeling Languages, Concepts, and Tools*, volume 2185 of *Lecture Notes in Computer Science*, pages 76–90. International Conference on the Unified Modeling Language, Springer, Berlin, Heidelberg, September 2001. doi: https://doi.org/10.1007/3-540-45441-1_7. URL https://link.springer.com/chapter/10.1007/3-540-45441-1_7#citeas. 978-3-540-45441-0.

C. Durugbo, J. A. Erkoyuncu, A. Tiwari, J. R. Alcock, R. Roy, and E. Shehab. Data uncertainty assessment and information flowanalysis for product-service systems in a librarycase study. *International Journal of Services Operations and Informatics*, 5(4):330–350, December 2010. doi: http://dx.doi.org/10.1504/IJSOI.2010.037002.

E. Eloranta, A. Hameri, and M. Lahti. Improved project management through improved document management. *Computers in Industry*, 45(3):231–243, July 2001. ISSN 0166-3615. doi: https://doi.org/10.1016/S0166-3615(01)00099-9.

S.D. Eppinger, D.E. Whitney, R.P. Smith, and D.A. Gebala. A model-based method for organizing tasks in product development. *Research in Engineering Design*, 6:1–13, March 1994. doi: https://doi.org/10.1007/BF01588087.

Steven D. Eppinger and Tyson R. Browning. *Design Structure Matrix Methods and Applications*. MIT Press, 2012. ISBN 978-0-262-01752-7. URL https://app.knovel.com/web/toc.v/cid:kpDSMMA003/viewerType:toc/. Ebook.

E. Ergen and B. Akinci. Formalization of the flow of component-related information in precast concrete supply chains. *Journal of Construction Engineering and Management*, 134(2):112–121, February 2008. ISSN 0733-9364. doi: https://doi.org/10.1061/(ASCE)0733-9364(2008)134:2(112).

R. Eshuis and R. Wieringa. Tool support for verifying uml activity diagrams. *IEEE Transactions on Software Engineering*, 30(7):437–447, July 2004. ISSN 1939-3520. doi: 10.1109/TSE.2004.33. URL https://ieeexplore.ieee.org/abstract/document/1318605.

A. Fatawu. Assessing the quality of design and contract documentation and its impact on construction projects performance in northern ghana. mathesis, College of Art and Built Environmen, January 2017. URL http://hdl.handle.net/123456789/9935.

M. Fleischer, R. Kohler, T. Lamb, and H. Bongiorni. Shipbuilding supply chain integration project. resreport, Environmental Research Institute of Michigan, November 1998. Interim Report.

T. Gaertner, S. Terstegen, and C.M. Schlick. Applying dsm methodology to improve the scheduling of calibration tasks in functional integration projects in the automotive industry. *The Journal of Modern Project Management*, 3(2), August 2015. ISSN 2317-3963. URL https://www.journalmodernpm.com/index.php/jmpm/article/view/137.

E. D. Galvez and J.B. Ordieres. Evaluation of project duration uncertainty using the dependency structure matrix and monte carlo simulations. *Journal of Construction*, 14(2):72–79, August 2015.

C. Gane and T. Sarson. *Structured systems analysis : tools and techniques*. Englewood Cliffs N.J. : Prentice-Hall, 1979. ISBN 0138545472 9780138545475.

R. Gelbard, N. Pliskin, and I. Spiegler. Integrating system analysis and project management tools. *International Journal of Project Management*, 20(6):461–468, August 2002. doi: https://doi.org/10.1016/S0263-7863(01)00044-8.

R. George and P. Samuel. Particle swarm optimization method based consistency checking in uml class and activity diagrams. In V. Snášel, A. Abraham, P. Krömer, M. Pant, and A. Muda, editors, *Innovations in Bio-Inspired Computing and Applications*, volume 424 of *Advances in Intelligent Systems and Computing*, pages 117–127. Springer, Cham, 2016. doi: https://doi.org/10.1007/978-3-319-28031-8_10. 978-3-319-28031-8.

P.J. Gomes and S. Dahab. Bundling resources across supply chain dyads: The role of modularity and coordination capabilities. *International Journal of Operations and Production Management*, 30(1): 57–74, January 2010. doi: https://doi.org/10.1108/01443571011012370.

N. Gronau and E. M. Kern. Collaborative engineering communities in shipbuilding. In Camarinha-Matos L.M., editor, *Virtual Enterprises and Collaborative Networks*, volume 149 of *PRO-VE*, pages 329–338. IFIP International Federation for Information Processing, Springer, Boston, MA, 2004. doi: https://doi.org/10.1007/1-4020-8139-1_35.

D.E.G. Haartveit, M. Semini, and E. Alfnes. Integration alternatives for ship designers and shipyards. In J. Frick and B.T. Laugen, editors, *Advances in Production Management Systems. Value Networks: Innovation, Technologies, and Management*, volume 384. IFIP Advances in Information and Communication Technology, Springer, Berlin, Heidelberg, 2012. doi: https://doi.org/10.1007/978-3-642-33980-6_35. APMS 2011.

K. Halsbog, J. Moi, and Ø. Schjelstad. Team dynamics in cross-cultural collaboration: A multiple case study on norwegian companies' interaction with asian shipyards. mathesis, Norwegian University of Science and Technology, 2015.

A. P. Hameri, J. Nihtila, and J. Rehn. Document viewpoint on one-of-a-kind delivery process. *International Journal of Production Research*, 37(6):1319–1336, 1999. ISSN 0020-7543. doi: https://doi.org/10.1080/002075499191274.

M.F. Hasmori, I. Said, R. Deraman, N.H. Abas, S. Nagapan, M.H. Ismail, F.S. Khalid, and A.F. Roslan. Significant factors of construction delays among contractors in klang valley and its mitigation. *International Journal of Integrated Engineering*, 10(2):32–36, May 2018. ISSN 2600-7916. doi: https://doi.org/10.30880/ijie.2018.10.02.007. URL https://publisher.uthm.edu.my/ojs/index.php/ijie/article/view/2623/1462.

T. Held. *Supply Chain Network Management*, chapter Supplier Integration as an Improvement Driver – An Analysis of Some Recent Approaches in the Shipbuilding Industry, pages 369–384. Gabler, 2010. ISBN 978-3-8349-6000-9. doi: https://doi.org/10.1007/978-3-8349-6000-9_22.

A.R. Hevner. A three cycle view of design science research. *Scandinavian Journal of Information Systems*, 19(2):Article 4, 2007. URL http://aisel.aisnet.org/sjis/vol19/iss2/4.

A.R. Hevner, S.T. March, J. Park, and S. Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, March 2004.

C. Hicks, T. McGovern, and C.F. Earl. Supply chain management: A strategic issue in engineer to order manufacturing. *Int. J. Prod. Econ*, 65:179–190, 2000.

H.R. Hiremath and M.J. Skibniewski. Object-oriented modeling of construction processes by unified modeling language. *Automation in Construction*, 13(4):447–468, July 2004. ISSN 1060-3425. doi: https://doi.org/10.1016/j.autcon.2003.09.011.

E Holte, A Rialland, and M Westvik. Drivers and trends in global maritime production. innovation in global maritime production–2020 (iglo-mp). Norwegian University of Science and Technology (NTNU), 2009.

V. D. Hunt. *Process Mapping*. John Wiley and Sons, Inc., USA, 1996. ISBN 0471-13281-0.

A.M.A. Hussain, A.A.E. Othman, H.S. Gabr, and T.A. Aziz. Causes and impacts of poor communication in the construction industry. In *Sustainable Infrastructure and Transportation for Furture Cities*, pages 1–11, Aswan, Egypt, December 2018. 2nd International Conference.

N. Iakymenko, A. Romsdal, M. Semini, and J.O. Strandhagen. Engineering change management in the engineer-to-order production environment: Insights from two case studies. In M. Ilkyeong, editor, *Production Management for Data-Driven, Intelligent, Collaborative, and Sustainable Manufacturing*, volume Part I 535 of *APMS 2018*, pages 131–138, Seoul, Korea, August 2018. IFIP WG 5.7 International Conference, Springer. doi: 10.1007/978-3-319-99704-9_17. URL http://hdl.handle.net/11250/2585044. Proceedings.

N. Iakymenko, M. Semini, and J.O. Strandhagen. Aspects for better understanding of engineering changes in shipbuilding projects: In-depth case study. In F. Ameri, K. Stecke, G. von Cieminski, and D. Kiritsis, editors, *Production Management for the Factory of the Future*, volume 566 of *IFIP Advances in Information and Communication Technology*, pages 588–595. IFIP Advances in Information and Communication Technology, Springer, Cham, August 2019. doi: https://doi.org/10.1007/978-3-030-30000-5_72. APMS 2019.

R. Ibrahim and S.Y. Yen. Formalization of the data flow diagram rules for consistency check. *International Journal of Software Engineering and Applications*, 1(4):95–111, October 2010. doi: 10.5121/ijsea.2010.1406.

Iman. lhsgeneral(pd,correlation,n). MATLAB Central File Exchange, 2013. URL https://nl.mathworks.com/matlabcentral/fileexchange/56384-lhsgeneral-pd-correlation-n. MATLAB Function.

ISO. *Quality Management Systems - Fundamentals and Vocabulary*. International Standard Organization, Geneva, iso 9000:2005 edition, 2005. ISO 9000:2005.

H. Janssen. Monte-carlo based uncertainty analysis: Sampling efficiency and sampling convergence. *Reliability Engineering and Sytem Safety*, 109:123–132, January 2013. doi: https://doi.org/10.1016/j.ress.2012.08.003.

A. A. A. Jilani, A. Nadeem, T.H. Kim, and E.S. Cho. Formal representations of the data flow diagram: A survey. In *2008 Advanced Software Engineering and Its Applications*, Hainan Island, China, December 2008. IEEE. doi: 10.1109/ASEA.2008.34.

A.A.A. Jilani, M. Usman, and A. Nadeem. Comparative study on dfd to uml diagrams transformations. *World of Computer Science and Information Technology Journal*, 1(1):10–16, 2011. ISSN 2221-0741. URL https://arxiv.org/abs/1102.4162.

J. Johannessen and A. Hauan. Reaching out for heterarchy (a study of the reorganizing of a norwegian shipyard). *Cybernetics and Systems*, 23(2):169–189, 1992. doi: https://doi.org/10.1080/01969729208927456.

J.A. Johannessen and J. Olaisen. The information intensive organization: A study of governance, control and communication in a norwegian shipyard. *International Journal of Information Mangement*, 13(5):341–354, October 1993. doi: https://doi.org/10.1016/0268-4012(93)90004-N.

K. Kjersem and J. Emblemsvag, editors. *Literature Review on Planning Design and Engineering Activities in Shipbuilding*, Oslo, Norway, June 2014. Proceedings IGLC-22. URL https://pdfs.semanticscholar.org/cf45/212cb84c1ede4518449064bcabe673277aa8.pdf.

B.J. Kolltveit, J.T. Karlsen, and K. Grønhaug. Exploiting opportunities in uncertainty during the early project phase. *Journal of Management in Engineering*, 20(4), October 2004. doi: https://doi.org/10.1061/(ASCE)0742-597X(2004)20:4(134).

L. Lan, Y. Liu, and F. Lu. Discovering hidden tasks and process structure through email logs for dsm. *The Journal of Modern Project Management*, 3(2):243–253, 2015. ISSN 2317-3963. doi: 10.3139/9781569904923.024.

J. Lancaster and K. Cheng. A fitness differential adaptive parameter controlled evolutionary algorithm with application to the design structure matrix. *International Journal of Production Research*, 46(18):5043–5057, September 2008. doi: https://doi.org/10.1080/00207540701324176.

C.L. Le. A method for modeling and verifying of uml 2.0 sequence diagrams using spin. *Journal of Science and Technology on Information Security*, 9(1):20–28, August 2019. URL http://www.isj.vn/index.php/journal_STIS/article/view/37.

M.I. Leje, M.A. Kasimu, and A. F. Kolawole. Impacts of effective communication towards performanceof construction organization. *Path of Science: International Electronic Scientific Journal*, 5(8):3001–3008, August 2019. ISSN 2413-9009. doi: 10.22178/pos.49-4. URL https://pathofscience.org/index.php/ps/article/view/658/655.

V. Levardy and T.R. Browning. An adaptive process model to support product development project management. *IEEE Transactions on Engineering Management*, 2009.

R. Levering, R. Ligthart, N. Noorderhaven, and L. Oerlemans. Continuity and change in interorganizational project practices: The dutch shipbuilding industry, 1950–2010. *International Journal of Project Management*, 31(5):735–747, July 2013. ISSN 0263-7863. doi: https://doi.org/10.1016/j.ijproman.2012.12.010.

W. Li, X. Wang, and S. Rong. Probabilistic load flow method based on modified latin hypercube-important sampling. *Energies*, 11(11), 2018. ISSN 1996-1073. doi: 10.3390/en11113171.

P.E.D. Love, D.J. Edwards, and J. Smith. Contract documentation and the incidence of rework in projects. *Architectural Engineering and Design Management*, 1(4):247–259, 2006. ISSN 1745-2007. doi: https://doi-org.tudelft.idm.oclc.org/10.1080/17452007.2005.9684596.

T. Ludwig, F. Smets, and J. Tholen. Shipbuilding in europe. resreport, Institute Labour and Economy (IAW) and University of Bremen, Frankfurth, Main, June 2009.

M. Lukas, T. Gartner, N. Rohleder, and C. M. Schlick. A simulation model to predict impact of alterations in development processes. In *9TH International Design Structure Matrix Conference*, pages 127–136, Munich, Germany, October 2007. DSM'07. Matlab.

Z.M. Ma, F. Zhang, and L. Yan. Fuzzy information modeling in uml class diagram and relational database models. *Applied Soft Computing*, 11(6):4236–4245, September 2011. doi: https://doi.org/10.1016/j.asoc.2011.03.020.

R.N. Macariola and D.L. Silva. Coping with the information age: Development of a data flow diagram-based knowledge management system for mitigating delays for construction. In *IOP Conference Series: Materials Science and Engineering*, volume 652, Bangkok, Thailand, May 2019. 4th International Conference on Civil Engineering and Materials Science (ICCEMS 2019), IOP Publishing Ltd. doi: 10.1088/1757-899X/652/1/012070.

M.J.K. Malinda. Quality of project documentation as a major risk source in infrastructure projects in south africa. mathesis, Stellenbosch University, March 2017. URL http://scholar.sun.ac.za/handle/10019.1/100947.

C. Meier. Time-cost tradeoffs in product development processes. phdthesis, Technische Universität München, Fakultät für Maschinenwesen, February 2011. URL http://mediatum.ub.tum.de/?id=1063699.

C. Meier, A.A. Yassine, and T.R Browning. Design process sequencing with competent genetic algorithms. *Journal of Mechanical Design*, 129(6):566–585, June 2007. doi: https://doi.org/10.1115/1.2717224.

Mello, Gosling, Naim, Strandhagen, and Brett. Improving coordination in an engineer-to-order supply chain using a soft systems approach. *Production Planning and Control*, 28(2):89–107, 2017. ISSN 1366-5871. doi: 10.1080/09537287.2016.1233471.

M.H. Mello and J.O. Strandhagen. Supply chain management in the shipbuilding industry: challenges and perspectives. *Engineering for the Maritime Environment*, 225(3):261–270, August 2011. doi: 10.1177/1475090211406836. URL https://journals.sagepub.com/doi/abs/10.1177/1475090211406836.

M.H. Mello, J. O Strandhage, and E. Alfnes. The role of coordination in avoiding project delays in an engineer-to-order supply chain. *Journal of Manufacturing Technology Management*, 26(3):429–454, April 2015a. ISSN 1741-038X. doi: 10.1108/JMTM-03-2013-0021.

M.H. Mello, J. O Strandhagen, and E. Alfnes. Analyzing the factors affecting coordination in engineer-to-order supply chain. *International Journal of Operations and Production Management*, 35(7):1005–1031, July 2015b. ISSN 0144-3577. doi: 10.1108/IJOPM-12-2013-0545.

F. Meng, D. Chu, and D. Zhan. Transformation from data flow diagram to uml2.0 activity diagram. In *2010 IEEE International Conference on Progress in Informatics and Computing*, Shanghai, China, December 2010. IEEE. doi: 10.1109/PIC.2010.5687483.

T. Minato. Design documents quality in the japanese construction industry: factors influencing and impacts on construction process. *International Journal of Project Management*, 21(7):537–546, October 2003. ISSN 0263-7863. doi: https://doi.org/10.1016/S0263-7863(02)00083-2.

J. Moon, D. Lee, T. Lee, Ahn. J., J. Shin, K. Yoon, and D. Choi. Group decision procedure to model the dependency structure of complex systems: Framework and case study for critical infrastructures. *Systems Engineering*, 18(4):323–338, July 2015. doi: https://doi.org/10.1002/sys.21306.

NAVAIS. New advanced value added innovative ships. resreport, EU, Rotterdam, The Netherlands, 2020.

Knut Magnus Gjertsen Norbye. Engineering change management in shipbuilding. mathesis, Norges teknisk-naturvitenskapelige universitet NTNU, Trondheim, June 2012.

S. Obaid, S. Asghar, and S. Naeem. Data flow analysis of uml models by alf. *International Journal of Trade, Economics and Finance*, 5(1):12–18, February 2014.

J.T. O'Connor and J. Woo. Proactive approach to engineering and design deliverables quality enhancement. *Journal of Management in Engineering*, 33(4), February 2017. ISSN 0742-597X. doi: 10.1061/(ASCE)ME.1943-5479.0000508.

A.A. Oloufa, Y.A. Hosni, M. Fayez, and P. Axelsson. Using dsm for modeling information flow in construction design projects. *Civil Engineering and Environmental Systems*, 21(2):105–125, June 2004. ISSN 1029-0249. doi: https://doi-org.tudelft.idm.oclc.org/10.1080/10286600310001638474.

P. Otlet. International organization and dissemination of knowledge: Selected essays. Elsevier, Amsterdam, fid 684 edition, 1990.

M. Ozkaya. Are the uml modelling tools powerful enough for practitioners? a literature review. *IET Software*, 13(5):338–354, October 2019. ISSN 1751-8806. doi: 10.1049/iet-sen.2018.5409.

M. Ozkaya and F. Erata. A survey on the practical use of uml for di□erent software architecture viewpoints. *Information and Software Technology*, 121, February 2020. ISSN 0950-5849. doi: https://doi.org/10.1016/j.infsof.2020.106275.

E. Pedersen and J.F. Hatling. Computer integrated ship production. *Journal of Ship Production*, 13(3): 215–223, August 1997. SNAME-JSP-1997-13-3-215.

M. Phililps-Ryder, J. Zuo, and X.H. Jin. Evaluating document quality in construction projects, subcontractors' perspective. *International Journal of Construction Management*, 13(3):77–94, 2013.

I. Rábová. Using uml and petri nets for visualization of business document flow. *Acta univ. agric. et silvic.Mendel. Brun*, LX(2):299–306, 2011.

J.L. Rogers and A.O. Salas. towards a more flexible web-based framework for multidisciplinary design. *Advances in Engineering Software*, 30(7):439–444, July 1999. doi: https://doi.org/10.1016/S0965-9978(99)00006-X.

N. Russell, A. ter Hofstede, W. Van der Aalst, and P. Wohed. On the suitability of uml 2.0 activity diagrams for business process modelling. In Y. Kiyoki, S. Hartmann, and M. Stumptner, editors, *Conceptual Modelling 2006: Proceedings of APCCM2006*, pages 95–104, Australia, Tasmania, Hobart, 2006. Australian Computer Society Inc. URL https://eprints.qut.edu.au/25752/.

N. Sabbaghian. Product development process capture and display using web-based technologies. mathesis, Massachusetts Institutre of Technology, February 1999.

E. Safapour, S. Kermanshachi, and P. Taneja. Investigation and analysis of the rework leading indicators in construction projects: State-of-the-art review. In *CSCE Annual Conference*, volume 294, Laval, Greater Montreal, June 2019.

E. Saliby and F. Pacheco. An empirical evaluation of sampling methods in risk analysis simulation: quasi-monte carlo, descriptive sampling, and latin hypercube sampling. In *Proceedings of the Winter Simulation Conference*, pages 1606–1610 vol.2, San Diego, CA, USA, USA, January 2002. IEEE. doi: 10.1109/WSC.2002.1166440.

M. Sambasivan and Y.W. Soon. Causes and effects of delays in malaysian construction industry. *Int J Proj Manage*, 25(5):517–526, July 2007. doi: https://doi.org/10.1016/j.ijproman.2006.11.007.

Douglas Schwarz. Fast and robust curve intersections. MATLAB Central File Exchange, 2009. URL https://nl.mathworks.com/matlabcentral/fileexchange/11837-fast-and-robust-curve-intersections. MATLAB Function.

S. K. Seaholm, E. Ackerman, and S.-C. Wu. Latin hypercube sampling and the sensitivity analysis of a monte carlo epidemic model. *International Journal of Bio-Medical Computing*, 23(1-2):97–112, October 1988. doi: https://doi.org/10.1016/0020-7101(88)90067-0.

Y. Sered and Y. Reich. Standardization and modularization driven by minimizing overall process effort. *Computer-Aided Design*, 38(5):405–416, May 2006. doi: https://doi.org/10.1016/j.cad.2005.11.005.

A. Sharon, O. L. de Weck, and D. Dori. Improving project–product lifecycle management with model–based design structure matrix: A joint project management and systems engineering approach. *Sytems Engineering*, 16(4):413–426, October 2013. ISSN 1520-6858. doi: 10.1002/sys.21240.

Z. Sladoljev. Search for a model of effective ship production management. *Journal of Ship Production*, 12(4):220–229, November 1996. URL https://www.onepetro.org/journal-paper/SNAME-JSP-1996-12-4-220.

R Slater and A. Radford. Perceptions in the australian building industry of deficiencies in architects' design documentation and the effects on project procurement. *Australasian Journal of Construction Economics and Building*, 8(1), November 2012. ISSN 2204-9029. doi: https://doi.org/10.5130/AJCEB.v8i1.2995.

R.P. Smith and S.D. Eppinger. A predictive model of sequential iterations in engineering design. *Manage. Sci.*, 43:1104–1120, 1997a.

R.P. Smith and S.D. Eppinger. Identifying controlling features of engineering design iteration. *Manage Science*, 43:276–293, 1997b.

R.P. Smith and S.D. Eppinger. Deciding between sequential and parallel tasks in engineering design. *Concurrent Engineering*, 6:15–25, 1998.

M. Solesvik. Interfirm collaboration in the shipbuilding industry: The shipbuilding cycle perspective. *International Journal of Business and Systems Research*, 5(4):388–405, June 2011. URL https://ssrn.com/abstract=2624809.

M.Z. Solesvik and S. Encheva, editors. *Offshoring Decision Making in the Logistics of the Norwegian Shipbuilding Yards*, Dallas, Texas, USA, March 2007. 11th WSEAS International Conference on APPLIED MATHEMATICS.

D.V. Steward. The design structure system: A method for managing the design of complex systems. *IEEE*, EM-28(3), August 1981. ISSN 1558-0040. doi: 10.1109/TEM.1981.6448589.

S.X. Sun, J.L. Zhao, J.F. Nunamaker, and O.R.L. Sheng. Formulating the data-flow perspective for business process management. *Information Systems Research*, 17(4):327–444, December 2006. ISSN 1047-7047. doi: https://doi-org.tudelft.idm.oclc.org/10.1287/isre.1060.0105.

D. Tang, Y.B. Peng, and Z. Liu. Execution sequence planning of computational models based on incidence matrix and design structure matrix. *Chinese Journal of Mechanical Engineering*, 44(12): 173–179, 2008.

W. Tann, H. Shaw, and R. Bronsart. Integrating the collaborative environment in shipbuilding: An implementation strategy. *Journal of Ship Production*, 21(1):37–45, February 2005. ISSN 8756-1417.

P.A. Tilley. Design and documentation deficiency and its impact on steel construction. *Journal of the Australian Institute of Steel Construction*, 32(1), March 1998.

P.A. Tilley and R. Barton. Design and documentation deficiency-causes and effects. In *Proceedings of the First International Conference on Construction Process Reenigineering*, pages 703–712, Gold Coast, Australia, July 1997.

P.A. Tilley and S. Mcfallan. Design and documentation quality survey, designers' perspective, a survey investigating changes in design and documentation quality within australian construction industry and its effect on construction process efficiency. In *CSIRO Building*. Highett, 2000.

P.A. Tilley, S.L. McFallan, and S.N. Tucker. Design and documentation quality and its impact on the construction process. In P. Bowen and R. Hindle, editors, *Customer Satisfaction: a Focus for Research and Practice in Construction*, BCE reprint, page 9, Cape Town, South Africa, September 1999. W65 Joint Triennial Symposium, CIB W55. URL http://hdl.handle.net/102.100/210717?index=1.

Mr. V. S. Tiware and Dr. S. S. Kulkarni. Root cause analysis of variations in construction tasks and developing effective strategies to reduce variations. *International Journal of Scientific and Engineering Research*, 4(9):51–58, September 2013. ISSN 2229-5518.

S.U.R. Toor and S.O. Ogunlana. Problems causing delays in major construction projects in thailand. *Constr Manage Econ*, 26(4):395–408, January 2008. doi: 10.1080/01446190801905406.

K.M. Tran, T.N. Khan, and Y.C. Lan. A framework for transforming artifacts from data flow diagrams to uml. In M.H. Hamza, editor, *Software Engineering 2004*, Innsbruck, Austria, February 2004.

K. Ullah, M.S. Khan, M.T. Lakhiar, A.A. Vighio, and S. Sohu. Ranking of effects of construction delay: Evidence from malaysian buildig projects. *Journal of Applied Engineering Sciences*, 8(21):79–84, May 2018. ISSN 2284-7197. doi: 10.2478/jaes-2018-0011.

D.W.G. van Boekel, T. Dooper, W.J. Ermeling, S.J. James, T.F.W. Janss, and S. van Kemenade. Using dsms on the road to modular ship design. techreport, Technische Universiteit Eindhoven, Eindhoven, January 2020.

T.L.A. van der Mooren. Model based definition in shipbuilding. mathesis, Delft University of Technology, The Netherlands, Delft, March 2017.

S.I. van Niekerk and H. Steyn. Defining 'project success' for a complex project - the case of a nuclear engineering development. *South African Journal of Industrial Engineering*, 22(1):123–136, 2011. ISSN 2224-7890.

L. Wang and T. Raz. Analytic hierarchy process based on data flow diagram. *Computers and Industrial Engineering*, 20(3):355–365, 1991. ISSN 0360-8352. doi: https://doi.org/10.1016/0360-8352(91)90007-S.

T. Wang, S. Guo, and Y. Liu. Pareto process optimization of product development project using bi-objective hybrid genetic algorithm. *Advances in Engineering Softeware*, 65:12–22, November 2013. ISSN 0965-9978. doi: 10.1016/j.advengsoft.2013.05.011.

P.T. Ward. The transformation schema: An extension of the data flow diagram to represent control and timing. *IEEE Transactions on Software Engineering*, 12(2):198–210, February 1986. ISSN 0098-5589.

A. Weippert, S.L. Kajeqski, and P.A. Tilley. The implementation of online information and communication technology (ict) on remote construction projects. *Logistics Information Management*, 16(5):327–340, October 2003. ISSN 0957-6053. doi: https://doi.org/10.1108/09576050310499327.

M. Weske. *Business process management : concepts, languages, architectures*. SpringerBerlin HeidelbergNewYork, 2007. ISBN 978-3-540-73521-2. doi: 10.1007/978-3-540-73522-9.

D. Wesselman. Real time communication in shipbuilding. mathesis, Delft University of Technology, The Netherlands, Delft, June 2017.

S. Westin. Data and information quality research in engineering construction projects: A review of literature. *Building Sustainable Information Systems*, pages 583–594, October 2013. ISSN 978-1-4614-7540-8. doi: 10.1007/978-1-4614-7540-8_45.

S. Westin and T Päivärinta, editors. *INFORMATION QUALITY IN LARGE ENGINEERING AND CONSTRUCTION PROJECTS: A DELPHI CASE STUDY*, 2011. URL http://aisel.aisnet.org/ecis2011/273.

R.I. Whitfield, A.H.B. Duffy, G. Coates, and W. Hills. Efficient process optimization. *Concurrent Engineering*, 11(2):83–92, June 2003. doi: https://doi.org/10.1177/1063293X03032082.

G. Wirtz, M. Weske, and H. Giese. Extending uml with workflow modeling capabilities. In P. Scheuermann and O. Etzion, editors, *Cooperative Information Systems. CoopIS 2000.*, volume 1901, pages 30–41, Berlin, Heidelberg, 2000. Springer. doi: https://doi.org/10.1007/10722620_3. Lecture Notes in Computer Science.

J.C. Wortmann, D.R. Muntslag, and P.J.M. Timmermans. Customer driven manufacturing. 1997.

H.-S. Yan, Z. Wang, and X.-C. Jiao. Modeling, scheduling and simulation of product development process by extended stochastic high-level evaluation petri nets. *Robotics and Computer-Integrated Manufacturing*, 19(4):329–342, August 2003. doi: https://doi.org/10.1016/S0736-5845(03)00027-9. Petri nets.

Yassine. Investigating product development process reliability and robustness using simulation. *Journal of Engineering Design*, 18(6):545–561, 2007. doi: 10.1080/09544820601011690.

A.A. Yassine, D.E. Whitney, and T. Zambito, editors. *Assessment of Rework Probabilities for Simulating Product Development Processes Using the Design Structure Matrix (DSM)*, September 2001. ASME 2001 International Design Engineering Technical Conferences, Computers and Information in Engineering Conference. URL https://dspace.mit.edu/handle/1721.1/1673. DETC2001/DTM-21693.

Ali Yassine and Dan Braha. Complex concurrent engineering and the design structure matrix method. *Concurrent Engineering*, 11(3):165–176, September 2003. doi: 10.1177/106329303034503.

W. U. Ying-Han and S. Heiu-Jou. Knowledge management with xml integrated within the full specification in ship design processes. *Journal of Ship Production*, 20(4):256–261, November 2004. ISSN 8756-1417. URL https://www.onepetro.org/journal-paper/SNAME-JSP-2004-20-4-256.

A. P. Zambito. Using the design structure matrix to streamline automotive hood system development. mathesis, Massachusetts Institute of Technology, January 2000. URL http://hdl.handle.net/1721.1/29159.

Zhao, Lv, and You. Applying dependency structure matrix and monte carlo simulation to predict change in construction project. In *2008 International Conference on Machine Learning and Cybernetics*, pages 670–675. Kunming, 2008. doi: 10.1109/ICMLC.2008.4620489.

M. Zhao and W. Cui. System synthesis model for a hov in conceptual design. 13(13):426–443, 2009.