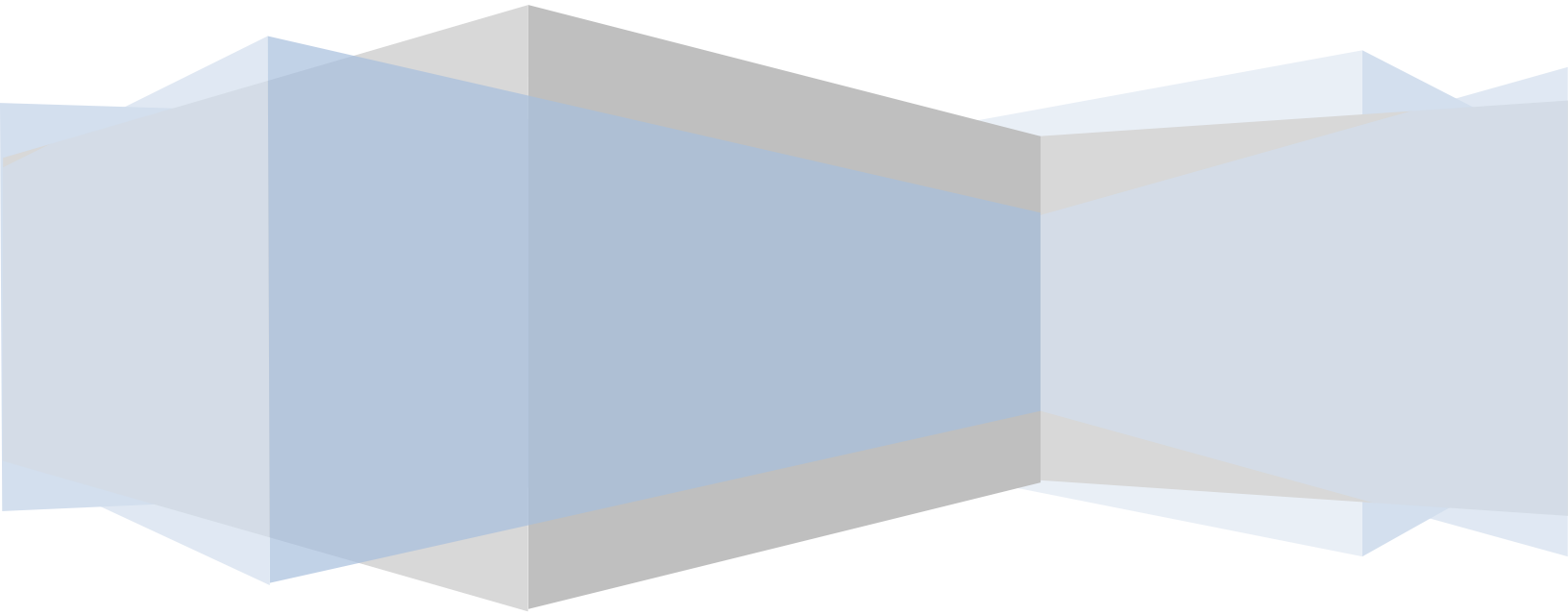
A TECHNOLOGY ROADMAP FOR SOFTWARE PLATFORM PRODUCTS

SATISHKUMAR SUGAVANAM

09 MAY, 2013



TU Delft

Master Thesis

A TECHNOLOGY ROADMAP FOR SOFTWARE PLATFORM PRODUCTS

by

SATISHKUMAR SUGAVANAM
Student Number: 4180461

Graduation Committee:

Dr. ir. Jan Van den Berg, First Supervisor
Associate Professor at ICT Section

Dr.ir. Victor Scholten, Second Supervisor
Assistant Professor at TSE Section

Prof. Dr. Y.H. Tan, Chair
Head of ICT Section

Ir. Paul Van Vuren, External Supervisor
Manager at SMART foundation Platform, Shell

Master Programme Management of Technology
Faculty of Technology, Policy and Management
TU Delft

May 9, 2013
DELFT

ABSTRACT

The business environments are characterized by fast developing technologies, ever more demanding customers & increasingly shorter product life cycles. As a result, sizes of the today's software engineering products are continuously growing. To tackle this, software platform are introduced by exploiting the commonalities among the organization's different software products. For better planning and control, these software platforms are planned on the basis of releases. Organization developing platforms are facing complex platform/product and roadmap dependencies, resource bottlenecks, geographically dispersed stakeholders and time market pressures. Stakeholders using the platform are continuously asking for more and better features during every release of the platform. To give proper answers, technology roadmaps are considered and these roadmaps take many aspects into account. A good technology roadmap is expected to improve requirement management process for the software platform and ensures that satisfactions of different stakeholders are maximized.

In this research project we determined four dimensions of challenges that influence the development of software platforms internally within the organization: requirements engineering, architecture changes, operational & support process and software process disturbances. Together they create challenges for an organization when developing software platforms. After a thorough examination of existing recommendations on technology road mapping approach and business needs, we developed a technology roadmap that supports product mangers in decision making for their strategic release planning & requirements management activities of the platform. The overall validity and reliability of the developed technology roadmap and approach were tested through a case study at Shell by using the roadmap in real release planning activity and through interview with the practitioners.

The developed technology roadmap covers all the aspects of the software engineering context including functional features, technology choices, architectural aspects, operational and support requirements and software process improvements. At the same time, the developed technology roadmap approach gives flexibility to customize the structure of the roadmap according to organizational and business context. The developed technology roadmap can be applied to all kind of software

products, but its main focus is on dealing with software platform & can be used to address the challenges dealt with platform release planning. It incorporates the recommendations found in the literature, as well as organizational requirements regarding the roadmap discovered through interviews. The technology roadmap and customization approach broadens already existing array of TRM approaches in literature by including the software platform field. It helps organizations to understand the current and future situation and an overview of the objectives and needs for new product releases and directions for further developing their technological basis.

ACKNOWLEDGEMENTS

Seven months ago I started this graduation research in ESDV at Shell Global Solutions. My objective was to successfully perform a graduation research with both theoretical and practical relevance and with the result in front of me I believe I have succeeded. This graduation research has given me an invaluable opportunity to work with lot of professional and highly inspirational people.

This graduation research would not have been as successful as it has been without my supervisors and mentors who challenged, pushed and supported me along the way. I would like to express my sincere gratitude and appreciation to my TU Delft supervisors: dr.ir Jan van den Berg and dr.ir Victor Scholten for their advice and great mentorship that provided guidance and direction for my graduation research. Your encouragement, suggestions and constructive criticisms enhanced my writing experience and sharpened my research skills.

I would like to thank my Shell supervisor, Paul van Vuren whose suggestions and advice enriched my knowledge and helped me when I was in doubt, his insights in the wonderful world of Shell and the moments he was there to just lend me an ear. Thank you for your positive influence in growing my vision.

A great part of my gratitude goes to everyone from the ESDV team of Shell. My internship turned out to be an amazing learning experience in such a friendly and yet highly professional environment. Bert C Natalicchio, special thanks to you for giving the comments on my roadmap design and the overall support during the internship.

I would also like to thank all the stakeholders who participated in the interviews and made this research possible by contributing their valuable time and expertise. Finally, I want to thank my family and friends, who have all inspired me, helped me stay focused and finish my research.

- Satishkumar Sugavanam
Delft, April 30, 2013

READER'S GUIDE

The structure of this thesis intends to be self- evident. The research questions are answered chronologically and the chapters aim to elaborate each subject in a coherent and logical manner. The thesis can be broadly divided into four parts: the introduction, the theoretical part of the research based on the literature review (chapter 2 and 4), the case study (chapter 3 and 5) and the conclusion (chapter 7). To really appreciate the complexity and interrelatedness of all the subjects of this research, readers are invited to read to whole thesis.

- Readers interested in the *results* of this research should focus on the introduction (section 1.5), the summary of each chapter (section 2.6, 3.6, 4.6, 5.5) and the conclusion (chapter 7)
- Readers interested in the *methodology of this research* should read the section 1.3, 4.3, 5.1 and 6.4
- Readers interested in the *developed technology roadmap* for software platforms should read chapter 5 and chapter 6
- Readers interested in *implementation of the roadmap* should read section 6.2 and 6.3
- Readers interested in *T- Plan technology roadmap approach* should read section 4.3.1, 4.3.2, 4.3.3
- Readers interested in *major elements influencing software platform* development should read section 2.3
- Readers interested in business need summary should read section 3.5 and *recommendations from literature* should read 4.5

Contents

Abstract ..... **v**

Acknowledgements ..... **vii**

Reader's Guide ..... **ix**

Introduction **1**

1.1 Company Profile and Situation of Shell SMART Solutions Platform 2

1.2 Research Goal 4

1.3 Research Methodology 5

1.3.1 Data sources: 5

1.4 Theoretical GAP 7

1.5 Theoretical and practical relevance 8

1.6 Thesis Structure 9

Developing Software platforms: Challenge for Modern organizations **13**

2.1 Why Software product platforms? 13

2.1.1 Key Definitions 14

2.2 Getting to know about software platforms 15

2.3 Major Elements of Software platforms 17

2.4 Four dimensions influencing the platform challenges 23

2.4.1 Requirements Engineering 24

2.4.2 Architecture Changes 26

2.4.3 Software Process Disturbances: 26

2.5Need for technology roadmap 27

2.6 Summary 28

Case Study of Shell **31**

3.1 Organization 31

3.1.1 Royal Dutch Shell 32

3.1.2 TaCIT 32

3.2 SMART Solutions Platform 33

3.3Need for the technology roadmap 34

3.4 Key findings 36
3.4.1 Requirements Engineering Challenges 36
3.4.2 Architecture Challenges 37
3.4.3 Operational and Support Challenges 38
3.4.4 Software Process Challenges 39
3.4.5 Communication Challenges 39
3.5 Business Needs Summary 39
3.6 Summary 41
Technology Road mapping Approaches: A Literature Review 43
4.1 General Aspects about Roadmaps 44
4.2 Purpose of TRM 45
4.3 Methodology and TRM approaches 46
4.3.1 T- Plan "Fast Start" approach 47
4.3.2 Customization of the T- Plan 49
4.3.3 Implementation Plan 51
4.4 Road mapping in Software product field 52
4.5 Recommendations Summary 53
4.6 Summary 55
Technology Roadmap for Software Platform 57
5.1 Methodology 57
5.2 The Technology roadmap Prerequisites 59
5.3 The Technology roadmap Framework 60
5.3.1 Design Approach 60
5.3.2 Road mapping Process 61
5.4 The roadmap as an integrative tool of previous findings 78
5.5 Summary 80
Roadmap Evaluation 83
6.1 Technology Roadmap Evaluation Using Identified Challenges 83
6.2 Technology Roadmap Evaluation through practioners Interview. 85
6.2.1 The Technology Roadmap added value 86

6.2.2 The Technology Roadmap Implementation Possibility...88
6.2.3 Feedback Overview 89
6.3 Roadmap Evaluation in Release Planning....90
6.4 Validity of Technology Roadmap 93
6.4.1How to Validate 93
6.4.2 Evaluation of the research process. 93
6.4.3 External Validity 96
6.5 Summary 97
Conclusions & Contributions 99
7.1 Main Contributions....99
7.1.1 Reflections on Theory and Contribution 101
7.1.2 Reflections from Case (Shell – SSP Platform) 102
7.2 Research Limitations....104
7.3 Future Research Possibilities....105
Bibliography 107
Appendix 113
A.1 SMART Solutions....113
A 1.1.SMART Solutions Platform (SSP) - HOW WILL it work?....113
A.2Interview Outline and Questions 114
A 2.1 General Interview Outline 114
A 2.2 Questions Asked to the Stakeholders:....114
A.3 Technology Road mapping from Literature 116
A.4 Bulls Eye....118

List of Figures

FIGURE 1 RESEARCH QUESTIONS..... 5
FIGURE 2 THESIS STRUCTURE 10
FIGURE 3 EXAMPLES OF PLATFORM 14
FIGURE 4THREE PARADIGMS OF PLATFORM DEVELOPMENT 17
FIGURE 5 MAJOR ELEMENTS OF SOFTWARE PLATFORMS .. 18
FIGURE 6 PRODUCT FEATURE MAPPING 19
FIGURE 7 ORGANIZATION STRUCTURE 21
FIGURE 8 PLATFORM AND APPLICATION DEVELOPMENT CYCLE 22
FIGURE 9 FOUR DIMENSIONS OF CHALLENGES 24
FIGURE 10 SSP PLATFORM 34
FIGURE 11 STAKEHOLDERS FROM SSP PLATFORM..... 35
FIGURE 12 MULTI LAYER ROADMAP..... 45
FIGURE 13 T- PLAN APPROACH 50
FIGURE 14 REFERENCE FRAMEWORK FOR SOFTWARE PRODUCT MANAGEMENT 52
FIGURE 15 ROAD MAPPING PROCESS..... 63
FIGURE 16 LAYER TYPE ROADMAP..... 64
FIGURE 17 BULLS EYE FORMAT 66
FIGURE 18 GANTT CHART FORMAT..... 67
FIGURE 19 ARCHITECTURE OF THE TECHNOLOGY ROADMAP..... 68
FIGURE 20 APPLICATION LAYER..... 69
FIGURE 21 FUNCTIONAL FEATURE LAYER..... 69
FIGURE 22 TECHNOLOGY CAPABILITY LAYER..... 70
FIGURE 23 ARCHITECTURE CHANGES LAYER..... 71
FIGURE 24 OPERATIONAL REQUIREMENTS..... 72

FIGURE 25 PROCESS IMPROVEMENTS 72
FIGURE 26 WORKSHOPS 74
FIGURE 27 HOSTED IN SHAREPOINT (DIGITAL ROADMAP STORAGE) 76
FIGURE 28 IMPLEMENTATION PROPOSAL 89
FIGURE 29REFERENCE FRAMEWORK FOR RELEASE PLANNING 91
FIGURE 30 STAKEHOLDER VOTING 91
FIGURE 31 REQUIREMENT SCENARIOS FROM ROADMAP 92

List of Tables

TABLE 1 STAKEHOLDER 20
TABLE 2 BUSINESS NEED FOR THE TECHNOLOGY ROADMAP 40
TABLE 3 RECOMMENDATIONS FROM LITERATURE..... 54
TABLE 4 MAINTENANCE INTERVALS..... 77
TABLE 5 IMPLEMENTATION OF FINDINGS IN THE ROADMAP..... 78
TABLE 6 CHALLENGES 84
TABLE 7 DESIGN EVALUATION METHODS 94

1

INTRODUCTION

Nowadays, the rate of technology change, complexity & cost are increasing while sources of technology and competition are becoming more global [1]. Organizations are expected to manage their technology assets more strategically and be more responsive to technical change [1] [2]. The business environments are characterized by fast developing technologies, ever-more-demanding customers & increasingly shorter product life cycles [3]. As a result, complexity and size of today's software engineering products are continuously growing. Many companies are now focusing on the commonalities between the different products and capturing those in software platforms as associated set of reusable assets rather than developing software from scratch for each product [1] [4]. The vision of any software platform is to set up the platform as a common base across multiple application product portfolios [5].

Since stakeholders of the application products need information in order to plan their activities and communicate with end users, the demand for an overall view of the software platform and its future release has become important [2] [4]. Therefore developing high quality and reliable software platform within a budget and on time requires a well co ordinate and executed software process [6].

Among the many tools available *"technology – road mapping represents a promising approach to manage high-level view of technologies, link aspects of business with requirements engineering and planning of the firm"* [7]. The technology roadmap combines the technology and market perspectives over time by bringing together the different business processes, considering time dimensions and both relevant internal and external factors [8].

Technology road mapping (TRM) process has been widely adopted in many different organization and industries [8]. Likewise, Shell Global Solutions (Shell), the sponsor & project principal for this master thesis has decided that technology road mapping can be of a high value for them. The quest for technological innovation is high for Shell, since they are acting in industry and market where product and process technologies

are of a vital importance to be competitively superior. One such technological innovation to ensure consistency and technical integrity for the software products used in their oil & gas solutions are now being built upon on one common software platform called SMART Solutions Foundation Platform (SSP). In turn Shell can realize maximum value from their existing investments in technology – helping them save on further IT Costs.

Awareness about long term technology management & to make the platform a robust and maintainable product led to the decision to customize a TRM Process and to implement it into the organization in order to support release planning process for the software platform.

Although, it is well known that such TRM process has to be specifically customized to the subject organization [9] [10] [8], there is one significant aspect about Shell (SMART Solutions Platform) that demands special attention when designing and implementing technology road mapping process. Here difference is that they produce complex software platform to host software product applications.

This is exactly where focus of this thesis lies on – technology road map and its customization approach in the context of the software platform. What the specific situation of Shell is, what exactly technology road mapping means for the software platforms, and why this topic is of relevance for a master thesis are explained in the following sections of this chapter.

1.1 Company Profile and Situation of Shell SMART Solutions Platform

Shell is a global group of energy and petrochemical companies. The company is active in more than 80 countries and territories and employs 90,000 people worldwide. More information about the company profile and organization context of this research is explained in section 3.1.

Recently, TaCIT (Technical & Competitive IT) one of the business unit within Shell, started establishing software platforms to support faster development of innovative application and deliver value quickly to business. SMART Solutions Foundation platform (SSP) is one such program within TaCIT aimed at delivering a common software platform to develop SMART applications to monitor Shell assets around the globe. SSP delivers the common functionality required in the areas of

SMART applications by exploiting the commonalities among the organizations different systems or products.

For better planning and control, the SMART solutions platforms are planned on the basis of releases or release planning. The success of the SSP platform depends on the release of new capabilities/features and upgrades required by the SMART application within the right time.

Product Managers & stakeholders of the SMART applications are continuously asking for more and better features from SMART solutions platform. But which ones should be finally selected for the next release and why? And which features not enough attractive are better to left (or postponed)? Ultimate accountability for selecting which requirements/features are implemented in a version of the software platform lies with the platform product manager. In addition, product manager are now faced with endless product planning challenges: complex platform/product and roadmap dependencies, resource bottlenecks, and time – to market pressures and geographically dispersed stakeholders and project teams [11].

As a consequence, the SMART Solutions platform team has initiated an improvement activity to tackle these problems and to develop a technology roadmap. Technology roadmap development is integral part of evolutionary software platform products & the goal of any product manager is to determine the most promising roadmap such that the total degree of satisfaction of all different stakeholders is maximized [12].

The project started in SMART Solutions platform team at Shell. As introduced above – technology road mapping is chosen method to support this process. The main goals of the project are: to develop a technology roadmap and approach suited for Shell – SMART Solution foundation platform situation, to derive a first technology roadmap covering all the technology choices, best practices, process improvements and functional and non-functional requirements, and to integrate the technology road mapping process in order to make this process a continuous organizational process.

As part of the project team and an intern at Shell, my task is to develop an appropriate technology roadmap and process, to accompany and support the implementation and to document the entire procedure. Shell expectations with respect to this master thesis, and therefore my tasks for this study are to facilitate the development of technology roadmap, to validate the technology roadmap in one of the release

planning activities and to give a recommendation for their future implementation.

1.2 Research Goal

Decision – making considerations described in the previous section eventually form a big question for product managers on how to create an effective technology road mapping procedures to support the strategic release planning activities of the platform. There is a critical need for the technology roadmap to cover all the aspects of the software engineering process in line with both platform and application portfolio's existing requirements. This technology roadmap should give an answer to when, what and most importantly, which features are released to enable the application portfolios as well as make the platform product robust and maintainable.

The goal of this thesis research is defined as follows:

"To design a technology roadmap for software platform products"

Such a technology roadmap with an implementable plan covering the aspects of software engineering process will provide guidance for product managers to articulate the requirements and provide visibility to content for the subsequent release planning activities. The visualization of research questions is depicted in figure 1.

In order to gain a profound understanding of the problem environment and in-depth knowledge on current business goals and challenges, the following sub-questions should be addressed prior to design of technology roadmap.

1. Why is developing software product platform a challenge for modern organizations?
2. How Shell company develop software platforms? What are the main requirements to be addressed by the technology roadmaps?
3. What are the various road mapping approaches discussed in the literature for developing an effective technology road map?
4. What should be the process of arriving at technology roadmap taking into account the information gathered from previous questions?
5. What is the validity of the developed technology roadmap in practice?

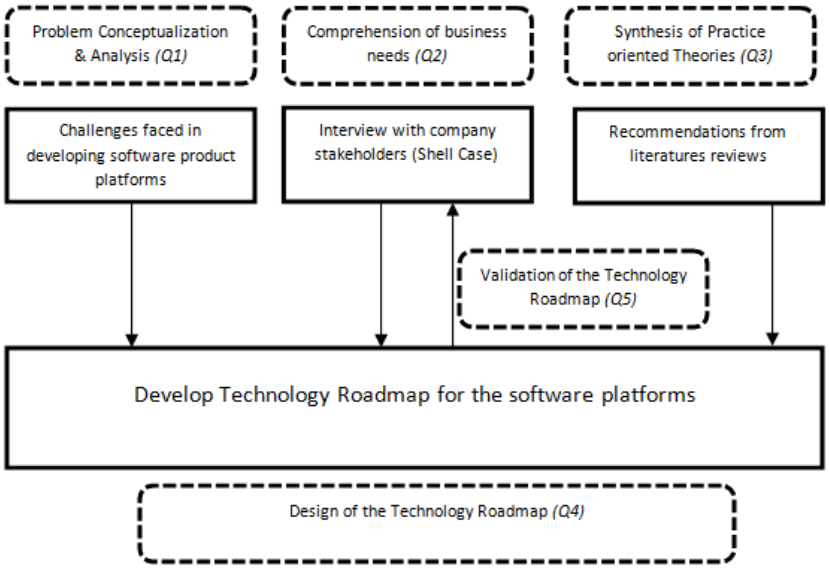


FIGURE 1 RESEARCH QUESTIONS

1.3 Research Methodology

The thesis methodology not only accords the project planning from Shell, but also follows the design science methodology, in which research is done through process of building and evaluating artefacts for solving business problems (see section 5.1 & 6.4 for more description and justification of the methodology). This master thesis is based on the Design Science research approach in detail elaborated by Hevner et al. A design science research is a set of nested problems in which the top level problem is always a practical problem. The main question of this research is a practical problem on designing a technology roadmap, which is decomposed into set of knowledge sub problems (Q1 – Q3) and practical sub problems (Q4-Q5) introduced in the previous subsection. Seven guidelines for design research are presented by Hevner in his article [13]. The first three are important and explicitly used during this research. The first is the guideline to design an artefact, in this case a technology roadmap. Secondly, the artefact has to be relevant with regard to a business problem. Finally, the designed artefact has to be evaluated rigoursly.

1.3.1 Data sources:

During this research, we made use of several data collection sources most importantly literature research, semi structured interview with experts, stakeholder workshops and internal project documents (gap assessment documents, requirements management plans, reference architectures,

solution architectures, improvement plans, technology maturity plan, investment proposals and project records of previous projects) from Shell.

The literature study was based on a multitude of papers describing specific challenges and major elements within the field of software platforms (Q1). This research method is preferred as it allows for analysing and processing a high amount of rich information in short time span. Next, we performed a case research at Shell SSP platform by interviewing stakeholders to comprehend the business needs of technology road mapping and research relevance (Q2). The stakeholders were identified on the organisation structure and work partitioning description from literature and based on the current organization context of SMART Solutions program. Based on the identified challenges and requirements from business, major customization approaches on technology roadmaps from scientific articles were identified to assure rigor of the study (Q3). The next step was to design (Q4) the technology roadmap and process by using the inputs from the literature and case study. In the development of technology roadmap and customization approach multiple iterative steps were made. These iterations are omitted from this thesis for reasons of readability, however they were crucial to leverage the comprehensiveness of this thesis and thereby it's scientific value. The iterations were triggered by changing needs of the stakeholders in Shell because of changing perception and growing experience with technology roadmaps. To cope with changing needs constant feedback were collected during the design stage by conducting stakeholder workshops. Also, new insights from the literature and feedback from discussions with supervisors, stakeholders from Shell and practioners from vendor organizations triggered iterations while customization of the technology roadmap. The technology roadmap that was developed was validated by expert interviews at Shell SMART solution team and applying it to Shell Case in one of the release planning activities (Q5).

This research is conducted in the following phases that are aimed at answering the identified sub questions and explained below:

1 Problem conceptualization (Q1): In this phase the study focus is established. We discuss the challenges faced during the development of software platform products including, requirements engineering, architecture changes, and operation and support process & software process disturbances. We show that technology roadmap is becoming an important process for

modern organization to plan the releases for the software platforms. The major elements influencing the software platform development is also discussed.

2 Problem analysis & Comprehension of business needs (Q2): In this phase we discuss the current "state-of-the-art" software platform products and development approach from Shell Case. Results from semi-structured interviews are presented and critically examined in order to acquire a thorough understanding of business needs regarding roadmaps. These requirements are addressed by the roadmap to assure the research relevance.

3 Synthesis of Practice- Oriented Theories (Q3): In this phase we summarize major customization approaches on technology roadmap from scientific articles. These recommendations are integrated in the roadmap design to assure rigor of the study.

4 Design stage (Q4). Here, we design the technology road map & customize the approach for SSP platform in accordance with the acquired understanding of the business needs and synthesized knowledge.

5 Validation stage (Q5). Here, we validate the technology roadmap by applying to the Shell case and asking for expert opinion. Findings are relevant for future improvements of the technology roadmaps. They also set new research opportunities regarding technology roadmaps.

1.4 THEORETICAL GAP

That Shell (SSP Platform) wants to have a technology roadmap developed is in itself not a challenging problem. The literature offers several viable processes for the developing technology roadmap and their customization [8] [14] [11]. However, the situation of software platform was never an issue in the literature, though demands attention.

Since Motorola developed the approach in 1980 in order to support their long- range strategic planning, much has been written about technology- road mapping [15]. This approach has been adopted by a number of organizations according to their specific needs. *Phaal and Farrukh*, two researchers from Cambridge University, in 1997 have started to pioneer in the theoretical development of TRM by studying this method, which has led very much to generalization of the methods. Many papers have been published about technology roadmap since then,

including how & where technology road mapping can be applied to a specific organization in a specific context.

The literature suggests methods to customize the architecture and process to the specific needs of the organization because of the generic nature of the developed TRM Process [10] [16] [17] [14] . *Lee & Park* states that even though TRM is useful and flexible approach, still there is a problem to accommodate unusual circumstances and how to make the TRM approach fit to specific needs [9]. It is important to carefully alter the generic road mapping process to address business environment and firm specific circumstances, if there is a misfit; it leads to problems in organizations [10]. This is the reason, there are numerous article have been published about TRM process applied to specific circumstances, in organization & industry level to meet the specific, most contextual requirements [8] [14] [11].

Despite the many publications, the contextual requirements of the technology road mapping approach for the software platform are disregarded until now. Technology road mapping literature including the approaches of customizing the TRM process, only address the elements of the road mapping that are subject to change. There is little practical support available on how to actually design the process is – apart from general proposition that is should be adapted – not given for specific circumstances, including the software platform. An important starting point is that there is no commonly agreed technology roadmap and customization approach for software platforms. In addition, the customization and application of the TRM approach in the software engineering field is less investigated and rather new [12].

Although the TRM literature is extensive, this leads to conclusion, that there is still a gap in the literature about how a technology roadmap and its customization process might look like for the situation of a software platform in an organization.

1.5 THEORETICAL AND PRACTICAL RELEVANCE

Having define what the thesis is about both theoretically and practically, the question is now what the initial relevance is (section 7.1 elaborates exact contribution to theory and practice based on the findings)

For theory, this thesis complements existing technology road mapping literature and describe the lessons learned from developing and evaluating own technology roadmap & customization process for

software platform from Shell case and usage of this approach in a software engineering context. Further, the construct of software platform development and the linkage that is created and analysed between a technology road mapping and organization developing software platforms is clarified.

The practical contribution is more direct and most obvious and valid for Shell SMART Solutions platform. They finally receive a theoretically and practically robust technology roadmap covering all the aspects of software engineering process to make their platform product robust and maintainable. Recommendations for further implementation increase the likelihood for a successful outcome of their project.

In a broader context, however, it can be helpful for other software platform development organizations willing to implement such a process. The developed technology road mapping process is more suitable for that context and a good technology roadmap is expected to improve cross functional processes required for the creation of new application on the software platform. Furthermore, technology road mapping in software platform product can address the challenge dealt with product release planning – the process for the next release of an evolving product. Achieving such an objective is particularly critical in the development of software product platforms [18].

1.6 Thesis Structure

The set-up of the master thesis is aligned with the requirements and set-up of the Shell SSP Platform project. This has many advantages, such as that the experts in the organization can be involved in designing the process, that may yield valuable feedback for the further improvement of the process. Thus, based on the research questions and the set-up of the project, the master thesis is structured as depicted in figure 2

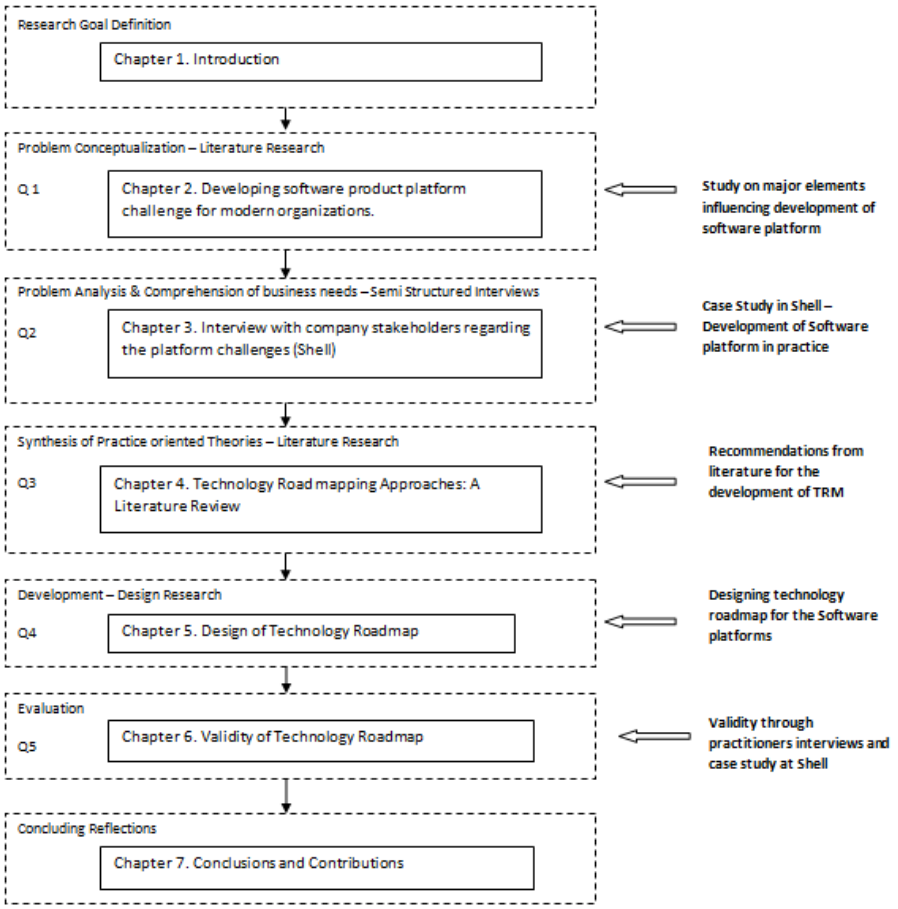


FIGURE 2 THESIS STRUCTURE

Chapter two introduces the major elements of the software platform that needs to be covered by the technology roadmap and challenges discussed in the literature and industry white papers. Chapter three introduces the case of Shell (SMART Solutions Platform) and requirements that lead to need of the technology roadmap. The challenges that need to be addressed by technology roadmap are explored by interviews. Chapter four is devoted to an extensive and critical review of literature. Different approaches to technology road mapping processes and its customization are reviewed, as well as the literature that is relevant for the aspect of software engineering field. This chapter lays down the foundation for a theory-based technology road mapping approach. In chapter five, the technology roadmap and process is developed. This includes the methodology, consisting of the technology road mapping process and final technology roadmap for the software platforms. In chapter six the developed roadmap is validated by expert interviews and

applying to the case of Shell SSP Platform in the release planning activities. Chapter seven concludes and discusses the thesis, especially in the light of the practical and theoretical contributions and additionally, further research in the field is suggested.

2

Developing Software platforms: Challenge for Modern organizations

2.1 Why Software product platforms?

The interaction between technology and business focused innovation is becoming diffused in today's erratic and changeable global economy [19]. For firms in different industries, especially in the high-tech sector, changing their centralized business approach is fundamental to remain highly competitive. This change mainly follows the innovation paradigm by adopting the remedy approach of "*developing software platforms*" [20].

Developing software platforms is a commonly used technique to compress product development lead time, in order to release a product that is perceived as "new" by making minor changes [18]. Many companies are now focusing on the commonalities between the different products and capturing those in product platform as associated set of reusable assets rather than developing software from scratch for each product [21].

Companies that successfully deploy software product platforms can achieve magnitude growth over a decade and reach major business milestones [19] [22]. Well-designed platform architecture for software products provides *productivity benefits* and enables rapid *growth in market share and revenue.* The benefits of a robust platform include the ability to more rapidly and flexibly create products tailored for particular niches within the company's business needs [19]. This approach can lead to market advantage through timely new product introduction; a richer product family covering a broader scope of a particular market, and ultimately, barriers to entry for competitors who lack an equivalently robust product line [23].

A Software product platform also offers important market advantages enabled by platform thinking and execution. If the platform is

built and method & techniques are communicated clearly, other companies or individuals can build modules that operate in or on the underlying platform. This software product platform has created the opportunity to become the standard or basis of *large scale innovation* [23].

2.1.1 KEY DEFINITIONS

In this research we refer to the software platform as [24]

"*a set of subsystems and interfaces that form a common structure from which a stream of derivate products can be efficiently developed and produced*"

(or)

defining it as a "*technical foundation on which several software application products are based*".

There are various examples (figure 3) of firms adopting "software platforms" like Intel, Microsoft (operating system), Cisco (one platform kit), Google, Apple (iphone product platform); Amazon (e- commerce platform) and SAP (core system that serves as the basis for all SAP components). These examples of software platforms have attracted innovation networks including variety of stakeholders (technology vendors, service providers, developers, customers, scientific communities & competitors) [23].

end-user programming	MS Excel, Mathematica, VHDL	Yahoo! Pipes, Microsoft PopFly, Google's mashup editor	*none so far*
application	MS Office	SalesForce, eBay, Amazon, Ning	*none so far*
operating system	MS Windows, Linux, Apple OS X	Google AppEngine, Yahoo developer, Coghead, Bungee Labs	Nokia S60, Palm, Android, iPhone
category / platform	**desktop**	**web**	**mobile**

FIGURE 3 EXAMPLES OF PLATFORM [4]

These different software platforms are being developed to position the products as external applications. This master thesis though looks at developing technology roadmaps for software platforms that are

developed within organizations to enhance the internal capabilities and deliver larger value to their own business.

It is important to understand what elements comprise of software platforms, who represents the platform and product team & what aspects of software engineering process are followed in large organizations. Being fully aware of challenges faced by the firms while developing software platform approach is the first step in developing technology roadmap for the software platform products. The remaining part of this chapter is aimed at answering these questions.

2.2 GETTING TO KNOW ABOUT SOFTWARE PLATFORMS

Several patterns have emerged that many organizations can successfully elevate their software development capability according to the character of the business changes. One pattern corresponds to a set of operational practices that we call the *software platform development paradigm* [23]. In this section, we try to establish the background and evolution of the platform development paradigm [19] [23] [23] [24] [25]. The three software development paradigms are mentioned in figure 4.

Paradigm I Serial Development Projects [23] – In Paradigm 1, early stage of new product's lifecycle is characterized by series of independent development projects. The architectures of the product are often poorly documented from software engineering aspects. Little structure and formality of work is established for development teams from organization structure aspect, as long as there is continuity between the initial product team and team that develop next products. Organizations choose different lifecycles for its development effort and development methodologies, however typically they adopt waterfall development model or an evolutionary delivery approach. In this paradigm, the time taken between the beginning of the project and its release to customers is defined as the time to market (TTM). Continuous learning and improvement process between two successive projects significantly reduces TTM for latter projects.

Paradigm II Multiple Parallel projects [25] [23]- When the products find increased market acceptance and greater demands from business, customers demand follow on products with shorter intervals between them. This put pressure to do faster product releases by reducing the time between successive products (TBSP). This in turn shifts the product

development organization to multiple parallel projects. This is paradigm II. This paradigm usually results in multiple teams working parallel to build closely related products. Typically, many organization follow paradigm II as their predominant mode of operation.

Also, concept of reuse functionality is introduced, to achieve greater reduction in TTM and TBSP. Reusable software components are designed with the future products in mind, that they can be plugged into new products without any modifications. Development process used to build reusable software components have significant differences compared to Paradigm I and follow agile project methodologies [23].

From organization structure aspect, in order to provide large degree of autonomy and independence, each individual product teams will have, dedicated project managers and architects. On the other hand, this makes little difficult to co ordinate and share the work between teams. The other main challenge for organizations following paradigm II is to make to the practice of reuse happening predictably across projects [23].

Paradigm III Platform Developments [19] [23]- The paradigm III is essentially an extension of paradigm II. In this paradigm, the common elements within product family are factored out and developed into platforms. The essence of platform development is to identify the product features, elements & subsystems that are well understood and stable, that provides a basis for differentiating features and value add for the products [26].

The composition of the platform development for a product family can range from a set of individual components to a collection of subsystems and to a complete product framework. The platform can contribute from 10% to nearly 90% to individual products in terms of development or code effort. Typically, products developed using pervasive structures (e.g. GUI standards & error handling) and features are resided within the platform to have much shorter TTM [23].

Generally, new features and functionality can be provided by either product software or platform. In some cases, new features are implemented as part of product, if it involves large degree of uncertainty. The new feature or the product functionality is migrated into the platform once it is stabilized and becomes available to next product development efforts [21].

TTM is reduced for individual projects as the result of implementing paradigm III. Better market responsiveness is enabled and organization involved in highly competitive environment develops whole series of platforms to support different product families [22].

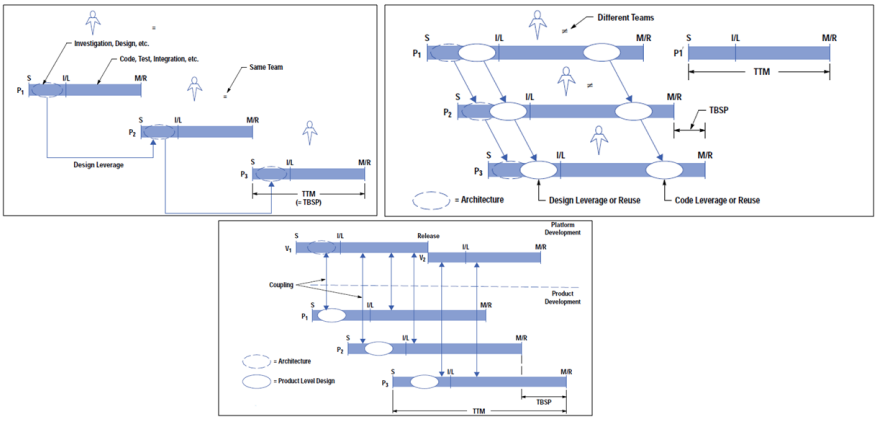


FIGURE 4 THREE PARADIGMS OF PLATFORM DEVELOPMENT [23]

2.3 MAJOR ELEMENTS OF SOFTWARE PLATFORMS

Firms adopting the software platform approach form a dedicated team supported by senior management to assess the firm's future potential and current situation. In planning a new software platform, the team assesses three main parts carefully – *their offerings, employed technologies and the stakeholders they control & target* [27]. On top of this, platform team also understands target user's requirements, recommends a new platform architecture and strategy; and proposes an implementation plan, budget and timeline. To achieve a competitive advantage, the firms need to enforce an effective software platform strategy [28].

To achieve an effective software platform strategy, companies execute excellent software engineering process and practices in place [6]. In general, the current literature on software platforms outlines the core elements (figure 5) that make up an organizations software platform development system [23] [29] [19] [23].

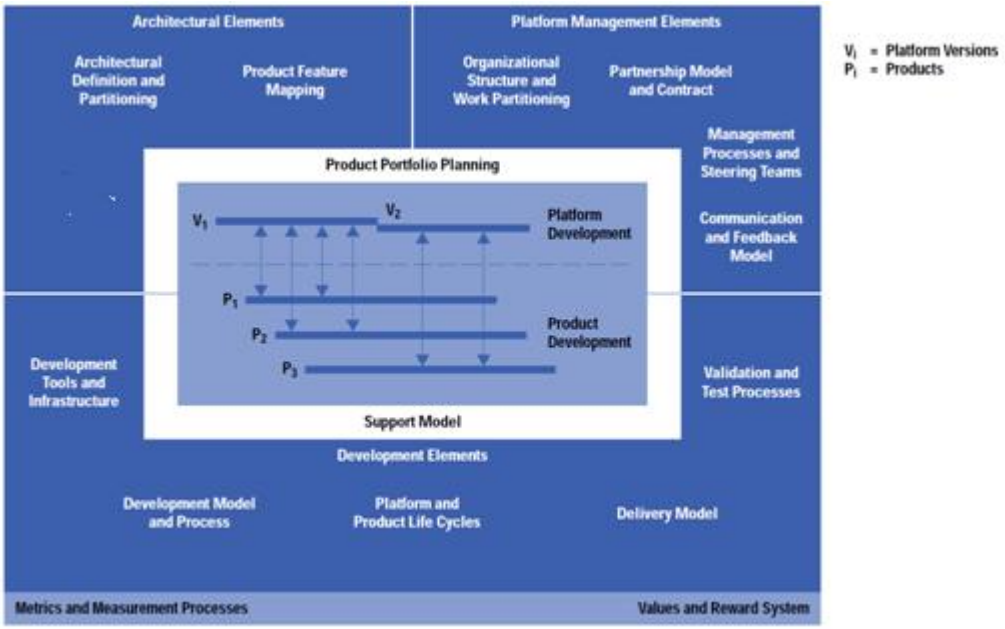


FIGURE 5 MAJOR ELEMENTS OF SOFTWARE PLATFORMS [23]

Each individual element of the model addresses a particular aspect of how an organization's development system works. The model is holographic, since each of the elements contains references to aspects of the other elements.

1. **Product Portfolio Planning:** The strategic relationship between the product versions and the platform are defined in this element. It also further identifies the key priorities & business drivers, sets the overall goal & direction and funding for the platform strategy. The *roadmap for the software platforms* are encompassed in this element and are part of the organization already, but more as a way to visualize than as a process. Platform team develops set of new functionalities /capabilities by gathering requirements from product teams and delivers them in each releases/versions [23] [30].
2. **Architectural Elements:**
   - *Architecture definition and partitioning:* This element includes major functional and technology subsystems and the interfaces between them. The shared challenge for platform and product architects is to determine where to draw the platform and product boundary. The boundary needs to be drawn so that it balances the foundation and the infrastructure provided by the platform with the amount of flexibility needed to support value added product features [23] [18].

- *Product Feature Mapping:* This element involves the translation of customer needs into product features to specific platform or product modules. The identified subsystems and components modules are used in the implementation of each feature. Figure 6 illustrates the decision making process those developers & architects go through to map the features to platform and product architecture [31].

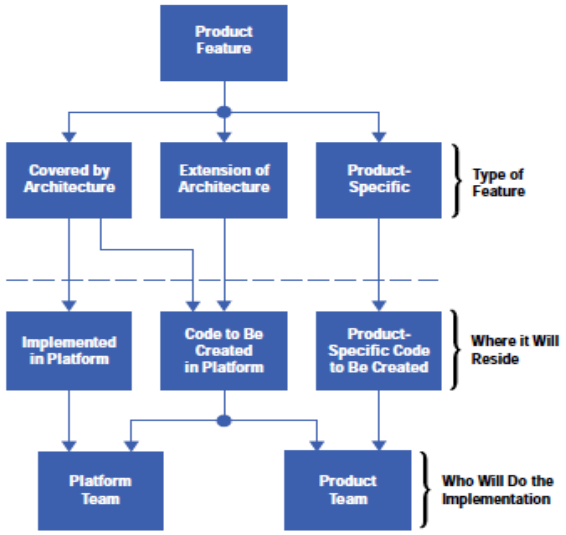


FIGURE 6 PRODUCT FEATURE MAPPING [23]

3. **Platform Management Elements** [23]**:**
   - *Organizational Structure and work partitioning*: This element defines the context within which teams are configured, how teams communicate with one another and organizations operations model at an abstract level. (e.g. reporting relationships and team organizations). The difference in roles between platform development and traditional development paradigms are at the heart organization structure [18] [23]. Platform development does not result in creation of new roles within an organization; rather it causes existing roles to become explicit and more formal. The brief definition of each of these roles is contained in below table 1

TABLE 1 STAKEHOLDER [23]

Role	Responsibilities
Program Manager	Integrate and co-ordinate all these functions involved in developing the platform and products & responsible for funding within the organization.
Platform Product Manager	Typically owns the platform and is responsible for managing requirements and defining releases (what features/functionalities are delivered in each releases).
Business Analyst	Responsible for requirements engineering process (gather, analyse, prioritize the requirements coming from different stakeholders)
Architects	Responsible for high level design of subsystems/components of the platform and involved in deciding the technology choices for the platform products.
Project Manager	Develop and maintain the project budget and schedule allocate resources and manage the delivery of each releases/versions of the platform.
Product Manager Applications	Typically owns the products and responsible for providing the functional features required from the platform and manages the development of application products on the platforms.
Test Manger	Develop the test strategy and architecture, including the selection, implementation, and execution of appropriate testing methods and procedures.
Operations Manager	Responsible for creating the support model strategy for assisting the product teams once they use platform features/functionalities.

These precise set of roles and responsibilities depends on the particular case, an organization still can distinguish general group of stakeholders depending on their motives and capabilities. For this research, we use these roles as the identified stakeholder groups for interviews in later chapter 3. The general organization structure for the platform development paradigm is illustrated in the figure 7:

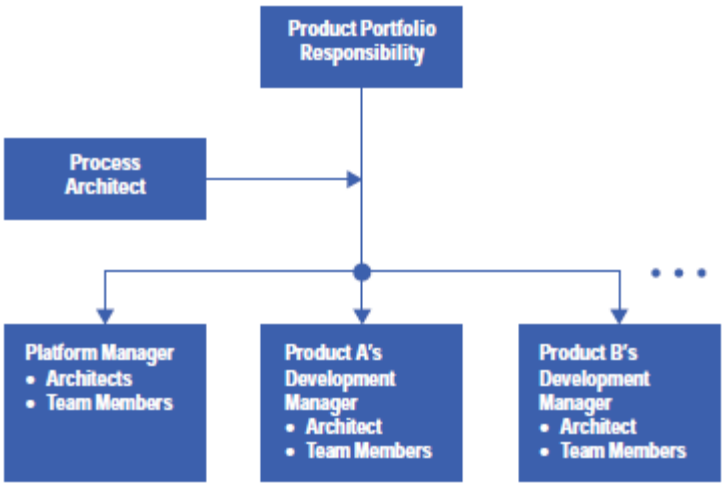


FIGURE 7 ORGANIZATION STRUCTURE [23]

- *Partnership Model & Contract:* This element provides the generic framework for instantiating the operating model between product and platform teams (e.g., expectations for their working relationships & interdependence between teams). This also further extends to the third party technology vendors, service providers & system integrators for the development of the software platforms [23] [21].
- *Management Process and Steering Teams:* This element defines the creation of product portfolio plan and how its execution is managed.
- *Communication and Feedback Model:* The success of a platform development depends on the strength of its communication and feedback paths. Good communication & feedback paths between the product and platform teams supports rapid decision making and reduces unexpected surprises. The timing and content of the information that flows between teams is further defined by this element.

4. **Development Elements** [23] [30] [32]:
   - *Platform and Product Lifecycles:* This defines the major phases, with checkpoints, goals, activities, and deliverables for both the products & platforms. Figure 8 shows the underlying structures of the platform and product life cycles. The major phases of the two lifecycles are similar and dependent, since the platform architecture and components flow from the platform into products.

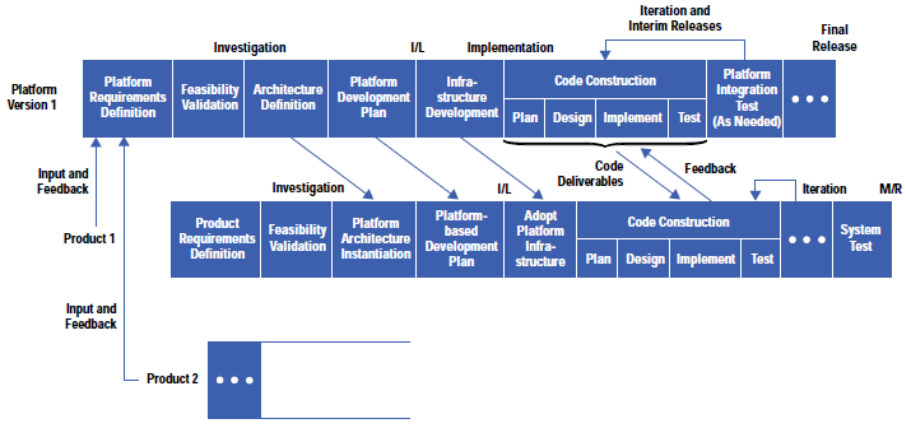


FIGURE 8 PLATFORM AND APPLICATION DEVELOPMENT CYCLE [23]

- *Development Model and Development process:* This element describes how the new functionality is created in the platform. They cover the process used in the creation and enhancement of a platform module through its integration into products [23].
- *Delivery Model:* This element defines how platform release/versions are deployed for the application products use. The platform delivery model provide details on what is delivered and how it is delivered, while the platform life cycle outlines the types of artefacts that get delivered and when they are delivered.
- *Validation and Test process:* This element defines the overall test strategy and architecture, including the selection, implementation and execution of testing methods and procedures. Verification and validation activities are performed as early as possible in the platform development lifecycles to catch defects early and to reduce the overall development effort and shorten back-end cycle time [6] [23].
- *Development tools and infrastructure:* This element provides a common development environment and process for product and platform work (e.g., procedures and tools for creating, finding, building, storing and testing components) [23] .

5. **Support model** [23]**:** In this element, the detailed support requirements are identified and the mechanics of how product teams and platforms work together are established. The support model provides the details on types of support provided, the overall service level expectation and the delivery mechanism through which it is delivered. It also further includes both ongoing supports during implementation and initial training. The activities like providing

documentation, making defect repairs, delivering training, answering questions, escalation path for resolving issues and releasing enhancements are covered by support model & process. In the platform development, support model is created explicitly for in house product development work. With the increase in number of product teams, the need for formal structures in support model and process in place increases simultaneously [23].

6. **Value Metrics and Measurement process** [26]: This element defines the way by which results and progress for each of the above mentioned elements are tracked to ensure achievement of business goals. The value of platform investments and benefits are typically tracked over the period of years to justify the investment proposal from the product portfolio planning [23].

2.4 FOUR DIMENSIONS INFLUENCING THE PLATFORM CHALLENGES

The first step in developing technology roadmap for software platforms within organizations is to look inside the black box of elements and define the main dimensions that shape the evolutions of software product platforms. Trying to satisfy the requirements of all elements, a company faces various challenges to deal with (introduced in section 3.4) in order to choose the most appropriate technology roadmap and process.

To get better understanding of the challenges faced by the organizations, we investigated the available literature regarding the platforms to identify the dimensions that influence/initiate the challenges. Van de weed et al. state that, the one main dimension that influence challenges for the platform products is "*origins of requirements related to characteristics of requirements engineering*" [12]. Architecture changes are mentioned by Brinkkemper and Richard as another determinant that influence challenges [2]. In general these claims correlate with the findings of Bosch [25], who in their research on compositionality of software platforms conclude challenges faced by the software platforms are influenced by three forces *"Requirements engineering, architecture changes and operational & support process"*

These authors however not consider the software process and practice itself as a motivation to influence the challenges. Still, more and more paper on software product development emphasize that good software engineering process and practices is a key components to successfully mitigate the most of the challenges and also reduce the

chance of their occurrence in the future [6]. Software platforms are being developed & implemented within firms to enable rich stream of application products aiming at delivering business value to their organizations [26]. In this case, good software process & practices followed in the software platform team can become a "life saver" for many application products [23].

As a consequence, in this research we add the fourth determinant software process & practices to the three ones previously identified by Bosch. Eventually, it gives us four determinants (or "dimensions", as we refer in this research) influencing the challenges of the software platforms: *Requirements engineering, architecture changes, operational & support process and process improvements (Figure 9)*. We chose the name "process improvements" rather than "software process & practices" to get better representation of what is meant by this dimension – included the development & test process and lesson learnt for continues improvements of the elements around software platform. The rest of the section explains how each dimension influences the challenges in developing software product platforms.

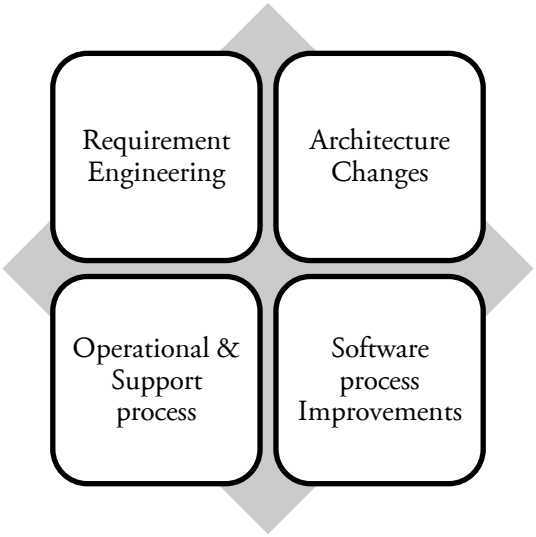


FIGURE 9 FOUR DIMENSIONS OF CHALLENGES

2.4.1 REQUIREMENTS ENGINEERING

Requirement engineering is an essential phase in the lifecycle of the platform development from the software engineering aspects [27]. In addition to the conventional requirement engineering challenges, software platform address new issues like – handling the continuous inflow of requirements from various product teams, architects, testers, developers, product managers and end users located across from different

geographical locations. It also needs to strike a right balance between catering to the needs of cutting edge features and being backward compatible with existing features and finally to align to the roadmaps of various product groups [31]. These requirement engineering challenges are highly prevalent for a software platform development and traditional requirements engineering techniques alone cannot handle these issues [33]. These entire issues combine together further adds to the ambiguity of the elicited requirements. *Implementing a standard mechanism of communication between the applications that use the platform and the platform team reduces requirements ambiguity* [20].

2.4.1.1 Multiple groups of stakeholders

Development of software platform faces multiple challenges when it comes to catering to the needs of stakeholders having various demands, expectations and diversified roadmaps [31]. Stakeholder expectation is the main challenge faced by platform group and to integrate application requirements seamlessly with the platform deliverables. This comes with stakeholder's expectation like flexibility of the platform group in accepting the evolving & changing stakeholder requirements, ease of stakeholder product development and providing frequent in progress with stable deliveries [12]. Here the platform group cannot be inflexible to the requirements scope stability from its stakeholders (application product teams). This is because platform group's existence directly depends on the spectra of demand created by application product units. In other words, software platform do not directly generate value, but value of the platform mostly lies in the options it creates to build applications and depends on the applications that can be implemented [28] [31].

2.6.3 Unresponsiveness of platform development:

Early in the maturation cycle of the application product categories, the slow release cycle of the software platforms are particularly frustrating. Many times, a new feature is required quickly in a new application product which required changes to platform components [4]. The platform is unable to respond to the request of the product team due to the slow release cycles. The platform team also does not allow the product team to implement new functionality or feature itself due to the potential consequences for the quality of the platform and product team [19].

2.4.2 ARCHITECTURE CHANGES

Implicit dependencies: In early software platform development approach, to increase the short term developer productivity, a high degree of connectivity between components in platform is accepted, as there are few disadvantages at this stage [19]. However, later when the scope of the product increases, the components need to be developed in more creative configuration and in this stage, implicit dependencies between components becomes a significant problem for the creation of new products [19] [29].

Incorporation of immature functionality: In the initial scope of the platform development there is often a tendency to incorporate a product specific functionality which is useful for one product and likely to become relevant for other products over time. Sometimes this further extends to incorporation of product specific functionality very early on before it is used in the first products. As the scope of the product family increases the disadvantages of incorporating immature functionality become apparent [29] [19].

Dependencies on external software: Software platform does not have control over the release cycles, development process and roadmap if it's integrating the external software from other vendors. This leads to number of difficulties and makes it more challenging to develop an architecture which suits all the application products and external software [22].

2.4.3 SOFTWARE PROCESS DISTURBANCES:

Developing reliable & high quality software platform within budget and on time requires a well-coordinated and executed software process. Investing in software process improvements can actually help the platform development in long term and pay off. However, external market and business demands (requirements prioritization goes high priority to application/business requirements) create disturbances in the software process [6]. This subsequently reduces the usefulness of the software process improvement and as a consequence the time and cost of the software platform development increases. Some of the process disturbances discussed in the literature includes:

Uncertain customer requirements: Often business or the end user knows their requirements but have problems in formulating them. This

leads changes later in the development and lot of requirement uncertainty. Many development efforts are underutilized due to lack of information to prioritize requirements. In these cases, a formal requirement specification will be helpful requirements management process [32] [31].

Non – official releases: Software product undergoes many process (e.g. QA) related activities before the official releases. At many times quality of the product or an official release (delay in releases – link with operational requirements) is compromised to meet the release dates [28]. Many "backdoor or unofficial" releases or patches are generated by the time of official release. Soon after the releases, a service pack with patches is released [6].

Tao Management: "A loose controlled management do not work well in a highly "process thinking" environment". "Test Squelch" is one such example which explains Tao Management. Testing and deployments are usually one of the last phases in the lifecycle of the software platform development. Typically, delay in the development & reverse engineering of the release dates (set release date first and determine how/when it can be done) causes the testing & deployment cycle to be reduced to the absolute minimum [31] [6].

2.5 NEED FOR TECHNOLOGY ROADMAP

The release of new features/capabilities/functionalities and platform upgrades with the right amount of features and quality within an open market window determines the success of the software platforms [22]. To achieve this, a methodological approach for managing the content, roles of platform architecture and future product releases and timing is needed. Such an approach is often missing in practice, due to lack of suitable process infrastructures, time to market pressures, unclear priorities and inexperience [18].

Managing these three parameters including content, roles of future product releases and timing based on the market information available is the pre requisites for timely delivery of good quality software platform [18].

- *Content* is about deciding which features to be included in which releases and refer to linking product features to business requirements and technology availability.

- *Timing* refers to making trade-offs between quality, functionality and time, identifying and exploiting a window of opportunity and assessing the product based on market and business needs.
- *Roles* is about intended business implications for the company, planned audience for the releases and refers to major, minor and patch release types

Due to virtually infinite "wish-list" from the business and external market pressure and limited resources, a calibrated process & well balanced roadmap is required [23]. This roadmap manages faster delivery of the software to the business and helps to streamline the development of the software product. Also, this roadmap will continuously align & track the process activities in the lifecycle of the software product development by establishing the process accountability in each of the releases. For both external reference and internal development calibrated technology roadmap is important [6].

The technology roadmap expresses the new innovation capabilities, operational support process and process improvements for the software product management, changes to underlying architecture & technology, composition of individual releases and release and development schedules [33]. Complexity can be reduced by showing the different types of possible features/capabilities required at initial stage of development by early product characterizing [34]. The roadmap also helps in setting customer expectations and formalizes requirements specifications which are crucial to ease the later external pressure of meeting business demand [33].

2.6 SUMMARY

In this chapter we showed how the development & evolution of the software product platforms has slowly become the challenge for the modern organizations. We revealed the major elements of the software platforms, their characteristics in terms of the software engineering process and what challenges are faced by the teams when developing the software platforms.

Frequent communication & coordination between platform and product teams was shown as important part of platform development system. The next step is to explore the state of art development of the software platforms internally in the organizations by exploring the case of Shell. This will provide us more requirements and business needs for the development of technology roadmap.

The next step is to collect all relevant recommendations from practice and theory that will provide a solid base for the developing technology roadmap and to ensure that it is built upon an existing knowledge base and considers the current business needs regarding technology roadmap for software platforms. .

3

Case Study of Shell

The previous chapter elaborated on major elements and existing challenges faced by the organizations when developing software platforms from the scientific literatures as well as industry white papers. Now the goal is to understand "state-of-the art" within an organization that develops software platforms internally. This development is, especially in the Oil & Gas industry, a logical development since software is an increasingly part of products and often defines the competitive advantage. The foundational principles of moving towards the software platform in the Oil & Gas industry is to ensure consistency and technical integrity for the applications built on platforms [35].

Oil & Gas industry are demanding software solutions that enable them to take advantage of new technologies including mobility, cloud services, apps, social computing and platforms that unlock the potential of Big Data [35]. Shell along with other industry partners is leading this initiative to next generation of solutions and working alongside the global oil & gas industry partners to ensure that latest versions of these technologies provide the foundation for their oilfield solutions [36].

The remainder of the chapter is aimed at exploring the context of the development of the software platform at Shell and their need for the technology roadmap and process. The insights, requirements and understanding of in-depth context of the environment and stakeholder needs will enable to look for the recommendations from the literature to customize the road mapping process and develop technology roadmap that suit the specific needs of the Software platforms at Shell.

3.1 Organization

This technology road mapping research for software product platforms was conducted at Shell Global Solutions in one of their business unit "TaCIT" (Technical & Competitive IT). This section explains the organizational context of this research to enable readers to put the scope of the research into perspective. First, I provide a high-level overview of Royal Dutch shell and then describe the scope of this

research on SMART foundation platform which fits into the bigger picture.

3.1.1 ROYAL DUTCH SHELL

Shell is a global group of energy and petrochemical companies. The company is active in more than 80 countries and territories and employs 90,000people worldwide. According to the corporate website, Shell's business strategy is '**to be the world's most competitive and innovative energy company**'. Royal Dutch Shell plc is build up from four different businesses and eleven different functions. Because these functions and businesses overlap each other in all sorts of combinations, it is not possible to depict this in one comprehensive figure [36].
Businesses are:

- Upstream International
- Upstream Americas
- **Projects & Technology**
- Downstream

And some of the functions are following:

- Finance
- HR & Corporate
- Legal
- Government Relations
- IT

Projects & Technology (P&T) department works closely with other two businesses (Upstream & Downstream) to deliver top quartile projects & wells, providing differentiating technologies, driving technical & operational excellence and enhance the competitive advantage. P&T consist of nine business units such as Innovation, Research & Development (IRD), Contracting & Procurement (PTC), Projects (PTP), Upstream Development and TaCIT (Technical and Competitive IT) etc., Our research of technology road mapping process falls under business unit of TaCIT [36].

3.1.2 TACIT

TaCIT is one of the business units within Projects & Technology department whose strategic role is to leverage information technologies for attaining competitive benefit for the business strategies of Shell. They provide enterprise – wide competitive and industry specific information technology across all business functions including upstream, downstream

& projects & Technology. One of the TaCIT strategies is to develop innovative applications and deliver quickly to business. The means to achieve this is by establishing software platforms that support development and deployment of applications.

3.2 SMART SOLUTIONS PLATFORM

SMART Solutions is one such program within TaCIT aimed at delivering a platform to develop applications to monitor Shell assets around the globe to make informed decisions. More information about the SMART solutions can be found in Appendix A.1. SMART solutions program mainly consist of:

- *SMART Solutions Platform (SSP)* delivers the common functionality required in the area of SMART Applications. Functionality and services delivered through foundation platform include common *portal & visualization* components using SharePoint, *calculations* engines to support modelling (ranging from simple excel based models to professional simulation software), standardized access to data using a common *data model* and reusable *data connections*. Advanced functions like *event monitoring* and processing, business rules, portable mobile use are also added to the foundation to simplify development and reuse of applications. The SMART solutions platform is delivered as a service; capacity is created and supported globally where the SMART application requirements drive the speed and location of implementation. The overall management and health of the SMART solutions platform is managed through a set of *landscape services* that support overall support ranging from application and data to the infrastructure level. The get an overview of SMART solutions platform refer to the below figure10

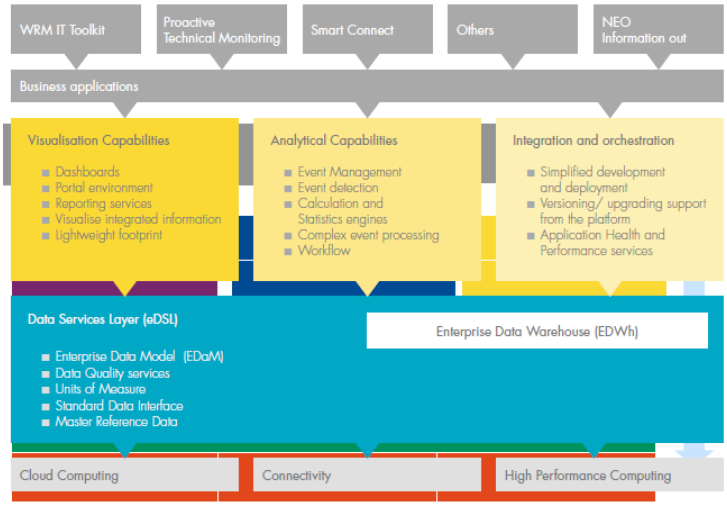


FIGURE 10 SSP PLATFORM [36]

- *SMART Applications* are created through the functionalities, configuration of capabilities & services' that are available from the SSP platform.

Our research mainly focuses on developing technology roadmap and process for the SMART solutions platform which is the software platform product. The more details on how the SSP works is provided in the Appendix A.1.

3.3 NEED FOR THE TECHNOLOGY ROADMAP

The need for the technology roadmap and process for the SSP platform was identified as one of the improvement activity to *support the release planning and requirements management process* and to *manage and communicate the delivery of functionalities.* The exhaustive business requirements for developing such technology roadmap were identified through stakeholder interviews. The following stakeholders (figure 11) were identified from the *organizational structure and work partitioning* element from the previous chapter (2.3.3.1) and based on the current organization context of SMART Solutions program.

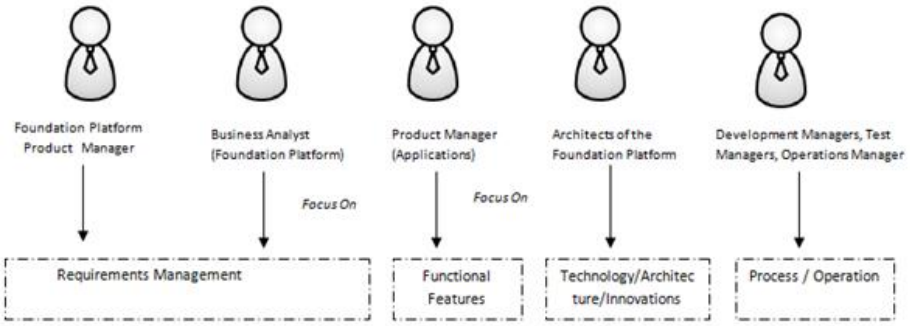


FIGURE 11 STAKEHOLDERS FROM SSP PLATFORM

- Platform Product Manager who manages the SMART foundation platforms
- Business Analyst from the Foundation Platform who is involved in requirements engineering process (gather, analyse, prioritize the requirements coming from different stakeholders).
- Product Manager of the application teams and their perspective on the foundation platform
- Architects of the foundation platform who is involved in deciding technology choices for the software product platforms
- Development, Test & Operations manager who manages the development, testing & deployment of the functional features on the platform for the application products.

By ensuring that all above stakeholders participate in the interviews, this research is able to draw insights on challenges faced by the organization in developing software product platforms in all four dimensions discussed earlier.

The interviews were planned as semi-structured, with a short introduction in the beginning for better understanding of the stakeholder positions within the program. A short list of standardized questions was developed to initiate a conversation; the list can be found in Appendix A.2. Later, depending on the answers, we could continue with open questions, trying to find specific details and challenges faced by the stakeholders in the current process of the platform development. Questions were developed with the goal in mind to know more about:

- The SMART foundation program and about the foundation platform

- Current process (Development, Release Management and Change Management) followed in SMART Foundation Platform
- Components/work streams/technologies in the SMART foundation Platform
- Issues, challenges and opportunities perceived by the stakeholders.

3.4 KEY FINDINGS

The key findings from the interview were grouped according to the four dimensions that influences challenges faced by the platform.

3.4.1 REQUIREMENTS ENGINEERING CHALLENGES

The first challenge is the requirements management process. This is mainly concerned with the allocation of the requirements to platform components, by separating application specific and platform requirements. Business analysts of the platform face *complexities in aligning the requirements* from application teams with internal initiatives coming from the foundation team.

The platform product manager needs to get the right content to make informed decisions on the release planning activities. The platform team further faces challenges in identifying capabilities for the foundation platform which can be used by many application product teams. Platform team needs to balance the priorities to deliver foundation business commitments while executing the improvement activities to the foundation internal process.

- Expect to have better oversight of known requirements from all the elements influencing the platform and other inputs to avoid last minute fire drills in the requirements selection process.

3.4.1.1 Functional Features

The product Managers of the application teams like to have *visibility on functional features delivery beyond the current releases from the foundation platform*. They expect a roadmap from the foundation team highlighting what capabilities/feature sets are delivered in each release of the foundation platform. The product groups that make use of the platform have different roadmaps for their products with respect to the core functionality and the release dates. While the platform group gathers and analyzes requirements it is imperative to factor these product roadmaps for two reasons:

1. Understand if the platform fits into the customer product.
2. Challenge between new features and release stability.

Product managers of the application team expect the SMART foundation platform to have their own vision in terms of technology choices, baseline capabilities and standard components, that will ease the application teams and help them align better with the foundation capabilities. The Foundation platform has to deliver both *business driven and technology push capabilities* in order cater the features/functionalities of the application products and incorporate technology changes needed to develop platform architecture.

-Expect visibility on functional features delivery beyond the current releases from the foundation platform & foundation roadmap highlighting capabilities/features sets delivered in each release.

3.4.1.2 Emergence of Functional Features

SMART platform have many components/ work streams/ subsystems, which differ, based on the functionality provided and the applications catered to. The platform group has to make the application product teams aware of these components/ work streams/ subsystems families and their features. This knowledge will trigger the application product teams while raising requirements for feature enhancements of the platform. The platform group has the responsibility of understanding the products that are based on the platform. The platform group can generate a clear roadmap for itself based on the usage of the platform in the products and by monitoring the trend of the customer requirements. This helps in creating further requirements and managing the platform better.

-Make aware of emergence of new features/capabilities and identify general functional capabilities for the platform based on future applications.

3.4.2 ARCHITECTURE CHALLENGES

The challenge is to evolve the architecture of the SMART platform according to the vision of the foundation platform. The foundation platform needs to identify general capabilities for the foundation platform based on the future SMART applications. The new requirements must be allocated to the different parts of the architecture, and in response to these requirements, the architecture must be adapted. This typically results in added and removed components, but the primary

area of concern is generally the impact on interfaces among existing components and, by extension, the teams responsible for these components. The challenge is to *have architecture drive consistently, across applications and platform, and to have the correct balance between architecture and functional requirements.*

-Incorporation of technology into midterm plan and make them visible in release plan & need to find balance between architecture and functional requirements.

3.4.2.1 Communication challenges

The architectural knowledge of the platform product is contained within the head of a single architect or small group of architects. Increasing complexity and simultaneous pull from many application teams often forces architects to spend most of their time communicating *and supporting others with the product architecture. The high level content of the architecture has to be documented and should be easily shared and communicated to other stakeholders.* This will support development teams and can provide feedback to the platform architects so that they can tune and evolve the platform and product architecture.

- Evolution of the architecture and its content has to be documented and should be readily available for reference and communication through roadmap

3.4.3 Operational and Support Challenges

The SMART solutions platform integrates new technologies from the vendors and onboard SMART applications continuously. To manage these integrations, the release cycle of the platform should be frequent and deployment time of each releases (or) new functionality should be reduced. The challenge is to identify & incorporate new ideas and methodologies to improve the operational process of the platform by reducing the downtime and increasing the availability and maintainability of the platform. Also, support model and process are vague and not clear for the application teams.

- *Identify & incorporate new ideas to improve the operational process of the platform.*

- *Document the lessons learnt from previous release & gather improvement ideas from stakeholders to step up the support process*

3.4.4 Software Process Challenges

The ideas and initiatives to improve the test and development process should be gathered and implemented to make the SMART platform product a robust and maintainable. Sufficient visibility should be given for the improvement of software engineering process (Test and development process) around the platform.

- *Gather ideas and initiatives to improve the software engineering process around the SMART platform.*

To counter the operational, support and process challenges, some of the platform improvements and best practices have to be well documented. Those include:

- *to improve the operational and support process around the platform by reducing the downtime, increasing the availability and maintainability of the platform.*
- *to enhance the release & deployment process by improving the development, testing and deployment strategies*

3.4.5 Communication Challenges

The key challenge in SMART platform is coping with the need for wider dissemination of information. As the number of interdependent teams increases, the number of stakeholders with interest in a given piece of information increases. Putting together a *communication model in the form of visual graphical format diagram* helps teams identify who needs to know about plans, assignments, issues, status, best practices, and successes.

- *Communication model in form of visual graphical format.*

3.5 Business Needs Summary

The majority of takeaways from the previous sections could be generalized, to a greater or lesser extent, in the business needs of an organization that is looking for technology roadmaps for the software product platforms. Below (table 2), we summarized the major business

needs when it comes to development of the roadmap for the software platforms:

TABLE 2 BUSINESS NEED FOR THE TECHNOLOGY ROADMAP

Business Requirements	Dimensions
- *To support the release planning and requirements management process* - *Manage and communicate the delivery of functionalities* - *Better oversight of known requirements from all the elements influencing platform release to avoid last minute fire drills in the requirements selection process.* - *Visibility on functional features delivery beyond the current releases from the foundation platform* - *Identify general functional capabilities for the platform based on future applications.*	Requirements Engineering Challenges
- *Incorporation of technology into midterm plan* - *Evolution of the architecture and its content has to be documented and should be readily available for reference and communication*	Architecture Challenges
- *Improve the operational and support process around the platform by reducing the downtime, increasing the availability and maintainability of the platform* - Step up the support process and make it visible for all the application teams	Operations and Support challenges
- *Platform needs to enhance the release and deployment process by improving development & Testing strategies.* - *Gather ideas and initiatives to improve the software engineering process around the SMART platform*	Software process Challenges
- *Document the lessons learnt from previous releases & gather improvement plans from stakeholders* - *Communication model in form of visual graphical format.*	General.

These requirements should be implemented in the development of technology roadmap and process to reach the intention of this research in solving an organizational problem. Later, it will be shown how each of the business requirements is addressed in the proposed technology roadmap.

3.6 SUMMARY

This Chapter presented Shell company's challenges on developing software product platforms and its main preferences and concerns for developing technology roadmap. It was a good case study since it allowed us to get the feedback on the specific challenges faced by the software product platforms.

The collected information is related to the particular company and causes certain difficulties in attempts to *generalize* it. In the end of the chapter, we tried to summarize those business needs that might stay relatively the same across all organizations developing software product platforms. We incorporate these needs in the final technology roadmap.

From the interviews we see that the major challenges of the current situation with respect to development of software platforms - is the requirements engineering challenges, architecture challenges, operations and support challenges & software process challenges.

Taking all these findings together, we see that there is a big need in having a technology roadmap that captures the high level features from (application product teams, architects, process improvements and operational requirements) and to be used as high level release calendar for the platform and baseline or the scope management for each platform release. This roadmap should further acts as a standard tool to communicate the release plans in visual intuitive manner. At this moment we have the first half of information required for the design science research - the knowledge base obtained from the business needs and problem environment to assure research relevance. The next step is to collect all relevant recommendations from theory that addresses the introduced challenges by customizing the technology roadmaps. These recommendations will be necessary to provide a solid base for building roadmap; to ensure that it is built upon current business needs and considers the existing knowledge base.

4

Technology Road mapping Approaches: A Literature Review

Having identified organizational challenges on developing software platforms, the next step is to review current scientific literature that may give useful recommendations on how to outline the roadmap process leading to the technology roadmap. Such review is essential for this thesis to guarantee that the final outcome is built upon existing knowledge and that the working advices have been taking into account.

Although, technology roadmap is a useful and flexible approach, an important starting point is that there is *no commonly agreed technology road map and process* [9]. It is true for any type of business & organizations, not only for software platforms. Every technology road map and customization approach has a unique set of traits plus it has to be often customized to specific needs of the organization and business context, so the academic and practitioner experts agree that road mapping process vary widely across organizations [37] [38] [14].

To have an effective release cycle of software platform features, it is first and very important for organization to have technology roadmap that will help managers (both platform and product) to articulate the requirements and provide visibility to content for the subsequent release planning activities [12]. This technology roadmap should assist in dealing with every challenge category we introduced earlier.

In this chapter, first we discuss general aspects about roadmaps. Next we discuss common types of TRM customization approaches leading to the technology roadmap. In the end, we gather the major recommendations for the customization process in one table to use them later in the design of the roadmap for Shell Case.

4.1 General Aspects about Roadmaps

Technology road mapping (TRM) can defined formally as a flexible technique that is used within the industry to support strategic and long range planning [39]. According to Lauro Lehtola the basic purpose of the TRM is to explore and communicate dynamic linkages between markets, technologies, products and markets over time [16]. Due to the popularity of the term technology road mapping, Kappel notes, defining road mapping is challenging action, since all kinds of forward – looking documents can be called roadmaps [10]. Motorola, developed technology roadmap more than 2 decades ago, from where it has been spread to other firms widely. For example in 1998, Bob Galvin, CEO of Motorola during the period, provides the following definition when road mapping was established [40]:

"A 'roadmap' is an extended look at the future of a chosen field of inquiry composed from the collective knowledge and imagination of the brightest drivers of change in that field".

The importance of expertise and knowledge plays in the process, its flexibility and the forward- looking nature approach is emphasized in this definition. At present, major companies including Philips, Lockheed-Martin, Philips, Corning and Lucent Technologies are utilizing the TRM techniques [9].

Visualization of the technology roadmap is generally two dimensioned and multi layered (see figure 12). The milestones and activities are allocated with time dimension along the horizontal axis. The business areas and their planning activities and milestones are placed on the vertical axis. Based on the context of the organization in subject and purpose both axes can be customized as shown in the next section [8] [41] [39] [38].

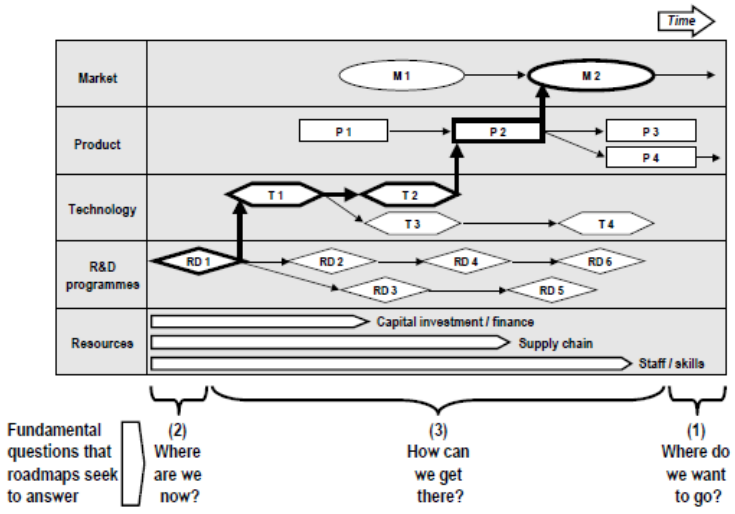


FIGURE 12 MULTI LAYER ROADMAP [8]

Technology roadmaps will seek to answer three simple questions even though they can take various forms [42].

1) *Where are we going?*
2) *Where are we now?*
3) *How can we get there?*

Albright and Groenveld, explains the multilayered roadmap can be constructed from a *market pull and technology push perspective*. The roadmap defines the most important requirements needed by customers from market pull perspective and from technology push perspective; it defines the key new innovations and technologies [43] [7]. According to Kappel and Phaal the road mapping activity can be both goal oriented – by defining strategies to realize clearly defined targets and exploratory – surveying future possibilities [10] [38]. These perspectives will actually support us in developing the technology roadmap and process for the software platforms. The integration of these perspectives is defined in the below table – 3 later in the chapter.

4.2 PURPOSE OF TRM

Phaal captured and classified technology roadmaps by its need and purpose into eight groups. These groups are product planning, strategic planning, long range planning, program planning, process planning, capability planning, knowledge asset planning and integration planning. Apart from the design of the roadmap, the process is different for each of the roadmap and depends on purpose of the roadmap [41].

4.3 METHODOLOGY AND TRM APPROACHES

What we noticed during the literature review is that, in general, there are many different approaches for customization of the road mapping process. Many authors have developed their own approach & customization of the road mapping process. More than twenty approaches could be identified in total and most of them differ in the kind of roadmap being produced and therefore the purpose being served [37] [14] [44].

For instance, Kappel and Albright define the road mapping process based on an organization's specific needs by including the stages of initiation, maintenance and restart process [46] [43] [10]. Lehtola identified three road mapping process: preparation, approval and communication [16]. A fairly similar three- stage product road mapping process has been proposed by Phaal including planning, facilitated road mapping workshop(s), and roll-out [44]. On the contrary, Van de Weerd defines product road mapping process consisting of theme identification, core asset identification and roadmap construction [45]. Further to this process McCarthy adds two more phases to the road mapping process that consists of team formation, focus, technology, implementation and review [46].

Four step model for creating and updating product roadmaps was proposed by Vahaniity which includes, (1) defining the strategic mission and vision of the company and outline of the product vision, (2) scan the environment for identifying major trends, (3) distil and revise the product vision as product roadmap, (4) estimate & evaluate the product life cycle and mix of the development efforts planned. To ensure up-to-date new information and to adjust the roadmap the steps in the model should be performed periodically [47] . In the Appendix A.3 – a summary of the existing road mapping process is presented including their main focus areas i.e. goals and main phases.

Not every one of them of the above mentioned road mapping process serves as an optimal basis for this thesis. Fulfilling a certain purpose is the first requirement and criterion for the technology road mapping process [14]. The purpose of this study depends on the previously described problem context at developing software platform internally within the organization. The purpose to be fulfilled is to provide overview of requirements influencing the software platform and support the release planning activities in long term and including all

aspects of software engineering aspects in the form of a technology roadmap.

Phaal states that roadmap structure needs to be considered in parallel with the technology road mapping process [41]. That is why it is essential to make the conscious decision to focus on a certain type of roadmap fulfilling a certain purpose, which is in this case the technology roadmap should show the *high-level view of all requirements required from product side and link aspects of software engineering process from platform improvement initiatives*.

The T- Plan offers an optimal foundation to develop technology road mapping process for the software platform and is used for the theoretical basis of this thesis. Phaal and his colleagues from the University of Cambridge are the most cited, comprehensive and internationally recognized ones in the technology road mapping method. They have published numerous articles about the technology road mapping approach offering solution to customize the approach and also focusing on the implementation of TRM [44].

T- plan approach has been applied by the authors to many contexts. This emphasizes and demonstrates flexibility on one side and certain robustness on the other side. In contrast to many authors, the level of details is very high, since Phaal describes not only the micro phases of the main phases, but also elaborates on macro phases [44]. This stream of offering comprehensive literature about the customization and implementation ideally fits to the intention of this thesis.

4.3.1 T- PLAN "FAST START" APPROACH [44]

This approach aims to identify important gaps in the product and technology and establish key linkages between features of the future product and technology. The T- plan process comprises of two main parts that has been developed to support the rapid initiation of road mapping approach in the business or organization:

a. *Standard approach, for supporting product planning* [45]

b. *Customized approach, which includes guidance on the broader application of the method, incorporating many of the techniques included in the standard approach* [45]

The macro phase of the T-plan approach consists of the three phases "planning", "road mapping workshops" and "roll-out". The implementation phases of the TRM process is also embedded within the above three process which is more embedded in the second phase, the road mapping workshops. In the following section, the second phase of the T-plan is elaborated, whereas the first and last phase is handled in next section which deals with customization and implementation in detail.

In phase II of the TRM process, Phaal suggests a workshop- based approach, where analysis grids are used to identify and assess the relationships between sub layers and layers of the roadmap and in each workshop a layer of the roadmap is handled.

In the first workshop, business and market drivers are identified and prioritized by deriving from the most important performance dimension driving product development. The appropriate product feature concepts satisfying market and business drivers are identified in the second workshop. The relationships and impacts between the drivers and product features are determined using the grid after grouping the features. The technological solution realizing the product features are grouped into technical areas and their impacts on product features are sought in the third workshop. The respective grids are connected to each other as the result of the three workshops and three business areas. The actual roadmap is charted in the fourth workshop using the output the previous workshops. Key milestones are determined for each sub – layer (y – axis) along the time axis (x- axis). Paths are drawn using the results from the impact ranking and resources (e.g. technological programs; suppliers; skills etc) are identified by negotiating with the attending experts in the workshop.

Key stakeholders and experts are brought together in the workshop- form which is the main benefit of this approach. They identify strategic issues, capture, share and structure knowledge about the strategic issues and plan the way forward. The agenda of the workshop constitute the activities of each layer to be followed and TRM process is structured by workshops.

Certain management activities are important parallel to workshop procedure and this includes the facilitation of workshops, follow-up actions and process co ordination. According Phaal many benefits are derived from the road mapping process including stakeholders are brought together from different parts of the business, which provides

opportunity for sharing perspectives and information and also means for supporting communication across functional boundaries in the organization.

4.3.2 CUSTOMIZATION OF THE T- PLAN [11] [44]

Standard process for developing a technology roadmap (4.3.1) has been proposed by Phaal in the T- Plan approach. They also offer a customization of this standard process in case the purpose of the roadmap differs. In the planning phase of the T-pan's macro process customization of the roadmap takes place. The roadmap architecture and road mapping process are the two interdependent design elements to be customized in the T- Plan approach. The context (scope of interest; aims of TRM, focus for TRM; available process resources for the process) has to be defined and boundaries and aims that will affect the TRM have to be identified before considering the design of the technology road mapping process.

Having clarifying this, in macro and micro process the architecture of the roadmap are designed. The sub layers, layers and time frame entail the architecture of the roadmap (figure13). Dependent on how far the organizations want to "look" into the future and aspects to be mapped the temporal dimension is drawn in the x axis. Based on the aspects and issues the layers are determined in the roadmap. Basically there are three layers in which lower layer represents the resource side (knowledge like technologies) that is required to respond to business and market drivers, the middle layer represents the products and products features concepts that satisfies the market and business drivers and finally the upper layer represents the purpose or drivers of the road mapping activity.

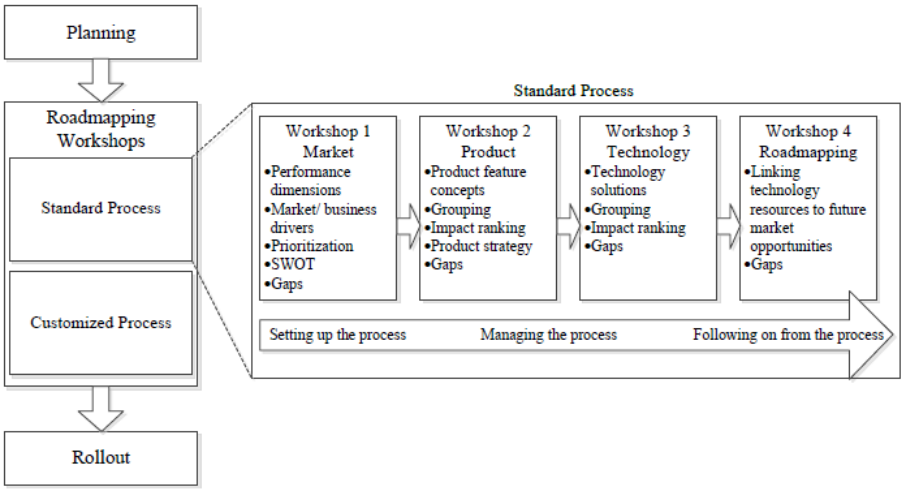


FIGURE 13 T- PLAN APPROACH [44]

The macro process is then developed, concerning wider goal that is desired to reach after the architecture of the roadmap is determined. Based on the aims to reach and layer of the roadmap the micro process can be established concerning the agenda of workshops.

The following checklist is used in T-Plan applications, which will be used as basis for design principle for road mapping process in next chapter.

Context – the aspect of the issues that started the road mapping needs to be articulated and explored, together with any challenges & constraints that will affect the approach adopted. The following items needs to be considered when designing the architecture of the roadmap.

Architecture – the structure of the roadmap, in terms of:

– *Timeframe*: the horizontal axis of the roadmap represents the chronological aspects, in terms of key milestones and planning horizon and can also include the past activities and events should be included.

– *Layers*: the broad layers and sub-layers closely related to business is viewed conceptually and structured physically in the vertical axis of the roadmap.

Process – the activities needed to build the roadmap content, maintain the roadmap, agree and identify actions and make decisions are staged. This further includes a "macro' & "micro "level which is associated with particular agenda that will guide the workshops in the short term and broad steps needed in the medium and long term as well.

Participants – the stakeholders that need to be involved in the workshops and process to develop a credible and well founded roadmap. Typically

people from multifunctional team representing both technical and commercial perspectives are needed during this process.

Workshop venue and scheduling – a suitable venue is arranged for the workshops to allow the participatory road mapping activity by the groups.

Information sources – all the available information for the road mapping activity should be taken into account as there is practical limit on quantity of the data that can be accommodated in the workshop environment. Relevant information from the necessary documents should be captured & assessed before the workshop and consideration should be given to what information should be incorporated after the workshop in the context of an ongoing road mapping process. The importance should be given to the information supplied to participants prior to the workshop and built in the roadmap architecture.

Preparatory work – actions needs to be considered prior to the workshops need to be agreed and identified by inviting participants, preparing briefing documents, booking workshop venues and facilitation materials .

4.3.3 IMPLEMENTATION PLAN

Robust maintenance plan including the implementation plan should be proposed based on the organization context and needs, once the roadmap is finalized [48]. One of the main challenge is the maintaining the road mapping process" alive" on long term in spite of have many methods and techniques. Phaal indicates that only 10% of companies have successfully applied the technology road mapping approach from a survey of 2000 UK manufacturing firms. The main challenges identified by survey respondents included developing a robust TRM process (20%), keeping the road mapping process "alive" on an ongoing basis (50%) and starting up the TRM process (30%). [38]

For easy dissemination of information, retrieval and maintenance, the main recommendation from the literature is that it should be digitally hosted [9]. For the easy implementation of the roadmap it should be integrated with normal work process followed within the organization. The roadmap should be further updated on a periodic basis, at least once a year or perhaps linked to strategy or budget cycles in practice.

4.4 ROAD MAPPING IN SOFTWARE PRODUCT FIELD

Now, we know the design principles of the customization process and chosen T-Plan approach can act as the baseline for developing technology roadmap for the software platforms. In this section, we examine the available few literature about road mapping experience in software product companies. In one of the literature, experience of product road mapping in small software company for maintaining its focus on the right issues and keeping long term goals clear were reported by Vahanitty [47]. However, application of the technology road mapping approach in the software platform context is rather new and less investigated.

Pertaining to the software engineering process, Vahanitty defines road mapping as method for defining and planning software product requirements based on the needs of the market and stakeholders [47]. The process of identifying stakeholder needs and documenting them in a form that is adequate for communication and analysis and process of discovering a system's purpose is defined as software systems requirements engineering (RE) [49].

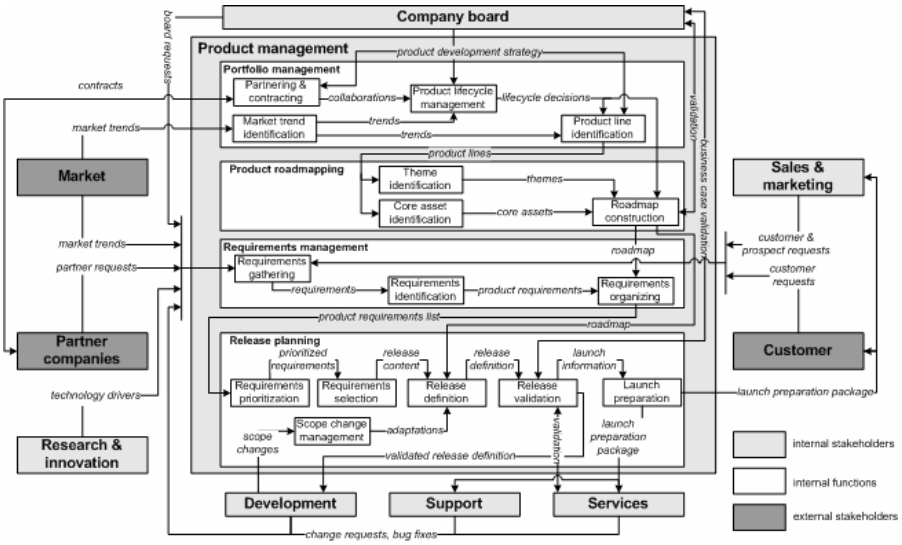


FIGURE 14 REFERENCE FRAMEWORK FOR SOFTWARE PRODUCT MANAGEMENT [12]

According to Leholta, product road mapping is also closely linked to RE since it deals with different releases of each product containing functional features. In a product roadmap the product is represented as product releases containing several functional product features [16]. Wiegers states that product functional features are a set of logically

related requirements that provide a capability to the user and enable the satisfaction of a business objective [49].

Also, Lehtola, have stated that targets for the product road mapping and release road mapping are same by the product road mapping is more high level. The main objective of product road mapping is to manage situations where same technical choices are included in several products and help product manger to create and maintain release roadmaps and release road mapping is to inform stakeholders about scheduled future releases. Furthermore, product road mapping can be seen as a part of broader perspective to software product management [47]. According to van de weerd "*reference framework (figure 14) for software product management consists of portfolio management, product road mapping, requirements management, and release planning*" [12]. According to the reference framework, from portfolio management product road mapping receives input regarding product lines.

- This will assist us in positioning and implementing the roadmap in development of software platform product in product portfolio planning (2.3) in Shell case.
- Also, based on the reference framework, the developed roadmap should assist the product managers deciding the requirements for releases and planning activities which in our thesis will be used as validation part (6.3) for the developed roadmap by evaluating in the one of the release activity of the SSP platform.

4.5 Recommendations Summary

Based on the literature overview on TRM approaches, we derive following recommendations to integrate in the customization process for developing technology roadmap. Therefore, it is important to give a clear summarization of the recommendations, in order to simplify the process of the roadmap development. Table 3 recaps the major findings of the chapter. Later, it will be shown how the roadmap addresses every advice from the table in its design approach.

TABLE 3 RECOMMENDATIONS FROM LITERATURE

Stages	Recommendations from Literature
Perspectives of Roadmap	- Roadmap should address both *market pull and technology push perspectives* - Road mapping activity should address both *exploratory and goal oriented*
Design Principles	- *Visualization* is a key aspect of effective road mapping, both in terms of the elicitation of information from groups (workshops), and also analysis and representation for communication purposes. - *Time* is a fundamental dimension that must be represented - *One-page views* are recommended wherever possible, in terms of visualizations and associated summary text. This supports communication and ensures that the key issues are focused on, set against the context provided by the 'big picture' view provided by the roadmap structure. - *Integration* with core business processes is critical if the road mapping is to have an impact on decisions and budgets. - *Multiple stakeholder perspectives* are always important for strategy and innovation roadmaps, internally and externally, with the commercial–technical interface of particular importance. - *Customization* is typically needed to adapt the roadmap structure and process to suit the particular application.
Customization Checklist	- The *context and scope* must also be clearly understood - *Architecture.* The roadmap 'architecture' (structure) should be adapted to suit the issue being addressed. - *Process.* The road mapping process needs to be adapted to suit the issue being addressed. - *Alignment.* Integration aspects should be considered, in terms of how road mapping links to other business processes, other management tools - *Preparation.* Identify available information, and any constraints or assumptions that need to be made, together with preparation required for workshops.
Maintenance	- *Digital Storage* of the roadmap for easy sharing, retrieval and communication - *Robust Implementation Plan*

4.6 SUMMARY

This chapter provided an overview on approaches related to technology road mapping. There is a little attention to technology road mapping for software product platforms in academia papers. They are more concerned about the general TRM process, different formats, purposes and uses for the manufacturing sectors. We see the need to effectively utilize the general TRM, format & strategies for the software product platforms in the later technology roadmap development.

We did not come across any TRM approaches specifically for software product platforms that would provide clear steps for an organization to develop their road mapping strategies. The main reason could be as we said earlier, the very specific nature of road mapping process for every specific nature and to customize technology roadmaps. However, we found that T-Plan fast approach by Phaal as an optimal foundation to customize the TRM. The design principles and checklist provided by Phaal to customize the road mapping process approach will be used in this thesis to customize the roadmap for the software platform.

At this moment we have the second half of information required for the design science research - the knowledge base obtained from the literature to assure rigor of the study. The next step is to integrate this information in the development of roadmap and this idea will be discussed in the next chapter dedicated to the design of the technology roadmap and process.

5

Technology Roadmap for Software Platform (SSP – SHELL Case)

In Chapter 2, we determined the organizational challenges and major elements influencing the development of software platforms. In Chapter 3, we discovered the major business need for the Shell SMART foundation platform when it comes to development of technology roadmaps. In Chapter 4, we gathered recommendations from the TRM approaches for the development technology roadmap and customization process itself. The goal now is to synthesize advice from the literature with the business requirements and to develop a technology roadmap that integrates all key findings and supports the release planning strategy. This technology roadmap will serve as an effective tool to support the decision making around requirements prioritization and help in planning the series of the software platform releases strategically.

Section 5.1 describe the methodology used while design the technology roadmap for software platform and synthesis of input from business needs and literatures. Section 5.2 describes the main guiding principle behind the technology roadmap followed by the roadmap development process and technology roadmap for the software platforms pertaining to Shell Case (SSP Platform) in section 5.3. The detailed discussion on how the developed technology roadmap incorporates the previous findings/address the challenges is given at section 5.4

5.1 Methodology

The expectation for the methodology is that it should enable the development of technology roadmap and process for developing software platforms internally within the organizations. From the previous chapter, we found that they are many approaches available, but only a limited number of approaches make sense given the boundary conditions. The schedule of the project and set-up at Shell, the requirements to have theory and business requirements incorporated are such conditions, as well as the limited time and resources available for the master thesis.

The case-based action research methodology is one of the viable approaches meeting these conditions applied by Phaal to develop and customize dozens of roadmap process in their research program [37]. Since action research covers a large variety of approaches, and the exact methodology of Phaal is not disclosed another approach qualifies to fulfil the expectations. This approach is Hevner's practical problem solving circle which is very similar to case based action research, and has three main advantages [13]. First, a strong theoretical orientation and understanding of business requirements before the actual design serves as strong foundation. Second, it is suitable in the context of change process, since the development of TRM process is considered in parallel with constant feedback and iteration with the involvement of stakeholders. Thirdly, it can be easily integrated in the already existing project schedule of Shell and is feasible given the available resources not only for Shell, but also for this master thesis.

In the first step, Hevener states that it is important to choose a correct theoretical approach for the design. With reference to the previous chapter 4, the TRM approaches serves as the theoretical basis for the technology roadmap to be developed. The best option available for the customization of TRM and development of technology roadmap is explained in the literature review. With respect to the design of the TRM, a combination of Phaal recommendations and current business needs from the problem environment (Shell Case) is used. This both combination ensures the research and rigor relevance of Hevener research methodology principles. Now the main task is to customize the T-plan to the specific requirements & major elements identified in the literature about the software platform and business needs with respect to Shell case.

In the second step, according to Hevener, the TRM (T- Plan) approach described in literature is evaluated, improved and customized to Shell needs. It is important to state that the actual design of technology roadmap was considered in parallel with the changes and inputs coming from the stakeholders. To cope up with changing needs constant feedback and iterations were done during the development of the structure/architecture of the technology roadmap by stakeholder workshops and regular meetings. The technology roadmap for the software platform is designed utilizing the available means while satisfying the business needs in the problem environment in accordance with the research guideline stated by Hevner. These changes were

captured during the *stakeholder workshops* (mentioned in section *5.3.2.3*) and during one to one meetings.

During this meeting and workshops, the approach and ongoing design process of technology roadmap was discussed and presented. The group of stakeholders, involved in this process is consistent with the stakeholder groups described in Fig 11. The stakeholder group has the competence to discuss the major elements influencing the development of software platform and effects of technology roadmap from different perspectives and to bring valuable input from their areas. Comments and remarks from the stakeholders were documented and incorporated in the actual design of the technology roadmap. The data for the technology roadmap were gathered from internal documents related to SSP platform project including improvement plans, requirements management plan, reference architecture documents, technology maturation plan documents, solution architecture documents, and project documents.

5.2 The Technology roadmap Prerequisites

As mentioned earlier in chapter 4, every technology roadmap has its own specific traits and needs to be customized based on the purpose and context of the organization in subject. Accordingly, every organization has its own approach in developing roadmaps [37]. With our roadmap we set several prerequisites to increase its overall utility and make it sufficiently generic to implement in a wide range of software platforms in different organizations. These prerequisites are derived from both literatures and business needs:

- Provide enough structure to include all aspects of software engineering choices.
- Give enough flexibility for organization to customize the roadmap according to their operational settings
- Make the road mapping approach easily integrated into normal/existing process followed in the development of software platforms in the organization.
- Incorporate road mapping design principles found in the literature.
- Incorporate organizational business needs discovered during the interviews.
- Make the roadmap applicable for all kinds of software products but also take the specifics of software platforms and release planning activities into account.

We consider and customize the technology road map and process for the specific needs in the scope of this research. Our goal is to develop the technology roadmap and TRM process tailored for the software platform (SSP Product) for Shell case that will later allow the creation of generic technology roadmap and TRM process for any software platform development in any organization.

5.3 THE TECHNOLOGY ROADMAP FRAMEWORK

5.3.1 DESIGN APPROACH

In Chapter 1 we described the main idea behind the technology roadmap for the software product platforms. Being a step-by-step guide, it should define a set of procedures which, when followed, will make it possible to customize the road mapping approach and outline including the definitions and context as well as find the best solution to support the strategic release planning activity. The literature review and business needs of the interviewed company helped us to create a customization of the road mapping process for software platforms (SSP) specific to Shell Case. Below we summarize the major road mapping process mentioned by the two sources:

- The TRM process is tailored to the needs of the software platforms consisting of seven stages: *purpose, visualization format, architecture of the roadmap, stakeholder mapping, features, storage and maintenance.*
- The TRM process starts with the business need and top management support in the organization. In order to gain the top management support the *purpose and boundaries* of the technology roadmap has to be clearly defined.
- To ensure the consistency in the taxonomy of the requirements and activities followed in platform development, the *visualization format* of the roadmap is chosen to provide supplementary information, key decision points, gaps and critical paths including linkages between different platform components.
- After the purpose and format, *architecture of the roadmap* based on the major elements of the software platform is outlined.
- *Stakeholders* responsible for different elements of the software platform are mapped in the roadmap to share information, perspectives and to make decisions.

- Then*, features* are captured, analysed & prioritized for the outlines present in the roadmap architecture through stakeholder workshops and stored electronically for easy sharing and communication of the roadmap.
- Finally, in the *maintenance,* the roadmap is validated, updated and analysed during the every release of the platform to maintain the quality the data and up to date information following the normal organizational process.

The TRM is a starting point in the technology roadmap design, as it allows us to map out the main activities and their order with respect to the major elements of the software platform product.

We want to mark out to the possible extent, the common steps of the road mapping process and use them as a basis for developing the technology roadmap for the software platforms. When a software platform release happens, the organization can modify the roadmap to reflect the changes by following the procedures proposed in the maintenance plan.

5.3.2 ROAD MAPPING PROCESS

In this research we use a flow chart diagram to illustrate the customization and development process of the technology roadmap performed at the SSP Platform (Shell Case). Our goal is to introduce a set of activities to arrive on an optimal technology roadmap for the software product platforms, which correlates with the flowchart idea to present a step by step solution to a given problem. The flow chart shows how to customize the road mapping process and arrive on the technology roadmap for the software platforms. As mentioned earlier, customization process needs to be considered in parallel with the development roadmap and hence, we used real examples from Shell SSP platform case in explaining the customization procedures.

It follows the common steps introduced earlier:
Step 1: The TRM process begins with the determination of purpose, choosing formats and defining the architecture (structure) of the roadmap. This thesis mainly focuses on the architecture (structure) of the roadmap for the software platforms that must be configured to suit the focus and scope of the issue being addressed, and to provide a technology roadmap that covers all the aspects of the software engineering aspects.

Step 2: the people that need to be involved in the process and workshop, with the knowledge and expertise necessary to develop a well-founded and credible roadmap are mapped to the corresponding layers of the roadmap.
Step 3: workshops are conducted with the relevant stakeholder (Figure 11) to gather content/features for the each layer in the roadmap; the content/features are prioritized and validated and stored electronically for easy retrieval.
Step 4: the roadmap is validated, and updated along with the normal process after every release of the platform.

In this thesis, we do not specify who exactly within an organization performs the activities described in the flowchart (figure 15); we assume though, that these people either belong to the development of the software platforms from the beginning, or join the team upon request.

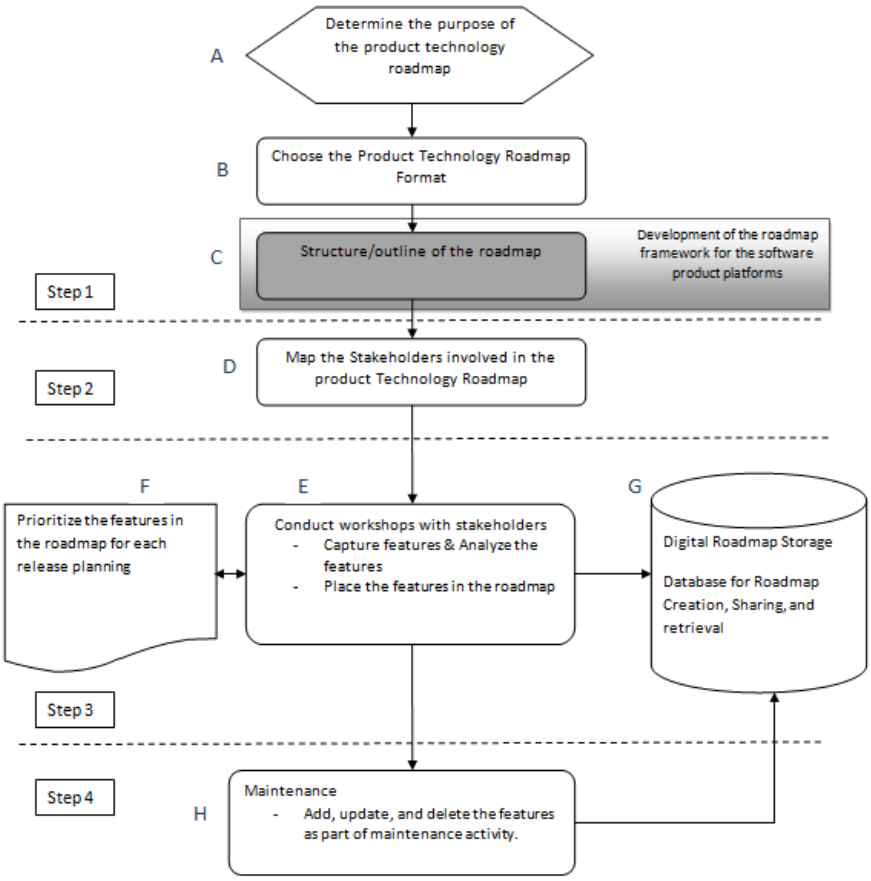


FIGURE 15 ROAD MAPPING PROCESS

The next section will describe every step in the flow chart, by giving an explanation on how every customization process works, together with the final technology roadmap for the software platforms (SSP) and overall justification of their presence in the flow chart.

5.3.2.1 First Step: Customizing the Architecture of the Roadmap

As a first step, the customization of the technology roadmap specific to the SSP platform (Shell Case) was considered. As a first step in the customization procedure, it is necessary to determine the context & format of the roadmap (A &B). Once the context & format is determined, architecture/structure of the roadmap has to be defined according to the business needs and challenges, in order to initiate the road mapping process (C)

A: Context of the product Technology Roadmap

The TRM process starts with the business need and management support in the organization. In order to gain the management support

the context of the technology roadmap has to be clearly defined. The nature of the issue that triggered interest in road mapping needs to be articulated and explored, together with any constraints that will affect the approach adopted, including the following considerations: -

Scope: The scope of the technology roadmap for the SSP platform is to look forward and provide the overview of all requirements influencing the platform from all major elements of the platform including the aspects of the software engineering choices.

Focus: The technology roadmap should assist in decision making while choosing the content/features during the strategic release planning activities.

Aims: The aim of the technology roadmap for SSP platform is that, it should be used as baseline document for the scope management for release planning activities.

B: Format of the product Technology Roadmap

To cover all the major elements of the software platform & to ensure consistency and to show the implicit linkages between the components "Layer" format is chosen as the standard format for developing technology roadmap forward. This type of format will also communicate the feature, plans and decisions moments intuitively.

- *All the major elements of the software platform including applications, features, technology capabilities, architecture strategies, operational requirements & process improvements are placed in terms of broad layers and sub layers of the vertical axis in more physical and conceptual way. (see figure 16)*
- *The chronological aspects, in terms of planning, supplementary information, key decision points, gaps and critical paths are included in the horizontal axis.*

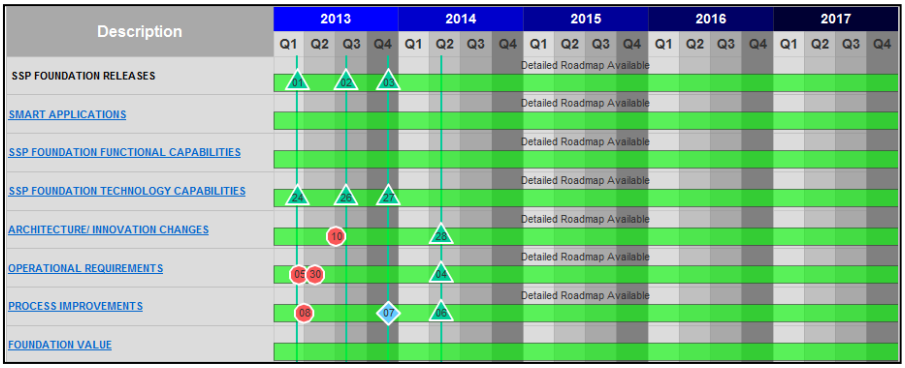


FIGURE 16 LAYER TYPE ROADMAP

The layer format allows the evolution of sub layers which can be explored, together with the interlayer dependencies, facilitating the integration of the functional feature with technology choices and applications. Also, this has the advantage of unifying and simplifying the required outputs which facilitates integration of different roadmaps and communication to all stakeholders from the development of software to support through road mapping.

As discussed in (chapter 3), multiple group of stakeholders are involved in the development of the SSP platform. These stakeholders need different level of content and information from the roadmap (not all the stakeholder will expect the same level of details in the roadmap). To cater their expectations, we decided to use two different types of product technology roadmap format.

- *Bull's eye View – Provides holistic overview of major elements captured in each layer of the roadmap. This format (figure 17) clarifies the area of value, opportunity, and gaps across the planning and evolution of the SSP platform. This format is further used to explore the future possibilities of the and also to capture the content for roadmap during stakeholder workshops. Program managers and senior management requires high level detail and bulls' eye view format is apt to provide them with overview of the evolution of the platform. The full Bulls eye view is attached in the appendix A.4.*

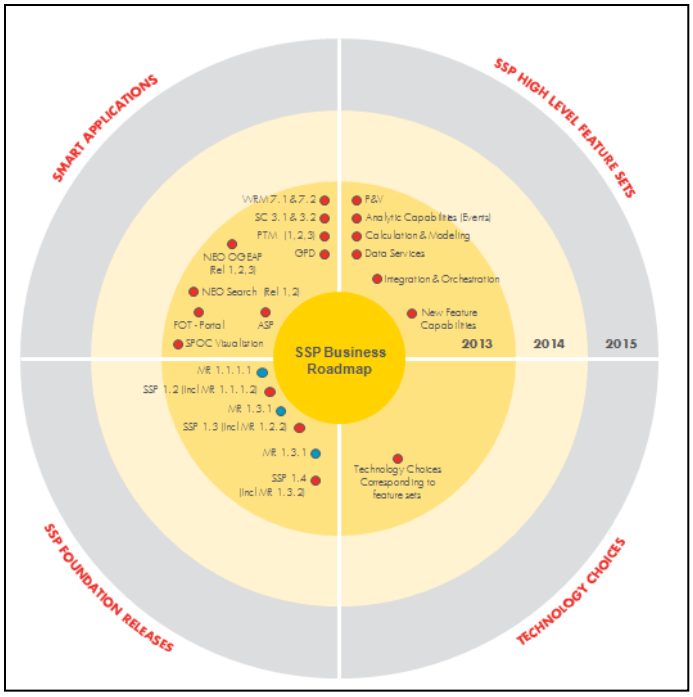


FIGURE 17 BULLS EYE FORMAT

- *Gantt chart view – This format provides next level of detail for the contents shown in the bull's eye format. In this view, stakeholder can know the timings for the release of the features, capabilities, improvement plans and prioritization value. The linkages, relationships and dependencies between the features are shown. This format (figure 18) is much useful for the product managers of the application product teams and project managers of the individual releases.*

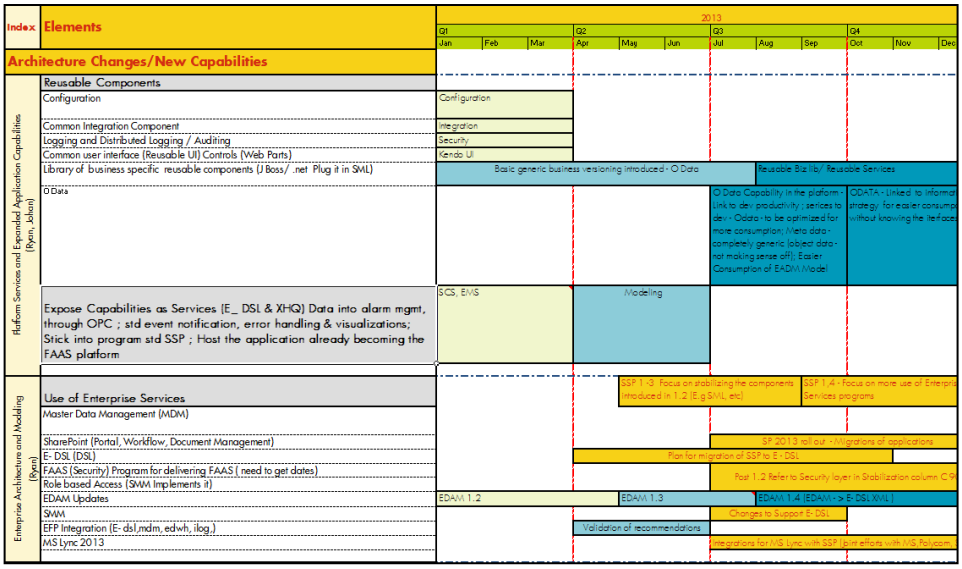


FIGURE 18 GANTT CHART FORMAT

C: Structure of the Roadmap for the software platforms

In this step, architecture of the roadmap (figure 19) is structured and major elements of the platform are mapped for each layer in the technology roadmap. In this structure we include the perspectives from literature both *market pull and technology push* to really distinguish the requirements coming the SMART applications and platform improvement initiatives. We mapped:

- SMART Applications, SSP Functional Capabilities, SSP Technology Capabilities in *Market pull perspective.*
- Architecture Changes/Innovations, Operational Requirements, Operational improvements in *Technology push perspective.*
  (For Shell specific reasons, we call Market pull as business driven and Technology push as Process improvements)

These six main vertical layers of the technology roadmap cover all the major elements of the software platform including all aspects of the software engineering choices. From the observation of Shell SSP platform case while performing this research, designing a technology roadmap with market pull (business driven) provides relevant knowledge for identifying future application and therefore, defining which functional capabilities and technology choices should be developed. Including the technology push perspective (process improvements initiatives) focusing on software engineering choices contributes more significantly for the development of

SSP platform than incorporating new technologies into the SSP Platform.

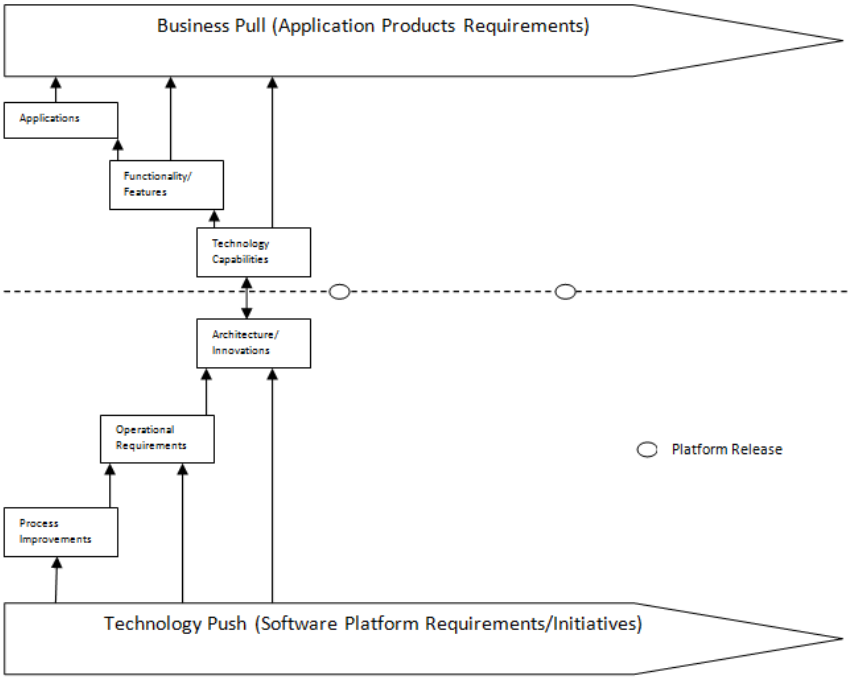


FIGURE 19 ARCHITECTURE OF THE TECHNOLOGY ROADMAP

Market Pull (Business Driven Perspective):

Applications – The main stakeholder of the SSP platform; the details about the SMART applications are captured in this layer (figure 20). The details include the release goal (schedules) and high level features/functionalities required for each of their releases.

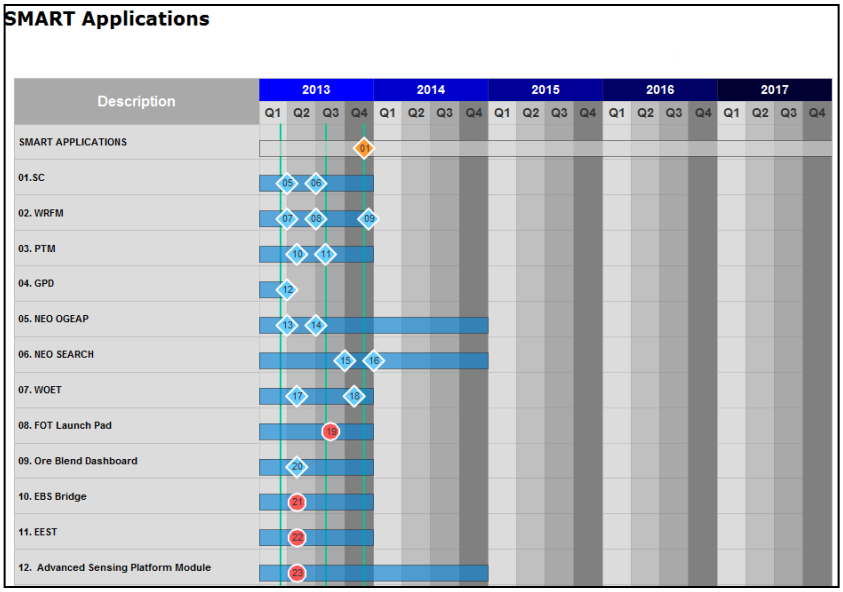


FIGURE 20 APPLICATION LAYER

Platform Functional Capabilities – SMART application's requirements are translated and grouped in to high level functionalities (or) features offered by the SSP platform which are visualization, calculation engines, events management, predictive analytics & data services. The details on enhancements, upgrades on these functionalities from the SSP platform are also captured in this layer. Constructing this layer (figure 21) might be a complex process since SMART application product mangers may be unable to conceive and communicate what they want or need from the platform, the difficulty in managing requirements and continuous forces that act towards requirement changes.

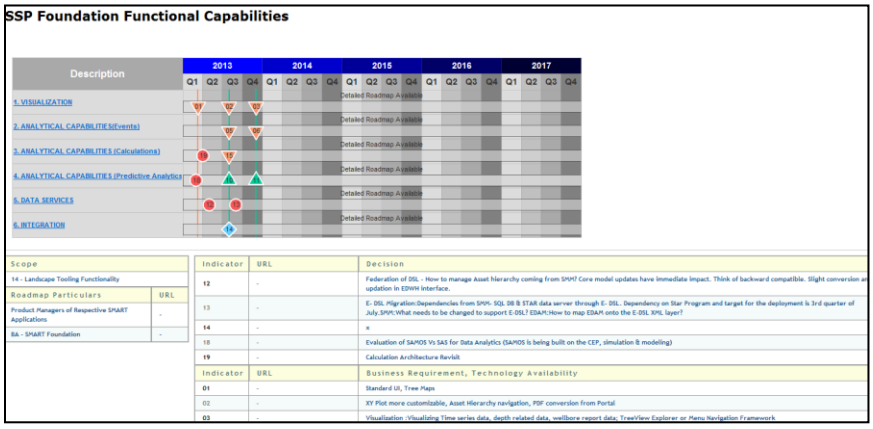


FIGURE 21 FUNCTIONAL FEATURE LAYER

Platform Technology Capabilities - Technology capability enabling the functionality/features are captured in this layer of the roadmap with the relationships and linkages. For example visualization functionality is enabled by *SharePoint* and *Telerik kendo UI* products. For each functionality corresponding technologies are mapped and their product updates, enhancements decommission plans are captured in this layer (figure 22)

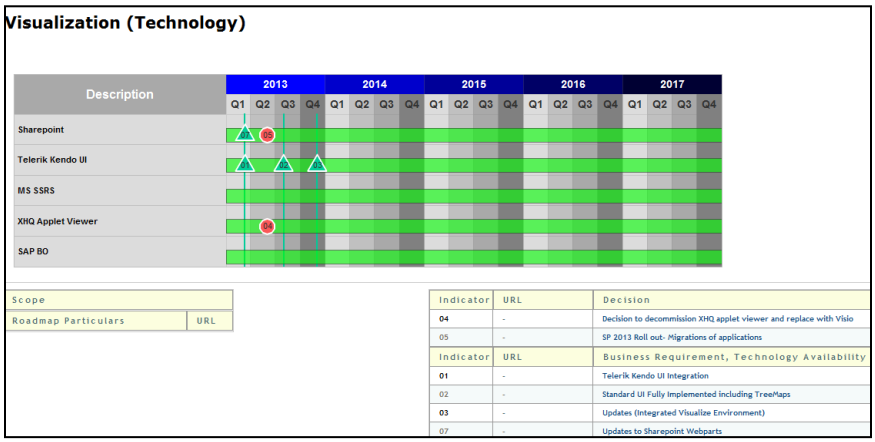


FIGURE 22 TECHNOLOGY CAPABILITY LAYER

Technology Push Perspectives (Process Improvements):

Architecture Changes/ Innovations – The architecture strategy and components of the SSP platform are captured in this layer. This includes the architecture vision of moving towards service based architecture, expanding platform services and application capability including reusable components, use of enterprise services, security, catering the needs of global deployments and including potential new technology capabilities in support of general vision of the platform and organization. Furthermore, target architecture of the platform, parameters required for the platform stabilization including performance, security and infrastructure roadmaps and details about the architectural partitioning are also captured in this layer (figure 23).

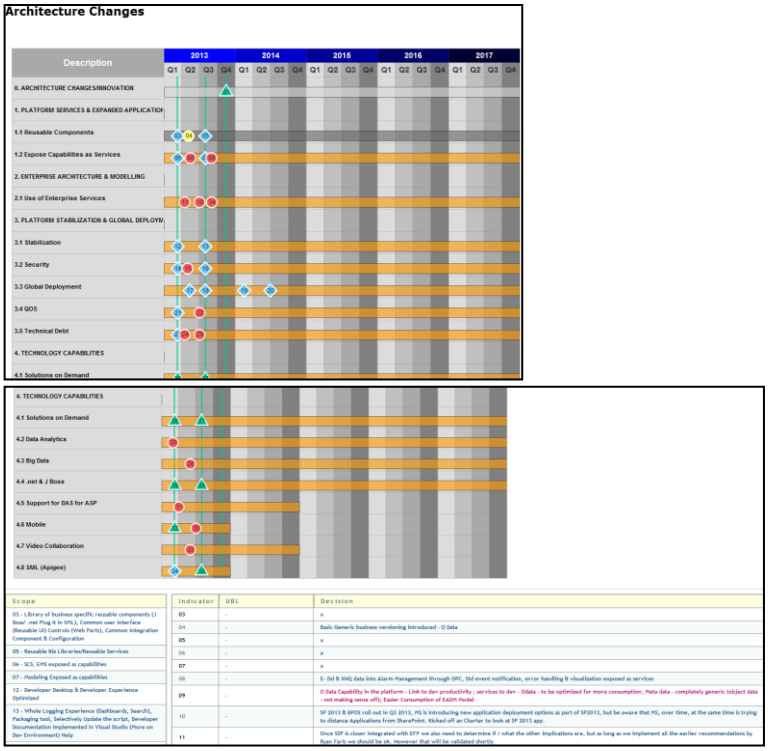


FIGURE 23 ARCHITECTURE CHANGES LAYER

Operational & Support Requirements - This layer (figure 24) captures the support process and operational requirements to stabilize and expand the application flexibility to make the SSP platform robust maintainable. The operational requirement helps the platform product to reduce the downtime of the platform and supports the faster deployment or release cycles. Furthermore the support model addresses how product teams get assistance as they work to incorporate SSP platform components into their SMART Application. In this case, ITIL support process checklist is incorporated in this layer to focus and align the platform IT services with the needs of the SMART application teams. (ITIL

Support process was chosen due to specific need of the case).

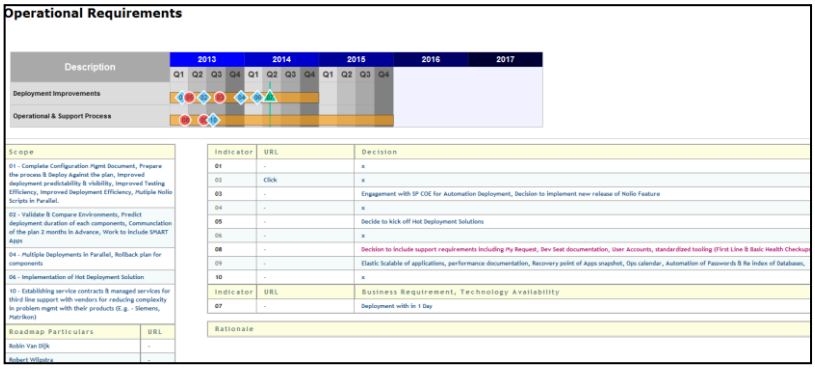


FIGURE 24 OPERATIONAL REQUIREMENTS

Software Process Improvements - The technology roadmap for the SSP platform should cover all the software engineering process/aspects involved in developing the platform, not only the software component of it. This layer (figure 25) captures the contents required for developing high quality and reliable SSP software platform. The quality assurance process (test management improvements), development methodologies and lessons learnt from the earlier releases are captured in improvements plan to make the platform development process robust.

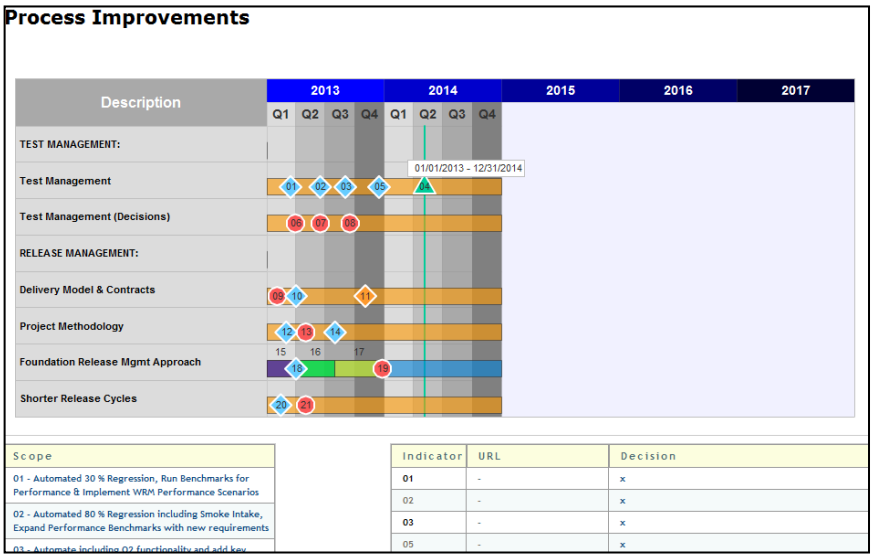


FIGURE 25 PROCESS IMPROVEMENTS

In this stage the architecture of the roadmap was customized for the need of the SSP platform which is the important step in the road mapping process itself.

5.3.2.2 Step 2: Stakeholder Mapping (D):

The stakeholders corresponding to all major six layers were identified and mapped in the roadmap. These stakeholders are responsible for providing more content and features, responsible for participation in the workshops and maintenance of the roadmap ongoing basis. A small road mapping team with the stakeholders contributing to each layer were formed. We identified three types of roles that need to be present in a road mapping team:

Owner – The roadmap for the SSP platform was owned by the platform manger who in future would be responsible for

- Controls the change management of the roadmap.
- Maintenance of the roadmaps, organizing workshops to initiate appropriate updating actions to ensure integral involvement of the contributors and input from the different functions.

Contributor - The contributors are typically the product/project managers involved in providing the content for the elements in each layer of the roadmap. In SSP platform roadmap typically contributors consists of:

Elements from the Roadmap	Key Contributors
Applications	Application Product Managers
Platform functionality/features	Business analyst of the platform & Application product Managers
Technology Capabilities	Architects from platform and application
Architecture Changes/Innovations	Architects from platform
Operational Requirements	Operations Manager and Support Staff
Process Improvements	Managers

Viewers – The viewer are the stakeholders who are influenced by the platform product releases in the organizations. Typically viewer group changes periodically based on their dependencies on the platform. The owner of the roadmap controls the distribution of the roadmap to the viewer group on the basis of their demand and necessities.

5.3.2.3 Step 3: Contents Gathering, feature prioritization and Storage

(E) Workshops:

To gather the content/features for all the six major layers in the SSP technology roadmap, we conducted six stakeholder workshops (figure 26) with the identified stakeholders. During this workshops both exploratory & goal oriented perspectives of information were gathered. Before the workshop, targets for each layer were gathered from various documents including architecture documents, requirements management plans, improvement plans, gap assessment documents. The bull's eye view formats were used to steer the discussion for gathering the exploratory views. The features captured were analysed to identify the pre requisites, dependencies and linkages among them. The features were then recorded in the roadmap with scope, decision moments and use cases. The important *finding* in this stage was that we were able to

"Bring together a range of expertise, structuring and sharing of knowledge together with simulation and brainstorming participation"

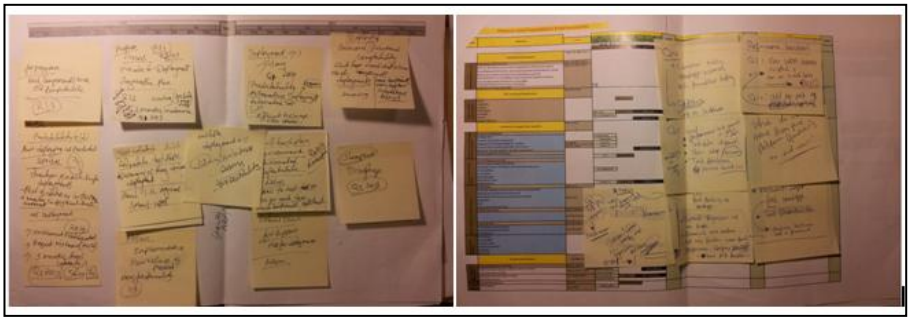


FIGURE 26 WORKSHOPS

(F) Prioritizing Features:

After the content/features were gathered from stakeholder workshops, we prioritized the features/content for all major six layers in the roadmap. This stage was difficult and time consuming since many features/content/requirements were related to each other and also competing for the same resources and budget during the release planning activity. Many methods & techniques were discussed in the literatures for the prioritization of the features. We used stakeholder voting method and Cost benefit analysis survey for the prioritizing the features for the release planning of the SSP platform releases.

- *Stakeholder Voting:* Stakeholders were given opportunity to vote on all the features from the entire layer in the roadmap according

to their preferences and business needs. The stakeholders included SMART application product managers, Platform managers, Business Analysts and Architects. The result of this exercise resulted in set of priorities assigned to feature groups according to stakeholder preferences.

- *Cost/Value Analysis:* The costs were estimated to the feature groups in the roadmap by the technical person or the architect of the platform. Based on the budget allocation (from investment proposal) for each releases, combined with the priorities from the stakeholder voting method the features were prioritized in the roadmap. After analysing and prioritizing the feature groups, the content of the features were placed in the SSP technology roadmap.

(G) Digital Roadmap Storage:

Keeping the roadmap alive is one of the challenges discussed in the chapter (4.3.3). Also, SSP platform roadmap should be updated frequently to handle the continuous inflow of the features from different stakeholders. The SSP platform roadmap was hosted electronically in SharePoint – organization database for the easy creation, updates and retrieval. It also provided, standard graphical user interfaced for easier communication of the roadmap content (decision points, milestones) to large group of stakeholders. Moreover, if hosted electronically (figure 27) it's easier to track the changes or the version history and easy to integrate with other organization roadmaps if necessary during the platform evolution.

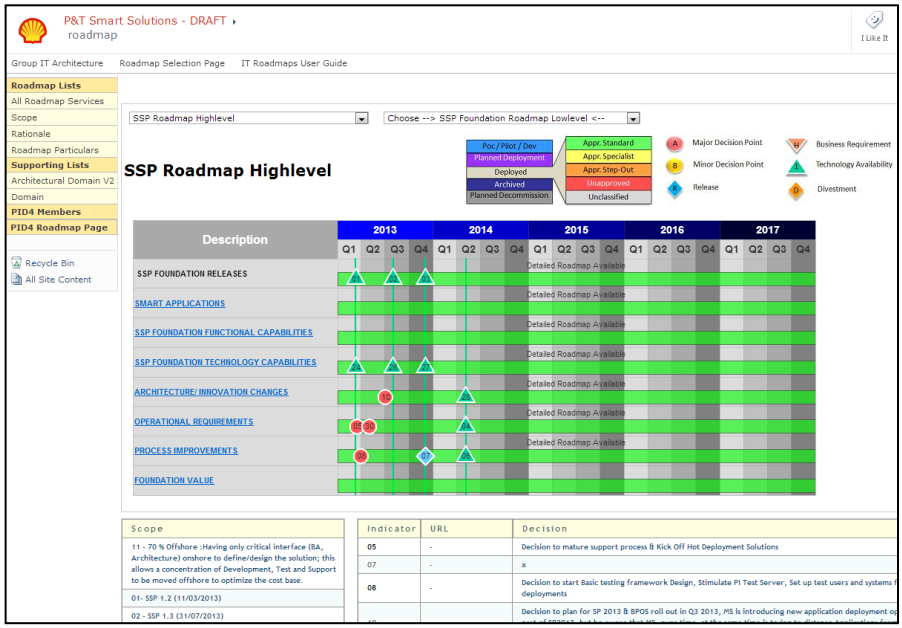


FIGURE 27 HOSTED IN SHAREPOINT (DIGITAL ROADMAP STORAGE)

5.3.2.4 Step 4: Maintenance of the Roadmap (H):

Since the SSP Technology roadmap is central tool for communication, the roadmap has to be constantly updated & validated to maintain the quality data and version controlled. The maintenance of the roadmap phase includes the change management of the roadmap and could be supported easily with building a roadmap digitally (step h) since it is a fast and simple technique and it allows the contributors/owners of the each portion of the roadmap to maintain control of edits and changes. In software product platform roadmap the maintenance of the roadmap involves owner and contributors as detailed in (Step D).

The primary purpose of the SSP roadmap as mentioned in Step (1) is to show the key features of each releases and expectations on the demand of requirements impacting the product platforms for each release. Hence, we recommend the maintenance of the roadmap should be linked to the release cycles and standard work process of the software platform product team. The priority of the features in the roadmap, changes for each SSP platform release depending on the SMART applications business demands. The re- prioritization of the content in the roadmap for

the maintenance is linked with normal requirement management process of the SSP platform. Further, the priorities are reformed in different intervals with the road mapping team stakeholders.

The SSP platform roadmap is built to ensure minimum maintenance process is involved for keeping the roadmap updated. The below table 4 represents the maintenance intervals for each layer of the roadmap. To ensure consistency in the road mapping process all the stakeholders involved in the road mapping team should meet face to face twice a year to keep the roadmap updated with evolution of the platform product.

TABLE 4 MAINTENANCE INTERVALS

Elements	Update Intervals	Stakeholders involved	Maintenance Method	Source of the Documents
Application Product Layer	Once in beginning of the Year	Program & Portfolio Managers	Workshop Based	Investment Proposals
Feature Functionality Layer	Three times a year (After every platform release)	Business Analysts, Product Managers of the Applications	After every release of the platform product.	Gap Assessment Documents, Architecture Documents
Technology Capabilities/ Innovation	Three times a year (After every platform release)	Architects	After every release of the platform product.	Architecture Documents
Architecture changes	Three times a year (After every platform release)	Architects	After every release of the platform product.	
Operational Requirements & Support process	Three times a year (After every platform release)	Operational & Support Staff, Platform Product Manager	Workshop Based	Improvement plans/Lessons Learnt Documents
Process Improvements	Three times a year (After every platform release)	Development and Test Manager, Platform product Manager	Workshop Based	Improvement Plans/Lessons Learnt Documents

5.4 THE ROADMAP AS AN INTEGRATIVE TOOL OF PREVIOUS FINDINGS

As we mentioned in Section 5.2, one of our main prerequisites is to incorporate advice found in the literature, and also to address organizational business needs regarding technology roadmaps discovered through the interviews. Table 5 shows where in the framework we implemented the collected recommendations and organizational requirements.

TABLE 5 IMPLEMENTATION OF FINDINGS IN THE ROADMAP

	What to Integrate	*Where in the Technology Roadmap*
Literature	- *Roadmap should include all the four perspectives (Market Pull, Technology Push, Exploratory & Goal Oriented)*	*Structure/Outline of the roadmap ; Workshop*
	- *One-page views of Visualization*	*Hosted in SharePoint for one page views visualization.*
	- *Integration with core business processes*	*Maintenance of the roadmap linked with release cycle and standard working process*
	- *The roadmap should be easily updated, created, stored and retrieved*	*Roadmap is hosted electronically*
	- *Multiple stakeholder perspectives*	*Workshop Method*
	- *Digital Storage of the roadmap for easy sharing, retrieval and communication* - *Robust Implementation Plan*	*Hosted SSP Technology Roadmap in Sharpoint*
	- *Right abstraction (level) of information should be communicated to stakeholders*	*Graphical format of the roadmap and Layers Type of the roadmap.*

Business Needs	- *To support the release planning and requirements management process*	*SSP Technology Roadmap*
	- *Manage and communicate the delivery of functionalities*	*Six Major Layers of the SSP Technology Roadmap*
	- *Better oversight of known requirements from all the elements influencing platform release to avoid last minute fire drills in the requirements selection process.* - *Visibility on functional features delivery beyond the current releases from the foundation platform* - *Identify general functional capabilities for the platform based on future applications.*	*Six Major Layer of the SSP Technology Roadmap* *Functional Feature Layer(SSP Technology Roadmap)* *Workshop Method (Exploratory Perspective)*
	- *Incorporation of technology into midterm plan* - *Evolution of the architecture and its content has to be documented and should be readily available for reference and communication*	*Architecture Changes/Innovation (Structure/Outline of the roadmap)*
	- *Improve the operational and support process around the platform by reducing the downtime, increasing the availability and maintainability of the platform* - *Step up the support process and make it visible for all the application teams*	*Operational requirements (Structure/Outline of the roadmap)*

- *Platform needs to enhance the release and deployment process by improving development & testing strategies.*	*Process Improvements (Structure/Outline of the roadmap)*
- *Gather ideas and initiatives to improve the software engineering process around the SMART platform*	*Workshop Based (Exploratory Perspectives)*
- *Document the lessons learnt from previous releases & gather improvement plans from stakeholders*	*Process Improvements (Structure/Outline of the Roadmap)*
- *Communication model in form of visual graphical format.*	*Bull's Eye, Gantt Chart, Layer Type Format*

5.5 SUMMARY

This chapter presented a technology roadmap and customization process for the SSP platform which is the software platform. It comprises a road mapping process flowchart that built upon four processes that we discovered to remain relatively constant with T-Plan design principles and checklist. These processes are: 1) Customization of the roadmap to suit business needs, 2) stakeholder mapping, 3) Workshop, gather & analyse features 4) maintenance & storage.

The major advantage of the developed roadmap is that it clearly integrates all the major elements/ characteristics of the software platform. Currently, no technology roadmap like that exist, there are only scattered recommendations across the scientific and business literature. We analysed and integrated these recommendations in the developed roadmap as such it enables the technology push and market pull perspective of the software platform and categorizing all the elements in layers including the maintenance proposal which make the final roadmap is implemented in any organization developing software platform.

The roadmap responds to the practitioner requirements to consider the process improvements and operational requirements and includes different format of visualization to communicate the plans more

intuitively. The road mapping processes themselves are not technically complicated to employ it within any company, and they are flexible enough to be readjusted for the particular needs.

The next step is to evaluate the developed technology roadmap in one of the SSP Platform releases and also asking for the opinion of a practioners, to see whether the developed roadmap indeed reflects current business needs, the pitfalls it possesses at the current stage, and what can be improved upon.

6

Roadmap Evaluation

The final phase of this thesis is dedicated to evaluation of the designed technology roadmap for the software platform. Our goal is to connect dots and show that the technology roadmap indeed deals with the identified set of challenges related to release planning of the software platform. Additionally, we examine the quality of the final technology roadmap via an interview with stakeholders (from figure 11) including product managers (section 6.2) and with evaluation methods such as validating the roadmap in one of the release planning according to the software reference framework (section 6.3). The expert interview also serves as the first step in the process of communication of the research findings. Further, section 6.4 discusses the validity of the technology roadmap by evaluating the research process from which this roadmap is the product.

6.1 Technology Roadmap Evaluation Using Identified Challenges

During the analysis of the problem environment in Chapter 3, we determined a set of challenges a company has to deal with while developing software product platforms. The final goal of the technology roadmap is to help organizations address every challenge from the list. Hence the first test for our technology roadmap would be to show that it address all identified challenges from the dimensions. In below table 6, we list all the challenges together with explanations how the technology roadmap will help in solving the issue.

The technology roadmap in a nut shell is a collated view of all high level requirements that influence the platform and facilitates the process of making release decisions during the meetings of a requirements advisory team. The roadmap will not produce a concrete release decisions strategy, but it will provide necessary information including the priorities, major & minor decision moments for all the elements in the platform during the release planning situation. This information will help the team to prioritize a balanced content from all the elements for the platform

releases. The column "Technology Roadmap" shows how roadmap is being used in order to obtain information required to resolve each challenge.

TABLE 6 CHALLENGES

Dimensions	Challenges	Technology Roadmap
Requirements Engineering Challenges	- *To support the release planning and requirements management process*	Full Technology Roadmap
	- *Manage and communicate the delivery of functionalities*	Technology Roadmap + Visual Formats+ Digital hosting in SharePoint
	- *Better oversight of known requirements from all the elements influencing platform release to avoid last minute fire drills in the requirements selection process.*	Six major elements covering all aspects of software engineering choices are included in the roadmap structure
	- *Visibility on functional features delivery beyond the current releases from the foundation platform*	Exploratory perspective + conducting workshops with Stakeholders
	- *Identify general functional capabilities for the platform based on future applications.*	Exploratory perspective + conducting workshops with Stakeholders
Architecture Challenges	- *Incorporation of technology into midterm plan*	Technology as one of the main element included in the roadmap structure
	- *Evolution of the architecture and its content has to be documented and should be readily available for reference and communication.*	Architecture as one of the main element included in the roadmap structure
Operations and Support challenges	- *Improve the operational and support process around the platform by reducing the*	Operational requirements and support process as one

	downtime, increasing the availability and maintainability of the platform	of element included in the roadmap structure
Software process Challenges	- Step up the support process and make it visible for all the application teams	Process improvement as one of the element included in the roadmap structure
	- *Platform needs to enhance the release and deployment process by improving development & testing strategies.*	Process Improvements + Workshop method
General.	- *Gather ideas and initiatives to improve the software engineering process around the SMART platform*	Process Improvements + Workshop method
	- *Document the lessons learnt from previous releases & gather improvement plans from stakeholders*	Process Improvements + Stakeholder workshops
	- *Communication model in form of visual graphical format.*	Different type of Visual formats (Bulls Eye, Gantt Chart Format)

6.2 TECHNOLOGY ROADMAP EVALUATION THROUGH PRACTIONERS INTERVIEW

As the technology roadmap design was partly based on the information received from Shell SMART solutions platform, it would be beneficial to validate the final approach with a Shell SMART solutions platform. It should satisfy the same requirements as we set up for the interviews from Chapter 3. The product managers of Shell SMART solutions platform should look beyond typical quarterly or "next release" horizon and be interested in the implementation of the technology road mapping procedures that will facilitate an effective decision making during the strategic release planning activities.

We found a chance to introduce the technology roadmap to the stakeholders involved in the SSP platform during a one hour interview. As stated earlier, the SSP platform is highly exposed to a diverse range of

application product teams; hence this gave us a good opportunity to compare the roadmap with the current state-of-the art regarding the software platforms within a company, find which ideas behind the roadmap are valuable, and what may require improvement.

After the short roadmap introduction in the beginning of the interview, further discussion was held around the following questions:

1. Would it add value?
2. Would it be possible to implement such roadmap within the program of the software platforms?

Below, we summarize the major comments regarding these questions with a short evaluation overview in the end.

6.2.1 The Technology Roadmap added value

The stakeholders of the SMART program sees the main value of the framework in its mapping process that allows to extract the content on the best practices, process improvements and product features/capabilities all from one document.

- Business Analysts found technology roadmap as an important base line to be used in the evolution of future platform releases. Especially release goals and high-level features were seen as the missing link between business view and requirements engineering, since this information helped in focusing the requirements engineering decisions at the project level. In addition, business analyst and product managers found it easier link high level features than to smaller requirements. One of the platform business analyst commented –

  "It can be used as the high level release calendar for SSP foundation and starting point for Scope management"

- Application Product Managers felt to use the roadmap templates to communicate their decision about future development steps of the products other application product teams. This helped in explicating some tacit knowledge about the business viewpoint on the product to other stakeholders. However, as one product manager put it, *"We personally do not gain so much from doing this. This information we write down is something that we know already"*.

"The benefits from the co –operative working style through product technology roadmap seemed beneficial to practioners".

In practice, the aspect of being able to plan co –operatively was not used, even thought the process description encouraged doing so. Practioners further suggested that technology roadmap will help in starting the development by setting the high-level targets beforehand and cooperatively so that the application product teams and operational and support team can prepare their activities at the same time as platform release planning activities.

- Platform Product Managers found it difficult that the roadmaps seemed to get useless very quickly. Even though the roadmaps had been drawn up on the basis of the process, the future releases did not follow the decisions made in the roadmaps.

"If we have a look at the roadmaps we did last year, how much have we done of what we planned?"

complained one product manager. The roadmaps were valid at one moment, but afterwards new application product needs as well as resource conflicts had caused changes that were not anymore documented at the roadmap level. The platform product manager further emphasized that, they might lose sight of the business view, if the features/content in the roadmap were too technical and detailed and might be of no use in further development work if the information was too high level and rough.

Project mangers Release roadmaps can be drawn using the technology roadmaps. Usually, release roadmaps are written for just one or two forthcoming releases since, project mangers felt that they did not yet have enough content to make decisions concerning the future releases. But technology roadmaps seemed to help them identifying the features contents influencing upcoming releases. One of the project managers appreciated roadmap saying:

"Roadmap helps to plan the skill pool for the upcoming releases based on oversight of the features"

- Architects emphasized that the roadmap is helpful to draw their reference architecture documents and that

> *"It provides full holistic overview of the content/features influencing the platform to all involved stakeholders all the time"*

Even though it adds value above means, owner of the roadmap stated that maintenance of the roadmap is really difficult and tedious process.

6.2.2 The Technology Roadmap Implementation Possibility

According to the stakeholders (platform product manager & business analyst), in order to implement (positioning coupled with maintenance) the roadmap within the SMART platform program, we should base it on the current standard work process of the foundation platform and business analysis procedures that are relatively similar across organizations developing software platforms. The technology roadmap can be further positioned within the product portfolio planning element of the software platforms. During every release of the foundation platform, the roadmap is updated by the BA of the foundation. The implementation proposal is mentioned in the figure 28.

(3) Platform Product Manager creates the release initiation project Charter for the foundation releases.
(4) Business Analyst gathers and analysis initial requirements for scoping scenarios using roadmap as the starting point for the scope management.
(5 & 6) Business Analyst provides walkthrough on the initial requirements to RAT (Requirements Advisory Team) members and architects.
(7) The requirements (scoped scenarios) are shared to program council for the sign off on the content for the particular release.

- The corresponding scoped requirements are moved & updated in the new release folder in requirement management repository.
- In this stage the *roadmap is validated* and content/features are moved according to the release content in the repository. This validation is done by the Business Analysts of the foundation platform.

(8) The subsequent development process including development, test and deployment process are followed.
(9) At the end of the release the scope is validated with the release notes and repository is updated with the backlog. *The roadmap is again validated* by the Business analysts at this stage and content / features are moved according to the release notes and repository updates.

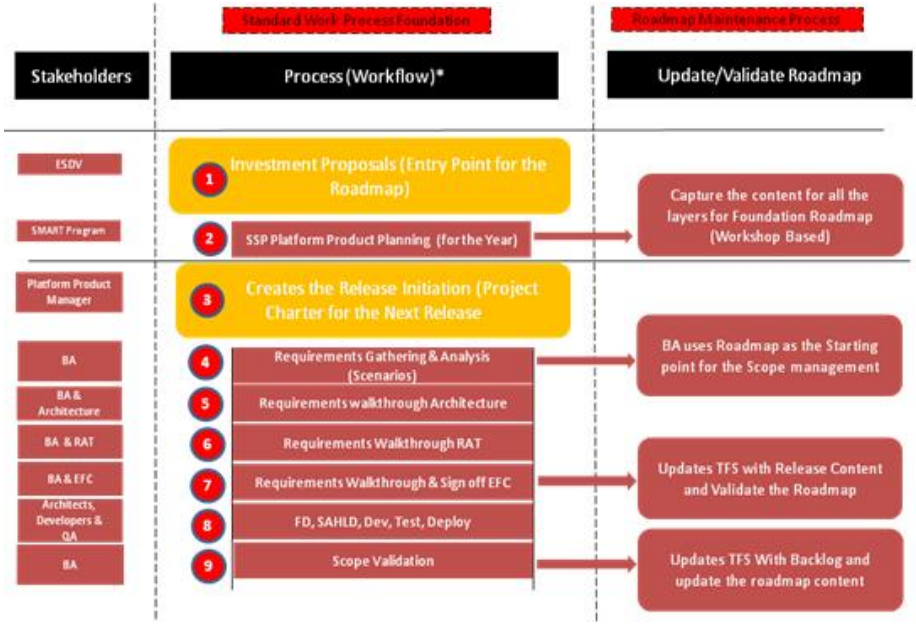


FIGURE 28 IMPLEMENTATION PROPOSAL

This implementation proposal is consistent with the standard work process of the software platform.

6.2.3 Feedback Overview

Based on the received feedback we can compare the developed technology road mapping process with the current approach within the company:

- Step 1: The robust maintenance process to update the roadmap frequently with quality content should be in place and implemented with the current process.
- Step 2: The road mapping process has to be linked to the program investment proposals for the year, then the budget can be allocated for the each release and content in the roadmap can be placed accordingly. However, this is not suitable in the practical situation, as the demand changes, delay in the previous

release leading to cost overruns and hence release content from the roadmap can't be executed as it is.

We think this comparison proves that we set the right direction for the road mapping approach; nonetheless, there is always a room for improvement. Below, we summarized some major comments from the practioners that would be useful to consider in future work:

- The maintenance process and implementation plan of the roadmap should be linked to the existing process of the developing the software platform products.
- The right level of abstraction level and amount of information needed by the different stakeholders should be agreed and roadmap hosted electronically should customize accordingly.

In general, during the discussion we observed that different stakeholders perceived value and challenges from the road mapping approach at different levels. There could be various explanations for it, but we think that the main reason lays in the scope and complexity of software platform and thus in the increasing amount of information required to make right decision during the release planning activities.

6.3 Roadmap Evaluation in Release Planning

The roadmap in this thesis has been validated by applying to shell case in SSP platform in relation to activity in software product management namely release planning as explained in section 4.4. According to the van de weerd software product management (figure 29) can be subdivided into four process areas: Portfolio management to deal with the products in the product portfolio planning, product road mapping to deal with the different release of each product; release planning to deal with the collection of requirements of each releases and requirements management to deal with the consent of each individual requirement. This is consistent with the product portfolio planning introduced in chapter 2. As the product technology roadmap was based on the information received from the Shell case, it would be beneficial to validate the final approach with the Shell Organization. Hence the final SSP Technology roadmap was validated within Shell in one of the SSP 1.3 release planning exercise.

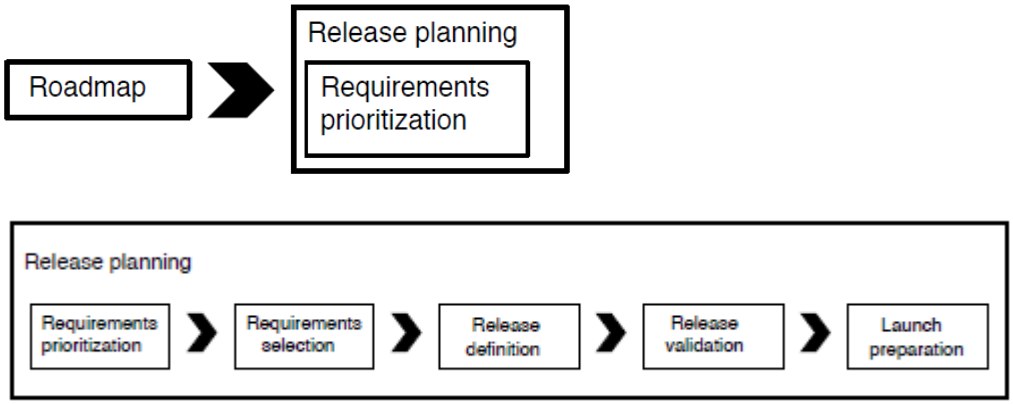


FIGURE 29 REFERENCE FRAMEWORK FOR RELEASE PLANNING

The SSP technology road mapping process from earlier chapter outlines high level feature for all the six elements assigned to subsequent releases of the SSP platform. The contents from the roadmap were used as the initial scope baseline during the (1.3) release planning. Due to change in the application product needs, demands & budget constraints at the time of release, the content/features in the roadmap were again prioritized during the requirement prioritization stage.

- Requirement Prioritization:

To decide on the relative priority of the things cost benefit analysis method were used, and developers estimated the effort associated with the each high- level features from the roadmap. Then the grouped feature groups were assigned priorities according to the stakeholder preferences using the stakeholder voting method (figure 30).

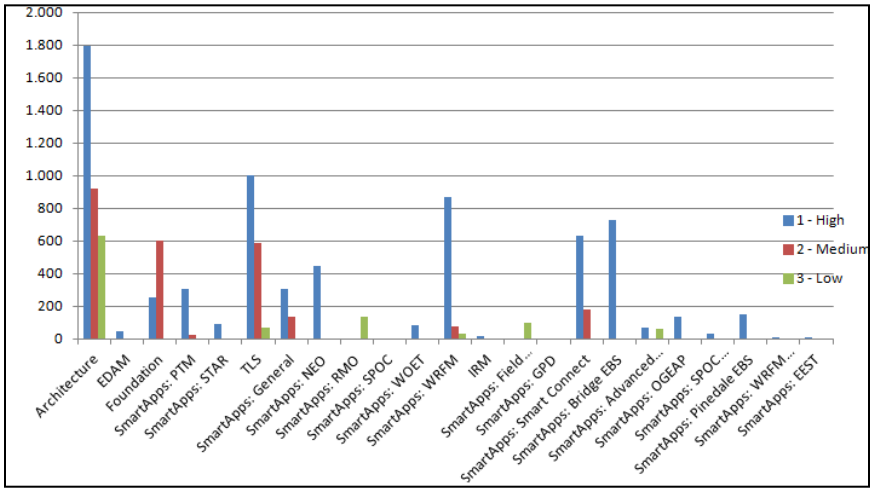


FIGURE 30 STAKEHOLDER VOTING

- Requirement Selection:

Once the feature groups were prioritized, it was the found that demands was more than the budget allocated for the 1.3 release. Hence the *requirement scenarios (figure 31)* were prepared from the prioritized feature groups in accordance with the *business objectives*. The scenarios included:

- *maturing the platform* (Architecture Changes, Operational requirements & Process Improvements)
- *enabling the SMART applications* (SMART Application, Functional Features & Technology capabilities)
- *enabling new technology capabilities* (Technology capabilities & Architecture changes).

FIGURE 31 REQUIREMENT SCENARIOS FROM ROADMAP

- Release Definition & Validation:

The scenarios were *validated* with architectures to understand the implicit dependencies among the feature groups. Then all the scenarios were presented to the "Platform Program Management" for the *decision making* and *sign off* for the launch preparation. The roadmap was again validated and updated based on the release definition for the SSP 1.3.

6.4 VALIDITY OF TECHNOLOGY ROADMAP

Validation of the technology roadmap is important because it is worth to invest in an implementation if it is likely that the technology roadmap will actually solve the practical problem. This section discusses the validity of the roadmap by evaluating the research process from which this technology roadmap is the product. Section (6.4.1) discusses how a solution design like the technology roadmap can be validated. Section (6.4.2) describes the internal validity by evaluating this research on the basis of seven guidelines. Section (6.4.3) states the external validity of the technology roadmap and generic elements that can be applied to other organizations. Finally, the last paragraph concludes that this technology roadmap is internally and for some cases externally valid.

6.4.1HOW TO VALIDATE

There are two different type of research in the IS (information system) field; behavioural science and design science. This research is an example of design science research which "creates and evaluates IT artefacts intended to solve identified organizational problems" [13]. The artefact created in this research is the technology roadmap for software platforms. Behavioural science studies ""phenomena that occur with respect to the artefact's use (intention to use), perceived usefulness, and impact on individuals and organizations (net benefits) depending on system, service, and information quality" [13]. In other words, creation and utility of artefacts are focused by design science and truth and validity of the artefacts are focused by behavioural Science.

One option to validate the artefact is the behavioural science. This option evaluates the research product. It answers whether or not the solution is true and it is the most thorough way. There is a second option that focuses on the creation process of the artefact and not focuses on the artefact itself. We suppose that if the research process is executed in a valid way, we can expect that the artefact is useful.

Since, the goal of this research is not truth but utility, and significant less resources is required in the second validation method, we will follow this approach in order to evaluate the validity of the technology roadmap.

6.4.2 EVALUATION OF THE RESEARCH PROCESS.

Hevner in his design science framework has identified seven clear guidelines for understanding, executing and evaluating this research. We

assume that the extent to which these guidelines are followed in this research process confirms about the validity of this research. The guidelines are shown in below table 7 and validity according to this research are discussed in the sections

TABLE 7 DESIGN EVALUATION METHODS [13]

Guideline	Hevner Description
Design as an Artefact	Design science research must produce a viable artefact in the form of a construct, a model, a method, or an instantiation.
Problem Relevance	The objective of design-science research is to develop technology based solutions to important and relevant business problems.
Design Evaluation	The utility, quality, and efficacy of a design artefact must be rigorously demonstrated via well executed evaluation methods.
Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artefact, design foundations, and/or design methodologies.
Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artefact.
Design as a Search Process	The search for an effective artefact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Guideline 1: Design as an Artefact

"*Design-science research must produce a viable artefact in the form of a construct, a model, a method, or an instantiation*" [13]

This research has produced technology roadmap for the software platforms that satisfies the criteria of an artefact, so this guideline has been followed.

Guideline 2: Problem Relevance

"*The objective of design-science research is to develop technology-based solutions to important and relevant business problems*" [13]

As described in chapter 2 and 3 the main problem of this research is both important and relevant to the business (Shell SSP Platform) and theory. Guideline 2 is followed as well.

Guideline 3: Design Evaluation

"*The utility, quality, and efficacy of a design artefact must be rigorously demonstrated via well-executed evaluation methods*" [13]

A cast study is performed at Shell – SMART Solutions platform to evaluate the technology roadmap. We did not evaluate an implemented version of the roadmap but assessed it via a practitioner's interview. We focused on usability of the technology roadmap in the interview as described in the previous section we were able to conclude that it adds value and usable. The data collected in the workshop plus the problem description in chapter 3 is knowledge base and might also be considered as form of informed argument. Concluding, three design evaluation methods have been used which confirms that guideline 3 as been used.

Guideline 4: Research Contributions

"*Effective design-science research must provide clear and verifiable contributions in the areas of the design artefact, design foundations, and/or design methodologies*" [13]

Three different possibilities to follow this guideline have been identified by Hevner. The contribution can come from the foundation or the methodologies or the design artefact. In this thesis, design artefact is the main contribution namely, the technology roadmap. It is implementable and demonstrates a clear contribution to the business environment, solving a practical problem as proven in the workshop and described in chapter. This demonstrates that we have followed this guideline as well.

Guideline 5: Research Rigor

"*Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artefact*" [13]

The literature has been used and explored as long it was appropriate which confirms that knowledge base has been effectively used and in broader sense by interviewing experts we also used the practical knowledge base. We do not claim that all view in practice and certainly all literature have been examined, but it has been to the extent that it was efficient. Therefore we believe that guideline 5 has been followed.

Guideline 6: Design as a Search Process

"*The search for an effective artefact requires utilizing available means to reach desired ends while satisfying laws in the problem environment*" [13]

This guideline states that artefact may not directly solve a problem in practice and hence it has go-through cycle of improvements. It also further explains the necessity of heuristics to find a good optimal solution. In broad context we improved the model of Phaal T Plan approach as well as the major elements of software platform by combining them and evaluating this through interviews and workshop. Constant feedback and iterations were used while designing the roadmap which does not give an optimal solution, but a good useable solution. Therefore, we believe that also guideline 6 has been followed.

Guideline 7: Communication of Research

"*Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences*" [13]

For the artefact to be truly valuable, this guideline emphasizes the importance of clear communication. Both management oriented and technology oriented audience should be served. "Technology-oriented audiences need sufficient detail to enable the described artefact to be […] implemented" and "Management-oriented audiences need sufficient detail to determine if the organizational resources should be committed to constructing […] and using the artefact" [13]

The management – oriented audience is the leadership team of SSP platform, because they have to decide whether to implement the roadmap or not. Partly they are also technology oriented audience and partly not. The communication to them is through the workshop as part of the evaluation methods as well as in several other meetings and discussions. The workshop itself turned out to be an effective communication technique. The final report on technology roadmap was also submitted to stakeholders as part of the project deliverable. Finally we believe that guideline 7 has been followed.

6.4.3 External Validity

On the basis of the guidelines, the internal validity of the technology roadmap was evaluated and we believe that the roadmap solves our problem statement. Now, we can describe something about the external validity, which discusses if the results from this case can be generalized to other organizations developing software platforms.

First, the technology roadmap architecture can be extendable to other organizations developing software platforms internally. The six layers of the architectures cover all the major elements of software platforms including application features, technology choices, architecture changes, innovation capabilities, operational requirements, and support process and software process improvements. All the organizations developing software platforms will need all these elements to make their software platform robust and maintainable. Hence the requirements from all these elements influencing the software platform can be easily captured within these six layers of the technology roadmap.

Even though the road mapping approach will differ from one organization to other organization, the customization approach from this thesis can be adapted by any organizations wanting to kick start the road mapping activity. Since it includes the major design principles of the technology road mapping approaches including context, format and stakeholder workshops from literature it gives enough flexibility for organization to customize the roadmap according to their operational settings.

The roadmap also provides enough structure to include all aspects of software engineering choices. For example, organization can alter the layers and sub layers of the roadmap to configure their interlayer dependencies among major elements and facilitate the integration of the functional feature with technology choices and applications.

To summarize, the customization approach described in this research can be adapted by any organizations developing software platform and changed according to their business needs or organizational settings. Nevertheless, the exact changes to adapt the technology roadmap to other organizations are subject to further research.

6.5 SUMMARY

In this chapter we performed the final phase of this research – the roadmap evaluation. We tested the utility and efficacy of the proposed roadmap by applying it to the identified challenges; by using it in the release planning activities and by discussing the approach with the practioners of the SMART foundation platform.

We discussed how the roadmaps are being used to gather information about all the elements influencing software platform to resolve challenge during the release planning activities. This information

serves as a key deliverable for requirement management meetings in order to arrive at efficient release planning strategy under time pressure.

The practitioner's interviews helped us understand what difficulties that can be associated with the implementation of the roadmap in the company. We also received some feedback on its overall value. The major change the roadmap makes to the development of software platform is that it facilitates and supports the decisions made during the features selection process during the release planning activities, while most of the companies still rely on group discussions during the release planning meetings.

7

Conclusions & Contributions

In this research, we present the technology roadmap for software product platforms which facilitates the decision making on prioritizing the content/features for the strategic release planning activities.

In order to develop the technology roadmap, different research phases have been accomplished, including investigating a wide range of scientific and industry papers, asking platform management & product managers about their needs regarding roadmap, integrating the received information in the roadmap, and undertaking an evaluation process of the developed roadmap and approach.

In this chapter, we summarize the main contributions of the present work, discuss research limitations, and suggest potential ideas for future work.

7.1 Main Contributions

We identify the following contributions of this research:

1. The research dives into software platform challenges of modern organizations and finds evidence of the importance of the technology roadmap in strategic release planning and requirement management activities. The challenges are categorized into four dimensions including Requirements Engineering, Architecture Changes, Operational & Support process and process disturbances.
2. The scattered advice from the numerous scientific and industry paper on technology roadmap and process is combined to develop the agreed technology roadmap and process. The architecture of developed technology roadmap addresses the literature gap to understand the complete domain of the software product platform. The architecture of the developed technology roadmap consist details of functionality features, technology choices, architecture changes, operational requirements and process improvements which covers all the aspects of the software

engineering choices and assist the product managers to understand the complete domain of the software product platform.

3. The developed technology roadmap integrates key findings and addresses every identified challenge, and shown to work in practice assuring rigor and relevance of this study. The developed roadmap for the software platform covering all the software engineering aspects such as functional features, technology capabilities, architecture changes, operational requirements &process improvements, hasten the process of release planning without affecting the quality of final decisions.
4. In general the technology road mapping process can be practically applied and implemented in the software engineering field to support the product managers in taking decisions for the release planning activities.

The practical contribution is inevitably given for all those organization developing software platforms who strive to implement a technology roadmap process. Although each technology roadmap process should be adapted to specific situation of the subject organization, the develop technology roadmap here at least already consider the requirements coming from the software platform and software engineering context and fulfils the purpose of the technology roadmap. If the subject organization wants to have another purpose fulfilled, the roadmap´s layers probably look different, and as a consequence there is a difference in the business processes to be dealt with in the workshops. But for the organization developing software platform who want to implement technology roadmap here developed and practically tested approach fits.

The practical contribution is even broader, namely for all organization wanting to implement and customize any kind of technology road mapping approach. Since this thesis also handles and outlines how to develop and customize a technology road mapping approach, the methodology can be followed by those who want to have technology roadmap process implements and cannot identify an appropriate one in the literature. The design process proved its value, which is demonstrated by the success of the project at Shell and theoretically by the Hevner design principle, therefore serving as a reliable guideline.

7.1.1 Reflections on Theory and Contribution

Road mapping technique has been used in the manufacturing industry for business oriented long term planning and technology forecasting [52]. The application of the approach in software engineering field is rather new and less investigated. The objective of this work is to provide information about the usage of this technique in software platform context and shed light on challenges that organizations introducing the road mapping approach may face. In this section we provide reflection on used literatures and then reflections from case study of Shell observed during product technology road mapping.

This research contributed academically in two ways. First, it verifies aspects about what is written in TRM literature. For instance, by applying the TRM approach in a special context, we add credit to Phaal statement which describes that TRM is a very flexible approach. Many authors moreover agree that it is necessary to adapt TRM approach to specific context of an organization which was accomplished by taking the requirements of the organizations developing software platform for the customization of the TRM approach. Measured by the feedback from the practioners interview and success of the first run- through it can be evidently said that T- plan from Phaal, including their customization approach, serves as an ideal basis to develop and customize a TRM approach for an untouched context so far. It also convinces the software reference framework by van de weed that roadmap actually works in the organization and support the release planning if it's positioned within the product portfolio planning. This was evaluated when validating the roadmap in the real release planning of the SSP platform and positioning the roadmap with in product –portfolio planning.

Second way, is that it adds knowledge and elaborates to the existing literature. The first contribution is a theory based technology roadmap structure covering all the aspects of the software engineering choices, which had not been adequately defined in the literature. Developed technology roadmap structure contrasts other technology roadmaps with an emphasis on the software engineering context and choices, which allows organizations to used this structure and build their own technology roadmap.

The most obvious and valuable contribution is the developed technology roadmap and approach itself, by adding another technology road mapping approach to the already existing array of approaches. With help of the definition of elements of software platforms and identified

requirements from organizations developing software platforms, developed technology roadmap and TRM approach adds a differentiating approach by considering the characteristics of the special context of software platforms. Thereby, it builds on the literature of software platforms and its effect on the organization, on the existing approaches, and closed the theoretical gap identified in the thesis introduction.

A contribution is made for the literature of the software product management, since the technology road mapping approach offers a solution to better manage the requirements influencing the software platforms and plan the release content subsequently. A prevalent short term view, problems in prioritizing the requirements influencing the platform for each releases from applications, technology choices, innovation capabilities & process improvements are typical phenomena when building software platforms. These requirements and phenomena are addressed in the technology roadmap, thus offering a solution by including them in each layer of the technology roadmap to address these complexities and cope with effects of complex software platforms for the organizations.

In a certain way, the chosen methodology also contributes to technology roadmap literature, showing exactly the way technology roadmap is developed on the basis of theory, business needs, practical feedback and iterative evaluation synthesis of design cycles. In contrast, many studies are mostly retrospective studies focusing more on the process description and content than on the design methodology making it complex for others to follow the procedure for their study or fully understand the methodology.

7.1.2 Reflections from Case (Shell – SSP Platform)

At Shell, we used our notation and process to discuss technology road mapping as applied to the SSP Platform, and carried out the road mapping process in co-operation with the company management. The road mapping activity was identified as one of the improvement activity and it was carried out separately apart from the normal work process of the SSP Platform. At the start of the study in 11/2012, SSP Platform had just release a major version of the platform product. The most important result of creating the initial technology roadmap were to get a clearer understanding of all the requirements influencing the platform for every release including software engineering choices to make the platform product robust and maintainable. Some of the reflections observed

during the technology road mapping in the SSP Platform are described below:

1. The visualization was found extremely helpful because it showed the development of the SSP platform, SMART Applications, functionalities, technology choices and architecture changes including process improvements in one picture. These issues had previously been found difficult to express and communicate. The feedback on the visualization resulted in two different formats, with the most important ones (high level detail) being shown only in the Bulls eye format and low level detail of information getting displayed in the Gantt Chart format.
2. The road mapping process helped the stakeholder around the SSP platform to refer to the components of the software platform and their relationships in common language for their envisioned product.
3. The Stakeholders of the SSP platform practiced road mapping by writing documents, power point slide decks that described as closely as possible the platform, the set of applications and their features as a function of time. However, the approach felt too cumbersome and the document was not kept up to date. Since then, the practice has been scaled down with less detail and a shorter time range. While our approach provides specifics beyond the roadmap visualization for the format of an actual roadmap document (hosted in SharePoint) by considering this issues.
4. In real cases, organizing stakeholder workshops is a great challenge due to the stakeholder's tight commitments on other tasks. During the road mapping process the stakeholders were contacted on one to one basis for gathering more information for each of the layers in the roadmap.
5. Estimating the cost of the features in the roadmap for prioritization was a great challenge. Most of the features were at high level and developers and architects had difficulties in estimating those features from the roadmap.
6. As described in the literature, the maintenance of the roadmap is difficult in the real case as well. Though the implementation plan for easy maintenance has been proposed, the maintenance of the roadmap ongoing basis is the difficult task due to sheer amount of already existing workload in the platform team. Also, other stakeholder may not see the benefit from their view points and

therefore feel unmotivated if just one person from the team has the responsibility of generating road mapping process.

7.2 Research Limitations

It cannot be claimed that the design process and the conducted research are free of limitations, or that the applied method represents the perfect choice to answer the research question. One of the main limitations is certainly related to the way of how the main research question is answered. In search for an appropriate TRM process, Hevener case based action research principle has been applied using the case study of one organization developing software platforms internally.

More robustness of the final TRM approach could have been reached by using a research design with multiple cases, meaning that the process is implemented in several software platforms to make inter-case analyses. But, this thesis focused on one case to make a start in suggesting an appropriate process. Furthermore, it does not seem feasible to manage several implementations within the scope of one master thesis. Instead, probably only multiple retrospective case studies would seem manageable for such a thesis. In contrast, the focus on one case allowed not only an in depth documentation and analysis, but also personal observation as a data collection method. This enabled triangulation in the analysis of data, and increased the validity. Finally, not to forget is the similar methodological approach taken by the previous authors of TRM literature.

Certainly, another main issue is that the process had been shaped to a great extent by experts coming from one organization. It provokes the questions if the TRM approach represents the best fit to SHELL SSP exclusively, or if is allowed to make inferences for other/all software platform. Undeniably, the evaluation is done only within Shell SSP Platform case. But thereafter, the evaluation had been backed up by further theoretical insights, so that justifications for the final design changes of the TRM approach could be made. Thereby, it can be claimed that the final TRM process is viable and effective also for other software platforms as well.

Another limitation is the involvement of experts in the design process. The experts for the identifying challenges and evaluation of the technology roadmap have been same in both the interviews. Moreover, the same stakeholders have participated in the entire TRM project at Shell SSP Case. Different experts in the different cycles could have brought more diverse views and feedbacks, and could have resulted in a

better TRM approach. However, after the first evaluation, it had the advantage that each person exactly knew what was expected. Relationships to the expert group could be built, and the introduction of the method could be shortened in the second iteration. The experts were familiar with both, the software platforms and the TRM process (at least after the first evaluation). Furthermore, they had personal interest, since it concerns their own organization

Despite the limitations, the TRM process was evaluated by Shell SSP Platform as successful. It is theoretically founded, and the major part of the process has already shown practical applicability. Therefore, further research may build on the findings of this thesis.

7.3 Future Research Possibilities

This research is a first attempt in the process of developing technology roadmap for the software platforms covering all the aspects of the software engineering choices to support the decisions made during strategic release planning activities. There is still a lot to investigate, discover, implement. This roadmap and process should be implemented in different software platform settings, business context and in different organization to identify all the pre requisites for practicing technology road mapping.

The limitations mentioned in the previous section serve as the base to continue investigating product technology road mapping issues. More software product platforms managers could be asked for feedback to discover whether some managers value the roadmap framework solution more than others, and if so which ones. Future adjustments to the roadmap framework should be based on the comments of those managers that see value in implementing a more automated process of arriving on release planning strategy.

As this report progressed, we noticed that there is a benefit in exploring the implementation of the roadmap within the existing process of the platform. The implementation of the technology road mapping aligned with the normal work process should be further researched by getting more empirical experience using our approach.

7

BIBLIOGRAPHY

[1] G. Ruhe and M. O. Saliu, "The art and science of software release planning," *IEEE Software,* vol. 22, no. 6, pp. 47-53, 2005.

[2] L. Xu and S. Brinkkemper, "Concepts of product software," *European Journal of Information Systems,* vol. 16, no. 5, pp. 531-541, 2007.

[3] W. Bekkers, M. Spruit, S. Brinkkemper and I. van de weerd, "A framework for process improvement in software product management.," *In Systems, software and services process improvement, Springer,* pp. 1-12, 2010.

[4] J. Bosch, ". Toward compositional software product lines. Software, IEEE, 27(3), 29-34.," *Software IEEE,* vol. 27, no. 3, pp. 29-34, 2010.

[5] M. Svahnberg, T. Gorschek, R. T. R. R. Feldt and R. Torkar, "A systematic review on strategic release planning models.," *Information and software technology,* vol. 52, no. 3, pp. 237-248, 2010.

[6] T. M. Jiang and M. Coyner, "Software process disturbances," in *Computer Software and Applications Conference, Compsac*, 2000.

[7] P. Groenveld, "Roadmapping integrates business and technology.," *Research - Technology Management,* vol. 50, no. 6, pp. 49-58, 2007.

[8] R. PhaaL, C. Farrukh and Probert, "Technology roadmapping—a planning framework for evolution and revolution.," *Technological*

forecasting and social change, vol. 7, no. 1, pp. 5-26, 2004.

[9] ,. S. Lee and Y. Park, "Customization of technology roadmaps according to roadmapping purposes: Overall process and detailed modules.," *Technological Forecasting and Social Change,* vol. 72, no. 5, pp. 567-583, 2005.

[10] T. A. Kappel, "Perspectives on roadmaps:how organizations talk about the future," *The journal of product Innovation Management,* vol. 18, pp. 39-50, 2001.

[11] R. Phaal, C. J. Farrukh, J. F. Mills and Probert, "Customizing the technology roadmapping approach.," in *Technology Management for Reshaping the World. Portland International Conference* , 2003.

[12] I. van de Weerd, S. Brinkkemper and R. Nieuwenhuis, "Towards a reference framework for software product management," in *In Requirements Engineering, 14th IEEE International Conference*, 2006.

[13] A. R. Hevner, S. T. March, J. Park and S. Ram, "Design science in information systems research.," *MIS quarterly,* vol. 28, no. 1, pp. 75-105, 2004.

[14] R. Phaal, C. J. Farrukh and D. R. Probert, "Developing a technology roadmapping system.," *Technology Management: A Unifying Discipline for Melting the Boundaries,* vol. 31, pp. 99-111, 2005.

[15] C. H. Willyard, "Motorola's technology roadmap process," *Research Management,* pp. 13-19, 1987.

[16] L. Lehtola, M. Kauppinen and S. Kujala, " Linking the business view to requirements engineering: long-term product planning by roadmapping.," in *Requirements Engineering proceedings* , 2005.

[17] G. DeGregorio, "Technology management via a set of dynamically linked roadmaps.," in *In Engineering Management Society*, 2000.

[18] D. Robertson. and K. Ulrich, "Platform product development.,"

Sloan Management Review, vol. 39, no. 4, pp. 19-31, 1998.

[19] J. Bosch, "Software product families: towards compositionality," *Fundamental Approaches to Software Engineering,* pp. 1-10, 2007.

[20] G. Ruhe, Product Release Planning: Methods, Tools and Applications., Auerbach Publications., 2011.

[21] J. Savolainen, M. Mannion and J. Kuusela, "Developing platforms for multiple software product lines," in *ACM*, 2012.

[22] M. H. Meyer and R. Seliger, "Product platforms in software development," *Sloan Management Review,* vol. 40, no. 1, pp. 61-74, 1998.

[23] E. Jandourek, "A model for platform development," *HEWLETT PACKARD JOURNA,* vol. 47, pp. 56-71, 1996.

[24] E. Parakal, " Process approach in realizing common software platform: a dream come true.," in *Proceedings of the 6th India Software Engineering Conference*, 2013.

[25] J. Bosch, "From software product lines to software ecosystems.," in *13th International Software Product Line Conference*.

[26] A. Taudes, M. Feurstein and A. Mild, "Options analysis of software platform decisions: a case study.," *MIS Quaterly,* pp. 227-243, 2000.

[27] M. Khurum, S. Barney, N. D. Fogelström and Gorsc, "Requirements management for continuous software product development.," in *The International Conference on Evaluation & Assessment in Software Engineering (EASE)*, 2010.

[28] S. A. Higgins, M. de Laat, P. M. Gieles and E. M. Geurts, "Managing product requirements for medical IT products," in *Requirements Engineering IEEE Joint Conference*, 2002.

[29] J. Bosch, "Maturity and evolution in software product lines: Approaches, artefacts and organization.," *Software Product Lines,* pp. 257-271, 2002.

[30] K. Pohl, G. Böckle and F. J. van der Linden, Software product line engineering: foundations, principles, and techniques., Springer, 2005.

[31] N. Chandrashekar, S. M. Gautam and K. S. Srinivas, "Challenges in Requirements Engineering for a Platform group: Case Study with a Medical Imaging Platform.," in *Software Engineering Research, Management and Applications*, 2006.

[32] B. Regnell and S. Brinkkemper, "Market-driven requirements engineering for software products.," *In Engineering and managing software requirements,* pp. 287-308, 2005.

[33] C. Ebert, "Requirements before the requirements: understanding the upstream impacts," in *Requirements Engineering, 13 International conference*, 2005.

[34] A. L. Fleury, F. Hunt, M. Spinola and D. Probert, "Customizing the technology roadmapping technique for software companies," *Technology Management for the Global Future,* vol. 3, pp. 1528-1538, 2006.

[35] N. Gerdsri, R. S. Vatananan and S. Dansamasatid, "Dealing with the dynamics of technology roadmapping implementation: A case study.," *Technological Forecasting and Social Change,* vol. 76, no. 1, pp. 50-60, 2009.

[36] Microsoft, "Microsoft Enterprise," 2013. [Online]. Available: http://www.microsoft.com/enterprise/industry/manufacturing-and-resources/oil-and-gas-mining/reference-architecture.aspx#fbid=jdyU0ZPlnKf.

[37] Shell , "Shell Internal Website," 2013. [Online].

[38] R. Phaal, C. J. Farrukh and D. R. Probert, "Customizing roadmapping," *. Research Technology Management,* vol. 47, pp. 23-37, 2004.

[39] R. Phaal, C. Farrukh and D. Probert, "Technology Roadmapping: linking technology resources to business objectives.," *Centre for*

Technology Management, University of Cambridge, pp. 1-18, 2001.

[40] D. A. Beeton, R. Phaal and D. R. (. D. R. Probert, "Exploratory roadmapping for foresight.," *International Journal of Technology Intelligence and Planning,* vol. 4, no. 4, pp. 398-412, 2008.

[41] M. Rinne, "Technology roadmaps: Infrastructure for innovation," *Technological Forecasting and Social Change,* vol. 71, no. 1, pp. 67-80, 2004.

[42] R. Phaal, C. J. Farrukh and D. R. Probert, "Characterisation of technology roadmaps: purpose and format.," in *Management of Engineering and Technology PICMET'01. Portland International Conference*, 2001.

[43] R. Phaal, C. J. Farrukh and D. R. Probert, "Technology management tools: concept, development and application.," *Technovation,* vol. 26, no. 3, pp. 336-344, 2006.

[44] R. E. Albright, "A unifying architecture for roadmaps frames a value scorecard. In Engineering Management Conference, 2003. IEMC'03. Managing Technologically Driven Organizations," in *The Human Side of Innovation and Change*, 2003.

[45] R. Phaal, C. Farrukh, R. Mitchell and D. Probert, "Starting-up roadmapping fast.," *IEEE Engineering Management Review,* vol. 31, pp. 54-60, 2003.

[46] J. D. Strauss and M. Radnor, "Roadmapping for dynamic and uncertain environments.," *Research Technology Management,* vol. 47, no. 2, pp. 51-58, 2004.

[47] T. Suomalainen, O. Salo, P. Abrahamsson and J. Simila, "Software product roadmapping in a volatile business environment.," *Journal of Systems and Software,* vol. 84, no. 6, pp. 958-975, 2011.

[48] J. J. McCarthy, D. J. Haley and B. W. Dixon, "Science and technology roadmapping to support project planning. In Management of Engineering and Technology, 2001. PICMET'01. Portland International Conference on (pp. 637-649). IEEE.," in

IEEE, 2001.

[49] L. Lehtola, M. Kauppinen and J. Vähäniitty, "Strengthening the link between business decisions and RE: Long-term product planning in software product companies.," in *IEEE*, 2007.

[50] K. Rautiainen, C. Lassenius, M. Vahaniitty and J. Vanhanen, "A tentative framework for managing software product development in small companies," in *Proceedings of the 35th Annual Hawaii International conference on system sciences*, 2002.

[51] K. E. Wiegers, Software requirements, Microsoft Press, 2009.

[52] D. Probert and N. Shehabuddeen, "Technology road mapping: the issues of managing technology change.," *International Journal of Technology Management,* vol. 17, no. 6, pp. 646-661, 1999.

[53] P. J. Whalen, "Strategic and technology planning on a roadmapping foundation. Research," *Technology Management,* vol. 50, no. 3, pp. 40-51, 2007.

[54] R. Zurcher and R. N. Kostoff, "Modeling technology roadmaps.," *The Journal of Technology Transfer,* vol. 22, no. 3, pp. 73-79, 1997.

[55] R. E. Albright, "How to use roadmapping for global platform products. PDMA Visions, 26(4), 19-22.," *PDMA Visions,* vol. 26, no. 4, pp. 19-22, 2002.

APPENDIX

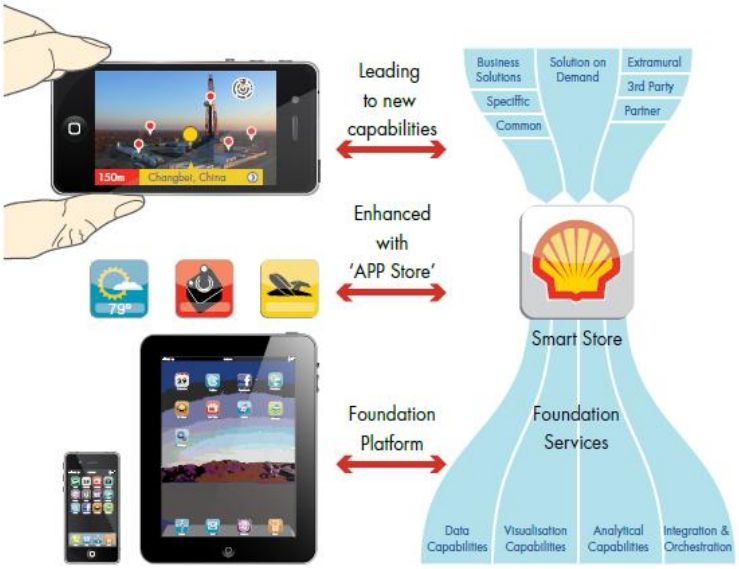
A.1 SMART SOLUTIONS

Smart Solutions are highly integrated and automated solutions in terms of Technology, People and Processes. They have an element of prediction to drive human behaviour (such as diagnostics & decision support) and maximize the exploitation of technology available on the market. Smart Solutions are about the integrated application of engineering, process automation, optimization, information and collaboration technologies applied to asset performance and integrity management across the entire asset lifecycle. This scope includes the application of emerging sensor technologies and addresses business processes ancillary to asset management (including logistics, engineering and supply chain). By adopting Smart Solutions, site and virtual asset teams are enabled to monitor performance and integrity of production, manufacturing and distribution systems, reliably forecast performance, recognize, predict, and diagnose issues, identify bottlenecks, evaluate options to remediate, make timely decisions based on reliable field data, and effectively implement interventions.

A 1.1.SMART SOLUTIONS PLATFORM (SSP) - HOW WILL IT WORK?

In order to best explain this model, the analogy of the Apple iPhone will be used. As a foundation the iPhone provides a platform with a basic set of functionality such as Wife wireless access, Phone, Internet browsing, email, Global Positioning (GPS) and various sensors. On the next level an 'Apps Store' is a placeholder where users communities can develop and publish applications that use the functionality of the iPhone platform and at the same time create new capabilities for the user community. An example is the phone Augmented reality application that adds information based on location; maps providing direction based on positional information and provides all kinds of location based information and services that are accesses through the wireless and internet connectivity. Likewise the Smart Solutions Foundation Platform will provide a basic set of functionality that users can 'configure' to meet specific business needs. Where for the iPhone 'apps' are developed, our aim is to make the platform such that 'Business Solutions' or 'Satraps' are made by ways of 'configuring required functionality'. The platform will unlock data from assets and make that available for further processing.

Based on business needs, new capabilities can be configured in 'Satraps' and stored in the 'Smart Store' for use by user communities across the Group.



A.2 Interview Outline and Questions

A 2.1 General Interview Outline

1. Introduction.
    - About me;
    - My research within TaCIT "SMART Solutions Platform"
    - My Goals of the interview
2. Overview of the stakeholder's role in the SMART solutions platform (Questions varied depending on the stakeholder)

A 2.2 Questions Asked to the Stakeholders:

Product Manager – Foundation Platform

1. What are the components/work streams of the foundation platform? What type of applications is currently using the foundation platform?
2. How are the business and technical roadmaps aligned from SMART foundation with individual APPS team?
3. How are the contents managed or requirements prioritized for the releases in SMART foundation? (How are the requirements for each release are built? – I understand, we get requirements from

apps project manager for each releases, but to improve the foundation for each releases how are the requirements listed?)

4. How internal improvement initiatives & requirements from the foundation platform are prioritized?

Product Manger – Application Teams

1. How the requirements from "Applications" are gathered and how it's being translated into SMART foundations?
2. What are the technologies used within "Applications" and expected technologies in future?
3. When does the application development starts within "Applications" (is in parallel with SMART foundation releases) or after the foundation is deployed?
4. What is the process to address the gap between SMART foundation releases and SMART Application releases?

Architects – Foundation Platform & Application Teams

1. What are the technologies that are now used in the SMART foundation platform and expected technologies in future?
2. What is the architecture style or pattern of SMART foundation? (Client –server or plug in or SOA etc)?
3. What is the vision of the foundation platform architecture?
4. What are the interdependencies between the work stream components?
5. How the technologies are scoped for each release and how are the technical roadmaps within SMART foundation planned?

Business Analyst – Foundation Platform & Application Teams

1. Who drives the requirements for SMART foundation platform? What are the requirements management process followed within the foundation platform?
2. How are the requirements prioritized for each releases of the foundation platform?
3. I assume that foundation platform itself have their own improvement initiatives & requirements for each releases; how are these improvement initiatives & requirements are balanced along with the functional requirements coming the applications teams?
4. What is normal Business analysis process?

Project Managers (Development, Test & Operations Manager) – Platforms

1. What is the development process followed within the foundation platform?

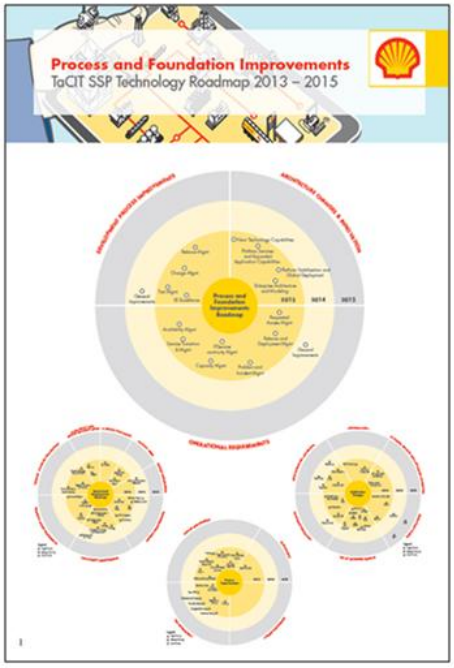
2. How are the test management process followed within the foundation platform? What challenges do they face since testing is end process of the development lifecycle?
3. How the performance of the platforms maintained is: since it connects to different app teams, how are the users or licenses maintained? What are the support model and operations process followed?
4. How are the deployment managed (global level and Regional Level)?
5. What are the internal improvement initiatives that can be implemented with respect to software engineering process and practices?

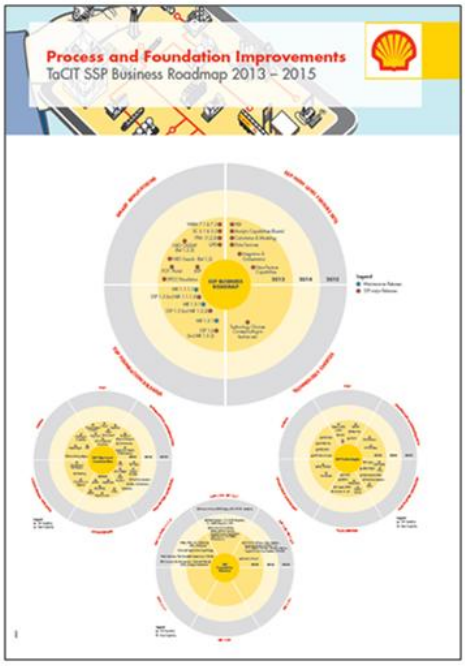
A.3 TECHNOLOGY ROAD MAPPING FROM LITERATURE

Name	Goal	Main Phases	References
Road mapping Life Cycle	Goal is to define and communicate product and technology strategy along with a longer, Smarter view of the future.	Initiation Maintenance Restarts	Albright &Kappel (2003)
T – Plan: Fast- Start Technology Road mapping (TRM)	Goal is to bring together key stakeholders and experts to capture, share and structure knowledge about the issue being addressed, to identify strategic issues and To plan the way forward.	Planning Facilitated Workshop Roll out	Phaal (2003)
Product Road mapping (PRM)	Goal is to help (1) Product Managers to create and maintain release roadmaps, (2) Managing situations where the same technical product is included in several products and (3) R&D to identify the needs for research projects	Preparation Approval communication	Lehtola (2005)
Release Road	Goal is to inform stakeholders	Data Collection Feature	Lehtola (2005)

mapping (RRM)	about scheduled future releases to help R&D, for example, plan their skills development and act as a trigger for early feature development, or marketing, to plan their future activities	Prioritization Release Planning Release roadmap Validation	
Road mapping Process	Goal is to improve internal process which may need improvement to increase R&D productivity or to upgrade a step in the drug discovery process that has fallen behind "industry standards"	Team formation Focus Technology /Workflow Analysis Implementation Review	McCarthy (2003)
Four-step model for creating and updating product roadmaps	Goal is to define and concretize the company's plans for technology and product development	Define strategic mission and vision, and outline product vision Scan the environment Revise and distil the product vision as product roadmaps Estimate product life cycle and evaluate the mix of development efforts planned	Vahanitty et. Al (2002)
Product Road mapping	Goal is to handle the development of the product roadmap, in which the future releases are planned based on themes and core assets	Theme Identification Core Asset Identification Roadmap Construction	Van de Weerd et al (2010)

A.4 BULLS EYE


Process and Foundation Improvements
TaCIT SSP Technology Roadmap 2013 – 2015


Process and Foundation Improvements
TaCIT SSP Business Roadmap 2013 – 2015