An Integrated Framework to Evaluate Information Systems Performance in High-Risk Settings: Experiences from the iTRACK Project

Ahmed A. Abdelgawad and Tina Comes

Abstract Evaluation and testing are significant steps in developing any information system. More attention must be devoted to these steps if the system is to be used in high-risk contexts, such as the response to conflict disasters. Several testing methodologies are designed to guarantee that software fulfills technology requirements; others will assure usability and usefulness. However, there is currently no integrated evaluation framework with agreed standards that bring together the three elements: technology requirements, usability, and usefulness. This gap constitutes a barrier to innovation and imposes risks to responders or affected populations if the technology is introduced without proper testing. This chapter aims to close this gap.

Based on a review of evaluation methods and measurement metrics for information systems, we designed an integrated evaluation framework including standard metrics for code quality testing, usability methods, subjective usefulness questionnaires, and key performance indicators. We developed and implemented a reporting and evaluation system that demonstrates our evaluation framework within the context of the EU H2020 project iTRACK. iTRACK developed an integrated system for the safety and security of humanitarian missions. We demonstrate how our approach allows measuring the quality and usefulness of the iTRACK integrated system.

Keywords Evaluation framework · Software quality testing · Requirements engineering · Usability · Usefulness · High risk · Humanitarian disaster

A. A. Abdelgawad (✉)
Department of Information Systems, University of Agder, Kristiansand, Norway
e-mail: ahmedg@uia.no

T. Comes
Faculty of Technology, Policy and Management, TU Delft, Delft, Netherlands
e-mail: t.comes@tudelft.nl

H. J. Scholl et al. (eds.), *Disaster Management and Information Technology*,
Public Administration and Information Technology 40,
https://doi.org/10.1007/978-3-031-20939-0_9

Introduction

In the Middle East and other high-risk areas, those who try to aid the most vulnerable are increasingly risking their own lives and safety. The number of humanitarian workers who fall victim to attacks continues to rise, according to the Aid Worker Security Database (Humanitarian Outcomes, 2019). Meanwhile, seeking to maintain access to populations in need, humanitarian organizations in the field are confronted with mounting tensions. Consequently, there is a new role for technology to support operations. Nevertheless, these innovations, particularly information and communication technologies (ICT) used in conflicts, can cause severe risks. These risks range from privacy violations to threatening the lives and safety of those the systems are designed to protect in the first place.

Evaluation and testing are a significant step in the development life cycle of any software system, and it is a vital phase in the quality assurance of ICT systems (Jovanović, 2009). The goal of software evaluation frameworks is to assess the quality and sophistication of the system from different points of view (Boloix & Robillard, 1995). However, thus far, there is no integrated evaluation framework combining testing functionality, quality, and usefulness of the software to assist in humanitarian conflict disasters. Such a framework requires the standards and problems of humanitarian innovation and experimentation to be met (Sandvik et al., 2017), and the context of the problem to be considered. In conflicts, a significant challenge is dealing with sensitive information and organizational barriers to information sharing (Van de Walle & Comes, 2015) and evaluating risks as they emerge (Van de Walle & Comes, 2014). The lack of an integrated framework and commonly agreed standards constitutes a significant barrier to innovation. At the same time, technology introduction without proper testing may impose risks to responders and beneficiaries alike.

Based on a review of evaluation standards and metrics, this chapter compiles and proposes an integrated evaluation framework for ICT systems in humanitarian conflicts. The proposed framework aims at assisting in measuring the quality and usefulness of a system on different levels, from the performance of individual components to the overall system. The Institute of Electrical and Electronics Engineers IEEE defines software system quality as *the degree to which a system, component, or process meets specified requirements* and *the degree to which a system, component, or process meets customer or user needs or expectations* (IEEE Computer Society, 1991). Meanwhile, the International Software Testing Qualifications Board ISTQB defines quality in general as *the degree to which a component, system or process meets specified requirements and user/customer needs and expectations* (ISTQB, 2018). It defines software quality as *the totality of functionality and features of a software product that bear on its ability to satisfy stated or implied needs* (ISTQB, 2018). In sum, the quality of the software is concerned with meeting the specified requirements and user satisfaction. The former is achieved by testing the software system's components individually or together, or the whole system against the requirements in terms of specifications, use cases,

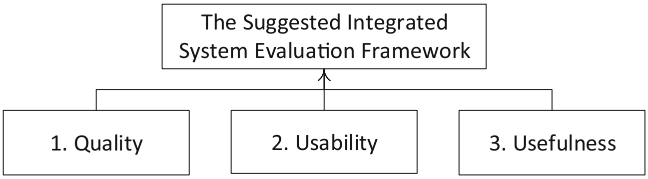

Fig. 1 The proposed integrated system framework

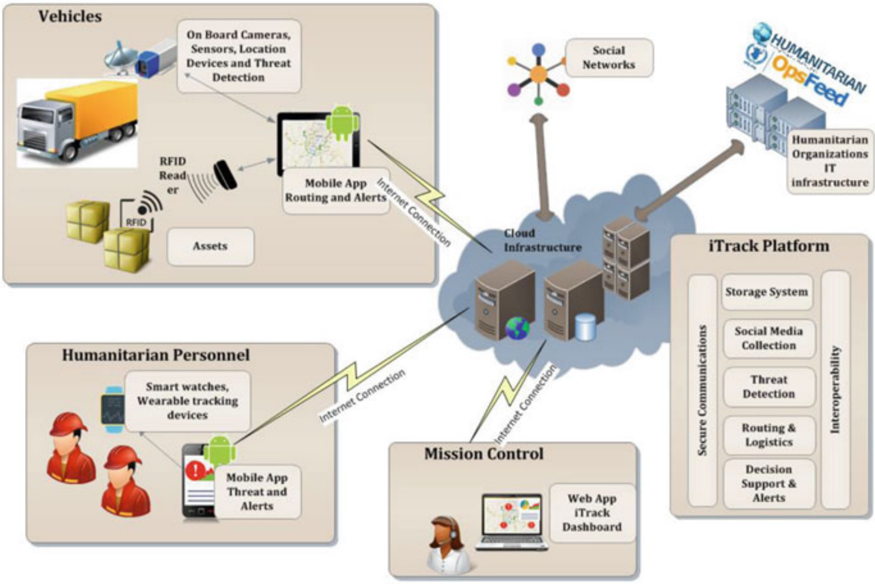

Fig. 2 Conceptual operational representation of the iTRACK system. (Adapted from iTRACK (2022a))

design documents, etc. In contrast, the latter is accomplished by testing the system usability and user satisfaction (Nielsen, 1993).

System usefulness means that *a product, website or application should solve a problem, fill a need or offer something people find useful* (Sauro, 2018a). Based on Fred Davis' usefulness construct, system usefulness is about helping users accomplish job tasks quicker, improving job performance, productivity, and effectiveness, and making the job easier to do in general. Figure 1 shows the pillars of our proposed integrated system evaluation framework.

The evaluation methods reviewed in this chapter and the methods included in our integrated framework are applied in the context of the EU H2020 project iTRACK (https://www.itrack-project.eu). The iTRACK project aims to develop a single open-source integrated system for real-time tracking of both people and assets, in addition to threat detection to support decision-making during civilian humanitarian missions run by humanitarian organizations operating civilian missions (iTRACK, 2018). Figure 2 illustrates the conceptual operational representation of the iTRACK

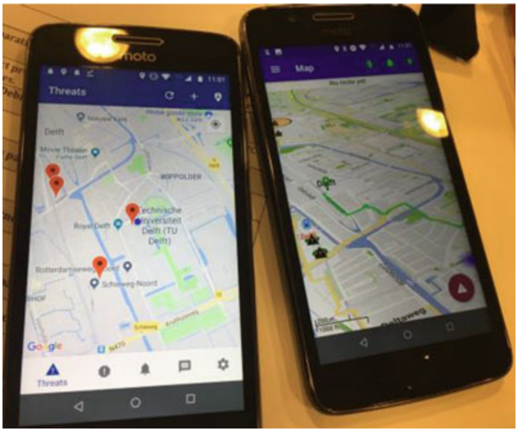

Fig. 3 iTRACK navigation app. (Adapted from iTRACK (2022b))

system, while Fig. 3 shows a snapshot of the iTRACK navigation app (displaying threat locations) running on mobile phones as an example of the iTRACK system components.

This chapter is organized as follows: the next section provides an overview of our methodology. The "Results" section describes the evaluation methods reviewed and the methods included in our framework in the context of the iTRACK project. Under the same section, we present our implementation of the evaluation framework in a computer system in terms of the iTRACK reporting and evaluation system. We conclude with a summary and discussion.

Methodology

To achieve the goal of this chapter, we surveyed relevant sources for "software testing methods" and "technology usefulness instruments" to collect quality and usefulness assessment methods and metrics. Websites of organizations connected to humanitarian conflicts were the target of our initial investigation, such as Aid in Danger, the European Interagency Security Forum EISF, and the United Nations Development Program UNDP. We followed an exploratory approach and used a variety of keywords like: "software testing," "software evaluation," "information system testing," "information system evaluation," "software quality," and "information system quality," sometimes even just using "software" and searched for relevant material in results. This search, however, did not yield sufficient results. To mitigate the situation, we have used the exact search keywords mentioned above and broadened our search circle to include sources like the following:

- International Organization for Standardization ISO (https://www.iso.org/publication-list.html)
- International Electrotechnical Commission IEC (http://www.iec.ch)

- IEEE (https://www.ieee.org)
- ISTQB (https://www.istqb.org)
- Scientific publications (via Google Scholar and others)
- Other sources which are available on the Internet in general

The results of this search were organized under the three pillars of our intended framework: quality, usability, and usefulness. The resulting framework was used to develop the iTRACK reporting and evaluation system.

Results

Framework Description

The quality of software, as indicated previously, is about meeting the specified requirements and user satisfaction. The former is achieved by testing the software system components individually or together and the whole system against the requirements in terms of specifications, use cases, design documents, etc. The latter is achieved via testing the system usability and user satisfaction directly with users and subjectively via questionnaires administered to them. System usefulness can be measured in terms of performance indicators of an individual user, a team, or an organization because of using the system. It can also be subjectively measured by explicitly asking the users to provide their opinions on the system's usefulness.

Our literature review results are compiled under the first two main subsections: "Software Testing and Quality" and "Software Usability." Each of these subsections was concluded by our selected methods and metrics for the iTRACK system. The third main subsection focuses on the usefulness of the iTRACK system.

Software Testing and Quality

Software Testing Methods

All software testing methods are classified under either Black-Box, White-Box, or in-between, i.e., Gray-Box (Jovanović, 2009). The software testing method is decided based on the testers' access to the internal structure of the software system under test (its source code):

- Black-box testing (a.k.a. specifications-based or behavioral testing) is a software testing method in which there is no need to access the source code of the tested item (Black Box Testing, 2018).
- White-box testing (a.k.a. clear-box, glass-box, transparent-box, open-box, code-based testing, or structural testing) is a software testing method to test a software

item with knowledge of its internal structure, design, and implementation (source code) (White Box Testing, 2018a, b).
- Gray-box testing combines the black-box and white-box software testing methods (Gray Box Testing, 2018).

Software Testing Levels

In addition to the testing method, software testing is also conducted on four levels:

- Unit testing level (a.k.a. component, module, program, or structural testing)[1] (Types of Software, 2018) is a typical white-box method testing level. *Unit testing is micro testing which is done by developers to ensure that each and every individual unit of source code performing well enough to match their expectation* (Types of Software, 2018; Müller & Friedenberg, 2011). This testing level is all about answering the question of "did we build it right?".
- Integration testing level aims at examining how units/components/parts of the system work together. The different units/components are tested working together to ensure that interfaces and interactions among them or other parts of the system (e.g., operating system, file system, hardware) are performing well and in compliance with the requirements/specifications (Types of Software, 2018; Müller & Friedenberg, 2011).
- System testing level is a system test concerned with the complete functionality and behavior of the whole system (Müller & Friedenberg, 2011). The environment where this testing level is conducted should resemble the production environment to reduce the environment-specific failures (Müller & Friedenberg, 2011). System testing level *may include tests based on risks and on requirements specifications, business processes, use cases, or other high-level text descriptions or models of system behaviour, interactions with the operating system, and system resources* (Müller & Friedenberg, 2011). This testing level inspects functional and nonfunctional requirements and could be conducted by an independent tester (Müller & Friedenberg, 2011).

Figure 4 shows the relationship between the testing methods and the testing levels.

The iTRACK Software Testing and Quality Assurance

Unit testing has been performed using the tools in the iTRACK development environment. The requirements for the tests were developed in a series of interviews, field research, and simulation tests (Noori et al., 2017). Complete documentation is available on the project website https://www.itrack-project.eu. Successive versions

[1] A structural or an architectural testing aims at knowing what is happening inside the system.

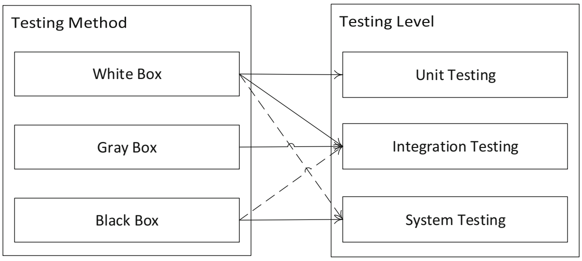

Fig. 4 Testing levels and methods

of the iTRACK corresponding deliverables have reported the resulted testing metrics. One of the metrics reported is the code coverage which is *an analysis method that determines which parts of the software have been executed (covered) by the test suite and which parts have not been executed,* e.g., *statement coverage, decision coverage or condition coverage* (McKay et al., 2016).

In the iTRACK development environment, integration testing for mainly the server-side components was carried out as well. In a simulation exercise in April 2018, another integration testing, including the client-side components, was performed in addition to system-level testing to evaluate end-to-end workflows. Before the final deployment, another system-level testing was conducted. After deployment, other metrics like the numbers and rates of bugs and issues reported, fixes, enhancements, improvements and new features released, and issues reopened (for others, please check (Data, 2018; Issues, 2018)) can indicate the quality of the iTRACK system.

Software Usability

Usability testing level (a.k.a. acceptance testing) is the final testing phase prior to sending the software to the production environment in the market. This level aims at answering the question of "did we build the right thing?". The testing is conducted firstly in the developers' workplace by the internal developers, testers, or users employed for that reason, which is called, in general, alpha testing. Then the testing is conducted at the users' place by the actual users to provide feedback before releasing the system to the market, which is called beta testing (Types of Software, 2018; Müller & Friedenberg, 2011). *The goal in acceptance testing is to establish confidence in the system, parts of the system or specific non-functional characteristics of the system. Finding defects is not the main focus in acceptance testing* (Müller & Friedenberg, 2011).

Acceptance in terms of usability is defined as "a quality attribute that assesses how easy user interfaces are to use. The word 'usability' also refers to methods for improving ease-of-use during the design process" (Nielsen, 2018a). Usability can be measured both objectively by asking users to complete specific tasks and observe them, and subjectively by asking users to fill out questionnaires about the usability of the software system.

Usability Testing Sessions

Usability testing aims at observing users using the tested software under test. A set of users, preferably similar in characteristics to the end users, should be employed and asked to fulfill goal-based tasks using the software; during these testing sessions, usability problems would be observed (Corona, 2019). Observations are made in terms of how users interact with the software. Then the developers will know the required features and understand issues facing the users while working with the software. Accordingly, developers can make improvements.

Usability Evaluation (Testing Metrics)

As mentioned above, the users will be given a set of tasks to complete during the testing session. The following metrics could be calculated:

Learnability

Is a metric for how easy it is for the user to learn using the system (Nielsen, 2018a; EN_Tech_Direct, 2018). Learnability can be measured by measuring if a user becomes faster in performing a task:

$$\text{Learnability} = \frac{T_2 - T_1}{T_1}$$

where T_1 and T_2 are the durations taken by the user to accomplish the same task for the first and the second times, respectively.

Efficiency

Measures how fast a user can accomplish tasks after learning the system (Nielsen, 2018a; EN_Tech_Direct, 2018). Efficiency could be measured by finding the total time saved between the first and the last times doing a specific task using the system.

Effectiveness

Measures how well the users achieve their goals by using the system (EN_Tech_Direct, 2018). Effectiveness could be measured by classifying the accomplishment level of the tasks by different users (in terms of **S** for success, **F** for failure or **P** for partial **S**uccess).

For example:

	Task 1	**Task 2**	**Task 3**	**…**	**Task N**
User 1	F	S	S		PS
User 2	S	S	F		F
…					
User M	F	S	PS		F

Completion Rates

"Often called the fundamental usability metric or the gateway metric, completion rates are a simple measure of usability. It's typically recorded as a binary metric (in terms of **1** for task success and **0** for task failure). If users cannot accomplish their goals, not much else matters" (Sauro, 2018b).

For example:

	Task 1	**Task 2**	**Task 3**	**…**	**Task N**
User 1	1	0	1		1
User 2	0	1	0		1
…					
User M	1	1	1		0

Usability Problems

This measure is about user interface problems that the users encounter during the test. The observer should "describe the problem and note both **how many** and **which users encountered it**. Knowing the probability, a user will encounter a problem at each phase of development can become a key metric for measuring usability activity impact and [return on investment] ROI. Knowing which user encountered it allows to better predict sample sizes, problem discovery rates and what problems are found by only a single user" (Sauro, 2018b).

Observer notes should be based on the **frequency** of the usability problem: "Is it common or rare?", the **impact** of the problem: "Will it be easy or difficult for the users to overcome?", and the **persistence** of the problem: *Is it a one-time problem that users can overcome once they know about it or will users repeatedly be bothered by the problem?* (Nielsen, 2018b).

Errors

"Record any unintended action, slip, mistake or omission a user makes while attempting a task. Record each instance of an error along with a description. For example, 'user entered last name in the first name field'" (Sauro, 2018b). Afterward, the observer can add severity ratings to the errors. Otherwise, categorize these errors. "Errors provide excellent diagnostic information and, if possible, should be mapped to [user interface] problems. Errors are somewhat time-consuming to collect, as they usually require a moderator or someone to review recordings" (Sauro, 2018b). Errors are detected via the observer's notes, for example, "user entered last name in the first name field" (Sauro, 2018b).

Task Time

"**Total task duration** is the de facto measure of efficiency and productivity. Record how long it takes a user to complete a task in seconds and or minutes. **Start task times when users finish reading task scenarios and end the time when users have finished all actions** (including reviewing)" (Sauro, 2018b).

	Task 1	**Task 2**	**Task 3**	**…**	**Task N**
User 1	00:05:30	00:14:30	00:05:30		00:01:30
User 2	00:04:25	00:13:20	00:04:25		00:01:20
…					
User M	00:06:45	00:12:15	00:06:45		00:02:15

Page Views/Clicks

"For websites and web-applications, these fundamental tracking metrics might be the only thing you have access to without conducting your own studies. Clicks have been shown to correlate highly with time-on-task which is probably a better measure of efficiency. The first click can be highly indicative of a task success or failure" (Sauro, 2018b). Page Views/Clicks could be detected by counting the clicks and page views by the system itself.

Expectation

"Users have an expectation about how difficult a task should be based on subtle cues in the task-scenario. Asking users how difficult they expect a task to be and comparing it to actual task difficulty ratings (from the same or different users) can be useful in diagnosing problem areas" (Sauro, 2018b).

Pre-task		1	2	3	4	5	6	7	
How difficult you think Task M will be?	Very easy	○	○	○	○	○	○	○	Very difficult
- Please explain your choice:									

Task Level Satisfaction

"After users attempt a task, have them answer a few or just a single question about how difficult the task was. Task level satisfaction metrics will immediately flag a difficult task, especially when compared to a database of other tasks" (Sauro, 2018b). For example, was "Task M" easy to do?

Post task		1	2	3	4	5	6	7	
How difficult did you find Task M?	Very easy	○	○	○	○	○	○	○	Very difficult
• Please explain your choice:									

Single Usability Metric (SUM)

"There are times when it is easier to describe the usability of a system or task by combining metrics into a single score, for example, when comparing competing products or reporting on corporate dashboards. SUM is a standardised average of measures of effectiveness, efficiency of satisfaction and is typically composed of 3 metrics: **completion rates**, **task-level satisfaction** and **task time**" (Sauro, 2018b).

Usability and User Experience Subjective Evaluation

Over the last 30 years, several usability and user-experience subjective questionnaires have been used to assess the usability aspects as well as reliability and validity of software systems. EduTech Wiki collected many of these questionnaires. They can be used for all systems, including websites and mobile apps (Usability and User Experience, 2018).

According to Perlman: "Questionnaires have long been used to evaluate user interfaces . . . Questionnaires have also long been used in electronic form . . . For a handful of questionnaires specifically designed to assess aspects of usability, the validity and/or reliability have been established . . ." (Perlman, 2018). In the following table, we enlist some of the subjective questionnaires resulted from our review.

Questionnaire title	Questionnaire type	Number of items	Sub-scales/construct	Reference
Perceived Usefulness and Ease of Use	7-points scale	12	Perceived usefulness, and perceived ease of use	Davis (1989)
Software Usability Scale (SUS)	5-points scale	10	Usability and learnability	Borsci et al. (2009), Brooke (1996), Sauro (2015) and System Usability Scale (2017)
Standardized User Experience Percentile Rank Questionnaire (SUPR-Q)	11-points scale	8	Usability, trust, appearance, and loyalty	Sauro (2015)
User Experience Questionnaire (UEQ)	7-points scale	26	Attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty	Laugwitz et al. (2006, 2008)

The iTRACK Usability and User Experience Testing

The iTRACK system consists of several packages with different roles in supporting humanitarian aid workers. Based on these roles, a list of usability tasks was prepared. This list compiles the possible iTRACK system features to be tested per the iTRACK system component. Each feature to be tested is provided with a description of its test. The idea, in general, is to find if the participants will be able to fulfill the required tasks with success, partial success, or failure. One of the iTRACK system features is the "threat creation," which, as the name implies, enables users to create a threat report so that other iTRACK system users can be careful. One example of a test activity description for this feature is "create threats on the map, indicate, e.g., threat types, estimated impact, etc."

The metrics mentioned previously in the review will be used whenever suitable to find our usability issues. For our selected usability task example, before doing this task, the participants should answer the following question:

Before Task

How difficult you think this task will be? - Please explain your choice:	Very easy	○ ○ ○ ○ ○ ○ ○	Very difficult

After finishing the task, the participants should log the time they took to complete it and report if the result was success, partial success, or failure. Then answer a question like the one they have answered before the task:

After Task

Task			Log
Task Time			__:__:__
Completion (Success, **F**ailure and **P**artial **S**uccess**)**			o S o PS o F
How difficult did you find this task? - Please explain your choice:	Very easy	○ ○ ○ ○ ○ ○ ○	Very difficult

These *Before* and *After Task* questions will enable calculating most of the usability metrics mentioned in the "Usability Evaluation (Testing Metrics)" subsection of this chapter.

As indicated in the review, many questionnaires could measure different constructs subjectively. Usually, users' time is limited and filled with several activities. To use this limited time efficiently, our team has selected only Davis's Perceived Usefulness and Ease of Use questionnaire and UEQ questionnaires to be administered as subjective usability measures. Davis's Perceived Usefulness and Ease of Use questionnaire is short and assesses the usefulness and ease of use, while UEQ provides more insights into the user's experience. These questionnaires are to be administered to users for each of the iTRACK system components individually to understand the text of the questionnaires within the context of each of these components.

iTRACK Perceived Usefulness and Ease of Use

Instructions:

- Try to respond to all the items.
- For items that are not applicable, use: NA
- Add a comment about an item if needed

Perceived usefulness			**1**	**2**	**3**	**4**	**5**	**6**	**7**		**NA**
1.	Using the iTRACK system in my job would enable me to accomplish tasks more quickly	Unlikely	○	○	○	○	○	○	○	Likely	○
2.	Using the iTRACK system would improve my job performance	Unlikely	○	○	○	○	○	○	○	Likely	○
3.	Using the iTRACK system in my job would increase my productivity	Unlikely	○	○	○	○	○	○	○	Likely	○
4.	Using the iTRACK system would enhance my effectiveness on the job	Unlikely	○	○	○	○	○	○	○	Likely	○
5.	Using the iTRACK system would make it easier to do my job	Unlikely	○	○	○	○	○	○	○	Likely	○
6.	I would find the iTRACK system useful in my job	Unlikely	○	○	○	○	○	○	○	Likely	○
Perceived ease of use			**1**	**2**	**3**	**4**	**5**	**6**	**7**		**NA**
7.	Learning to operate the iTRACK system would be easy for me	Unlikely	○	○	○	○	○	○	○	Likely	○
8.	I would find it easy to get the iTRACK system to do what I want it to do	Unlikely	○	○	○	○	○	○	○	Likely	○
9.	My interaction with the iTRACK system would be clear and understandable	Unlikely	○	○	○	○	○	○	○	Likely	○
10.	I would find the iTRACK system to be flexible to interact with	Unlikely	○	○	○	○	○	○	○	Likely	○
11.	It would be easy for me to become skilful at using the iTRACK system	Unlikely	○	○	○	○	○	○	○	Likely	○
12.	I would find the iTRACK system easy to use	Unlikely	○	○	○	○	○	○	○	Likely	○

iTRACK UEQ

Instructions: For each of the following items, mark one box that best describes the iTRACK system.

		1	2	3	4	5	6	7	
1.	annoying	○	○	○	○	○	○	○	enjoyable
2.	not understandable	○	○	○	○	○	○	○	understandable
3.	creative	○	○	○	○	○	○	○	dull
4.	easy to learn	○	○	○	○	○	○	○	difficult to learn
5.	valuable	○	○	○	○	○	○	○	inferior
6.	boring	○	○	○	○	○	○	○	exciting
7.	not interesting	○	○	○	○	○	○	○	interesting
8.	unpredictable	○	○	○	○	○	○	○	predictable
9.	fast	○	○	○	○	○	○	○	slow
10.	inventive	○	○	○	○	○	○	○	conventional
11.	obstructive	○	○	○	○	○	○	○	supportive
12.	good	○	○	○	○	○	○	○	bad
13.	complicated	○	○	○	○	○	○	○	easy
14.	unlikable	○	○	○	○	○	○	○	pleasing
15.	usual	○	○	○	○	○	○	○	leading edge
16.	unpleasant	○	○	○	○	○	○	○	pleasant
17.	secure	○	○	○	○	○	○	○	not secure
18.	motivating	○	○	○	○	○	○	○	demotivating
19.	meets expectations	○	○	○	○	○	○	○	does not meet expectations
20.	inefficient	○	○	○	○	○	○	○	efficient
21.	clear	○	○	○	○	○	○	○	confusing
22.	impractical	○	○	○	○	○	○	○	practical
23.	organized	○	○	○	○	○	○	○	cluttered
24.	attractive	○	○	○	○	○	○	○	unattractive
25.	friendly	○	○	○	○	○	○	○	unfriendly
26.	conservative	○	○	○	○	○	○	○	innovative

The iTRACK System Usefulness

System usefulness is about how the system is helping users in accomplishing their job tasks quicker; improving their job performance, productivity, and effectiveness; and, in general, making doing their job easier, in other words, the enhancement in performance of the users doing their jobs as a result of using the system (Davis, 1993). In predicting the actual system use, Davis found that system usefulness is 1.5 times more important than ease of use or usability (Sauro, 2018a; Davis, 1993).

The iTRACK system aims to improve the security and efficiency of civilian humanitarian missions. Using the iTRACK system is expected to enhance the performance of its users. In the following subsections, we will describe the metrics that we think would be useful in assessing the performance of the iTRACK system

*M*ission	***M*, *i***	***M*, *t***	***M*, *o***
*P*hase	***P*, *i***	***P*, *t***	***P*, *o***
*T*ask	***T*, *i***	***T*, *t***	***T*, *o***
	*i*ndividual	*t*eam	*o*rganisation

Fig. 5 Indicator measurement levels granularity (arrows go toward higher levels of aggregations)

components, the usage of these components, in addition to the performance of the individuals, teams, and overall organization because of using the iTRACK system.

A humanitarian mission could be divided into three phases: (1) planning, (2) executing, and (3) response and recovery. Each of these phases has different tasks according to the mission on the one hand and the threat/attack this mission is facing on the other hand. These tasks are performed by individuals who could be part of one team or gathered from different teams. Accordingly, an indicator could be on the highest resolution scale, i.e., measuring the performance of an individual working on one task. It could be scaled up to the case in which this individual is working through an entire phase or a whole mission. The same principle applies when the indicator is scaled up from an individual to a team or an organization. Figure 5 shows indicator measurement levels granularity that we have used while composing the performance indicators in the following subsections.

Usage Indicator of the iTRACK System

Individual Usage per System Component

Usage indicator ui_i: how many times an individual uses (open to look for or check anything) one of the iTRACK system components per time unit, therefore ui_i is measured in [times/hour].

Team Average Usage per System Component

Usage indicator ui_t: the average number of times of all individuals who belong to a team t use one of the iTRACK system components per time unit:

$$\mathrm{ui}_t = \frac{\sum_{i \in t} \mathrm{ui}_i}{|t|}$$

ui_t is measured in [times/hour], where $|t|$ is the number of all individuals who belong to the team t.

Organization Average Usage per System Component

Usage indicator ui_o: the average number of times of all individuals who belong to an organization o use one of the iTRACK system components per time unit:

$$\mathrm{ui}_o = \frac{\sum_{i \in o} \mathrm{ui}_i}{|o|}$$

ui_o is measured in [times/hour], where $|o|$ is the number of all individuals who belong to the organization o.

Coordination Indicator Using the iTRACK System

Reaction Time to Messages

The iTRACK system provides users with the ability to exchange text messages. The value of this indicator is based on how long it takes a user to react because of a message she/he has received on average. Indicators like replying to the message or performing an action because of the message content could be insightful. However, aside from being hard to measure, there are cases where a message does not need a reply or an action to be performed. For simplicity, reaction to a message could be considered as opening or reading this message (marking it as read). For example, during the first task of the planning phase PT_1, the time passed between receiving a certain message x by an individual until reading it is $\mathrm{rmt}_x^{\mathrm{PT}_1}$. Accordingly:

- For an individual, the total reaction time to all messages during this task is $\mathrm{rmt}_{\mathrm{total}}^{\mathrm{PT}_1} = \sum_{x \in \mathrm{PT}_1} \mathrm{rmt}_x^{\mathrm{PT}_1}$, and the average is $\mathrm{rmt}_{\mathrm{average}}^{\mathrm{PT}_1} = \frac{\sum_{x \in \mathrm{PT}_1} \mathrm{rmt}_x^{\mathrm{PT}_1}}{|\{x : x \in \mathrm{PT}_1\}|}$.
- For an individual, the total reaction time to all messages during all tasks of the whole planning phase is $\mathrm{rmt}_{\mathrm{total}}^{P} = \sum_{x \in P} \mathrm{rmt}_x^{P}$, and the average is $\mathrm{rmt}_{\mathrm{average}}^{P} = \frac{\sum_{x \in P} \mathrm{rmt}_x^{P}}{|\{x : x \in P\}|}$. Similarly, $\mathrm{rmt}_{\mathrm{total}}^{E}$ and $\mathrm{rmt}_{\mathrm{average}}^{E}$ and $\mathrm{rmt}_{\mathrm{total}}^{R}$ and $\mathrm{rmt}_{\mathrm{average}}^{R}$ can be calculated.
- For an individual, the total reaction time to all messages during the whole mission is $\mathrm{rmt}_{\mathrm{total}}^{\mathrm{mission}} = \sum_{x \in M} \mathrm{rmt}_x^{M}$ or $\mathrm{rmt}_{\mathrm{total}}^{P} + \mathrm{rmt}_{\mathrm{total}}^{E} + \mathrm{rmt}_{\mathrm{total}}^{R}$, and the average is $\mathrm{rmt}_{\mathrm{average}}^{\mathrm{mission}} = \frac{\sum_{x \in M} \mathrm{rmt}_x^{M}}{|\{x : x \in M\}|}$.

If the indicator is to be calculated for a team or an organization, the value can be calculated as the average of averages of all individuals who belong to that team or that organization.

Time-Saving Using the iTRACK System

This indicator requires two different entities (two individuals, two teams, or two organizations) to execute the same task. One of these entities uses the iTRACK system, while the other does not. Otherwise, a comparison can be conducted between the performance of the same entity in the current time and the last time this entity performed the same task, phase, or mission to measure the **learnability**. A comparison can also be conducted between the performance of the entity in the current time and the first time this entity performed the same task, phase, or mission to measure the **efficiency** (this answers questions like: how are we doing compared to the first time we have used the iTRACK system? and what is our overall trend using the iTRACK system?).

Individual Time-Saving Indicator

Let $\mathrm{ts}_i^{\mathrm{PT}_1}$ denotes the individual's time saved per task PT_1. Therefore, $\mathrm{ts}_i^{\mathrm{PT}_1}$ is the difference between the time elapsed by an individual (using the iTRACK) and the time elapsed by another individual (not using the iTRACK) – otherwise, the past reading of the time elapsed by the same first individual – executing the same task PT_1. Accordingly, the individual's time saved for all tasks during the whole planning phase is $\mathrm{ts}_i^P = \sum_{x \in P} \mathrm{ts}_i^x$; similarly, we can calculate the individual's time saved during the execution phase ts_i^E, and the individual's time saved during the response and recovery phase ts_i^R. Furthermore, the individual's time saved during the whole mission is $\mathrm{ts}_i^M = \mathrm{ts}_i^P + \mathrm{ts}_i^E + \mathrm{ts}_i^R$.

Team Average Time-Saving Indicator

For the task PT_1, the average time saved across individuals who belong to a team t performing this task is $\frac{\sum_{i \in t} \mathrm{ts}_i^{\mathrm{PT}_1}}{|t|}$. The same equation can be applied for a phase (e.g., P) and a whole mission, i.e., $\frac{\sum_{i \in t} \mathrm{ts}_i^P}{|t|}$ and $\frac{\sum_{i \in t} \mathrm{ts}_i^M}{|t|}$, respectively.

Organization Overall Average Time-Saving Indicator

For an organization o, the average time saved across all individuals who belong to this organization during the task PT_1, the phase P, for example, or the whole mission can be calculated by $\frac{\sum_{i \in o} \mathrm{ts}_i^{\mathrm{PT}_1}}{|o|}$, $\frac{\sum_{i \in o} \mathrm{ts}_i^P}{|o|}$, and $\frac{\sum_{i \in o} \mathrm{ts}_i^M}{|o|}$, respectively.

In general, the time saving related to specific tasks like loading trucks and completing deliveries. can be separately considered independent indicators.

Cost Saving Using the iTRACK System

The cost could be calculated as the actual cost of executing the task(s), phase(s), or mission(s) per an individual, team, or organization, which is challenging to be done quickly. Otherwise, it can be taken as the average cost per the time unit for an individual during executing task(s), phase(s), or mission(s) multiplied by her/his time elapsed executing this/these task(s), phase(s), or mission(s), respectively. The same approach can be applied to a team or an organization by summing the cost of the individuals who belong to this team or organization, respectively.

Like the time-saving indicator, this requires two entities (individual/team/organization) to execute the same task for comparison. One entity uses the iTRACK system, while the other does not. Otherwise, the comparisons can be conducted between the performance of the entity in the current time and the last time the entity performed the same task, phase, or mission to measure the **learnability**. The comparisons can also be conducted between the performance of the entity in the current time and the first time this entity performed the same task, phase, or mission to measure the **efficiency**. Like the time-saving indicators, cost saving for specific tasks like loading trucks and completing deliveries can be separately considered independent indicators on their own.

The iTRACK Usefulness Subjective Evaluation

Several questionnaires can subjectively assess the system's usefulness from the users' viewpoint. For example, from the reviewed questionnaires that cover usefulness in the "Usability and User Experience Subjective Evaluation" subsection of this chapter:

- Davis' Perceived Usefulness and Ease of Use
- CSUQ/PSSUQ
- USE

To subjectively measure the usefulness of the iTRACK system or one of its components, as mentioned earlier, Davis' Perceived Usefulness and Ease of Use questionnaire could be used, as it has been very well accepted and used for a long time (as it is part of the Technology Acceptance Model TAM) (Müller & Friedenberg, 2011). Considering the limited time of the users testing the iTRACK system, another reason to select Davis' is that it is shorter than the others.

System Implementation

System Overview

The iTRACK reporting and evaluation system are a web system implementation of the proposed integrated system framework that monitors different indicators concerning the iTRACK system and its users during different missions and presents these indicators. The web system was designed to serve the iTRACK system users by giving them indicators about the system performance and their performance. Figure 6 shows the context diagram or level zero workflow diagram (a.k.a. data flow diagram) of the iTRACK reporting and evaluation system. The main external entities in addition to the "User" are the "Database" on "MySQL Server" and the

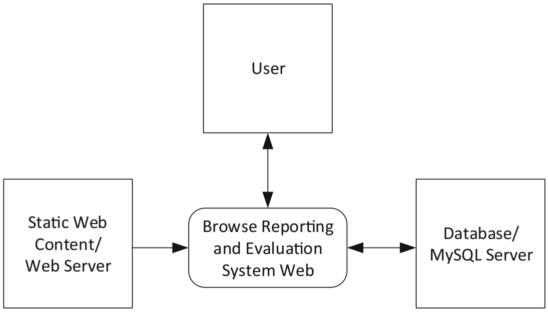

Fig. 6 Context-level workflow diagram of the iTRACK reporting and evaluation system

"Static Web Content" on the system's "Web Server." The database has several tables related to the users and system management and the main tables, which the system uses to store indicator-related data. Communication between the primary process of browsing the "Web Server" entity and other entities is two ways in all cases except with the "Static Web Content" entity, taking into consideration that the "User" entity can edit data in the "Database" entity when conducting system management.

The iTRACK reporting and evaluation system source code is available at https://github.com/ahgawad/iTRACK-Reporting-and-Evaluation-System, under GNU General Public License v3.0. The iTRACK reporting and evaluation system is web-based[2] and was built using common standard web technologies such as HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript (W3Techs, 2016) on the client-side. The iTRACK reporting and evaluation system uses Python/Django web-service framework on the server-side. Python programming language is popular among data scientists. According to the KDNuggets software poll in 2016, Python came in the second position after R with a share of 45.8%, with +51% growth over 2015 (R, Python Duel, 2017). Such popularity is reflected in the availability of several Python packages commonly used in developing scientific/data science applications like ours.

System's Graphical User Interface

The iTRACK reporting and evaluation system is a web-based system that provides different views corresponding to different functionalities. The system provides a *User view* for the users and an *Admin view* for the administrators to maintain the system's database. Figure 7 shows the components of the iTRACK reporting and evaluation system. The primary view is the *User view* which shows the iTRACK development indicators, the users' survey inputs and results, and the users' performance indicators, including standard operating procedures (SOP)/policies

[2] With proper installation, the system can be used offline on a PC or within a local area network.

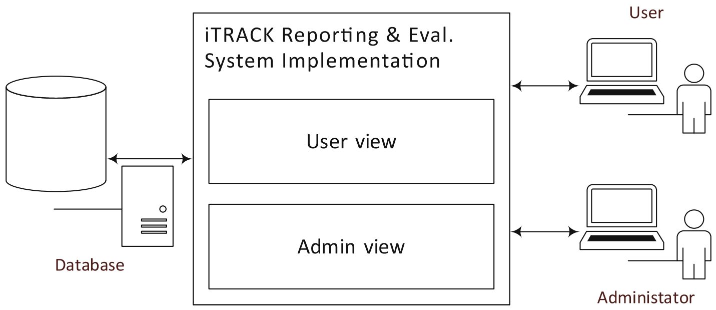

Fig. 7 Components of the iTRACK reporting and evaluation system implementation

compliance surveys inputs and results.[3] On the other side, the *Admin view* presents a tool for the system's administrators to add new development indicators, add new iTRACK future system components, add users' surveys, and add new SOPs/policies performance indicators in addition to users' accounts management. In addition to describing the system's graphical user interface, this section works as a user manual and guide on how to use the system.

User View

The iTRACK reporting and evaluation system has a main/instructions page. The primary/instructions page is shown in Fig. 8. Menus on the navigation bar at the top of this page and all other pages also work as an entry point to all system functionalities. In addition to the Home menu, which refers to this specific first page, the menus are:

- Development Indicators menu item refers and guides the user to the development indicators page.
- User Surveys menu item which takes the user to either:
  - Users Surveys Show page
  - Users Surveys Entry page
- Performance Indicators menu item which guides the user to one of four options which are as follows:
  - Performance Indicators Load sub-menu item which allows users with the correct permissions to load the performance logs generated by the iTRACK system to the iTRACK reporting and evaluation system

[3] Based on the best practices of the UN and other humanitarian organizations, the iTRACK project introduced a set of standard operating procedures (SOPs) and policies to support humanitarian missions.

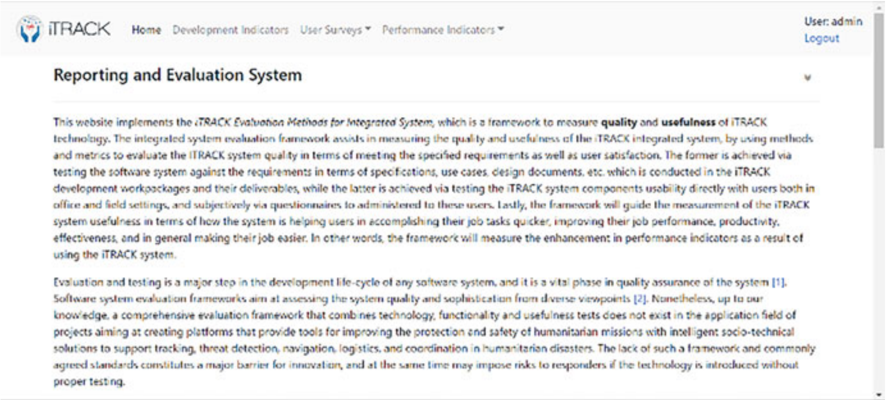

Fig. 8 The main and instructions page

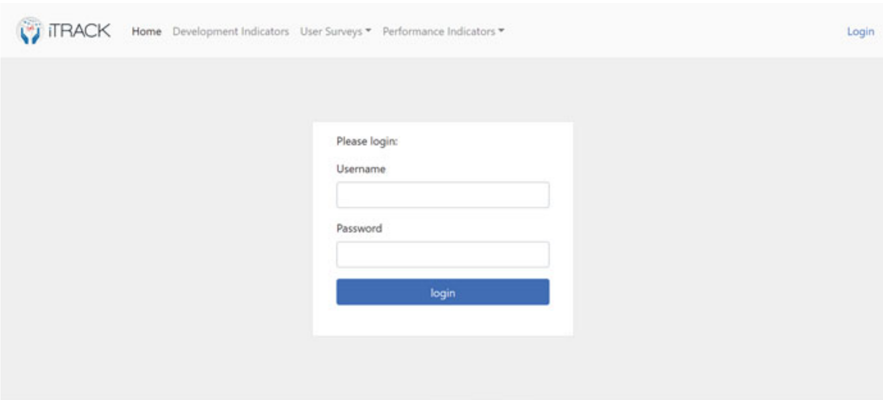

Fig. 9 Login page

– Performance Show sub-menu item which guides the user to show the results of the performance indicators logged by the iTRACK system components
– SOP Entry sub-menu item which enables a user with proper permissions to fill the SOPs/policies compliance survey for a particular mission
– SOP Show sub-menu item which allows a user to see the SOPs/policies compliance report

In general, the user view is possible to be accessed by users with proper permissions. These permissions can be set in the admin view by a system administrator (Fig. 9 shows the login page to the admin view).

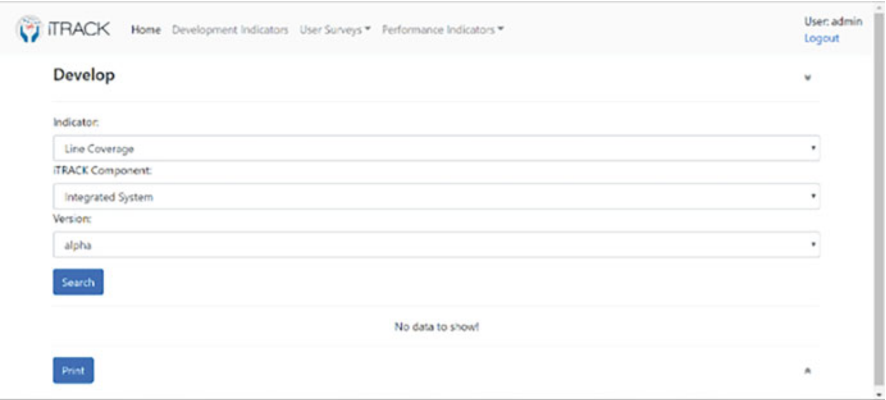

Fig. 10 The page of the development indicators (select indicator and component)

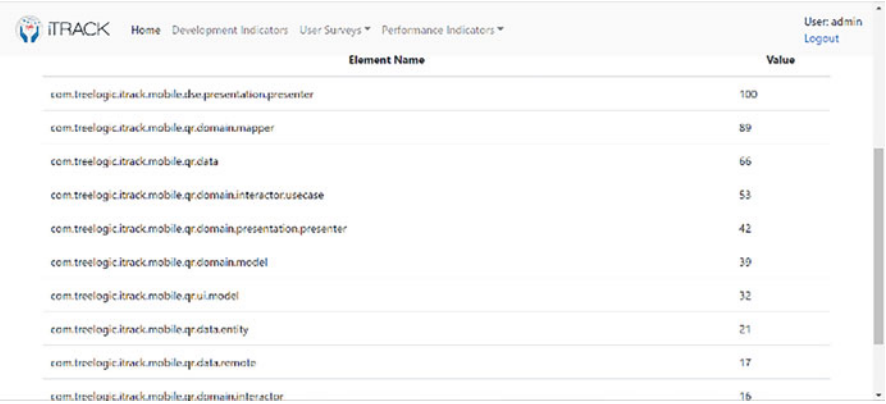

Fig. 11 The page of the development indicator (indicator values)

Development Indicators Page

Any logged-in user can view the development indicators on the Development Indicators page. The system is not limited to any of the development indicators, tested software components , or software versions presented on this page shown in Fig. 10 (Fig. 11 shows the indicator values). With future extensibility in mind, new development indicators, tested software components, and software versions can be added via the admin view.

Users' Surveys Entry Page

The users' survey page enables the logged-in user to fill one of the user surveys available in the iTRACK reporting and evaluation system for any tested software components. The user can select the survey, component name, and version, as shown

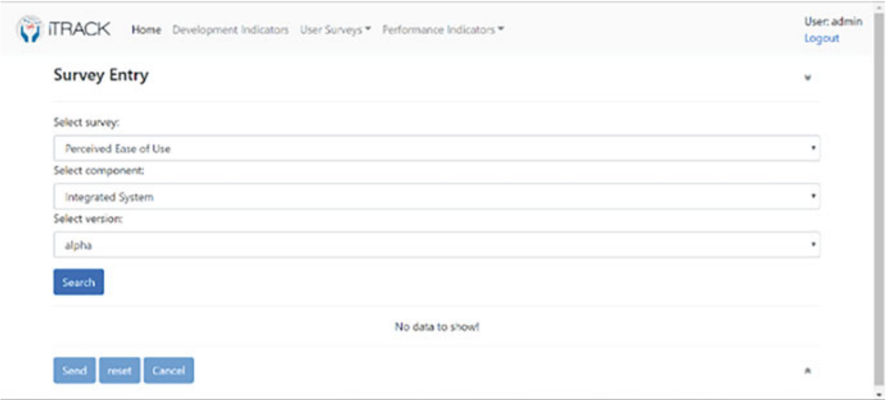

Fig. 12 The page of the user surveys input (select survey)

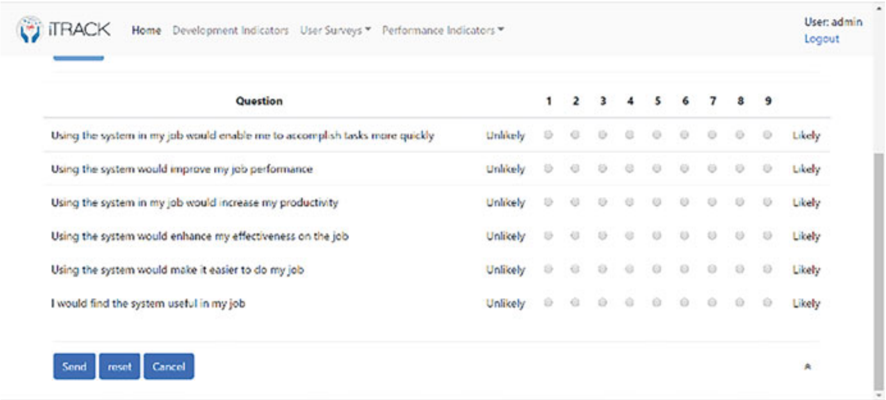

Fig. 13 The page of the user surveys input (survey to fill)

in Fig. 12. Pressing the Search button retrieves the selected survey from the system and shows it on the same page as shown in Fig. 13. The system will not allow the user to answer the same survey for the same combination of a tested software component and version more than one time.

Users' Surveys Results Page

The user-surveys results page displayed in Fig. 14 allows the user to see the collective results of a specific user survey for a particular combination of a tested software component and version for the team(s) she/he is a member of. If the user is an administrator, she/he will be able to see a collective result for the whole organization. The user might be interested in seeing more recent results or even older ones; the system allows the user to select a starting and ending date, which

Fig. 14 The page of the user surveys results (select survey)

Fig. 15 The page of the user surveys results

will be used to retrieve survey answers answered in between. The system retrieves the results from the database and shows them in the form of a diverging stacked bar chart (as shown in Fig. 15).

Performance Indicators Load Page

Software components log several indicators according to their design. The iTRACK reporting and evaluation system allows an administrator to upload any tested software component's log file (if prepared in the correct format, see the GitHub repository referred to earlier in the "System Overview" subsection for more information). Figure 16 shows the Performance Indicator Upload page, in which the administrator can provide the path of the log file. As soon as the log file is selected, the system parses and views it, as demonstrated in Fig. 17. If the selected file has

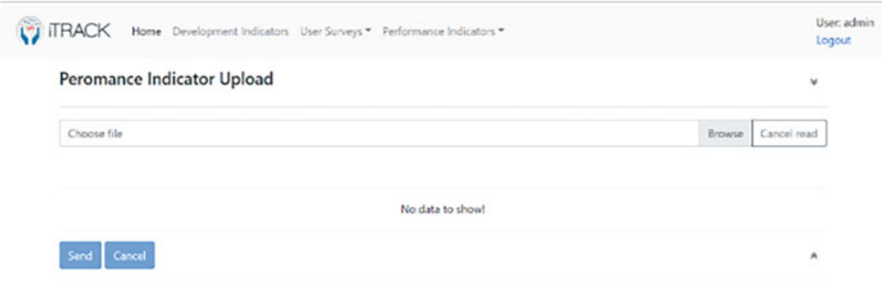

Fig. 16 The page of the performance indicators upload

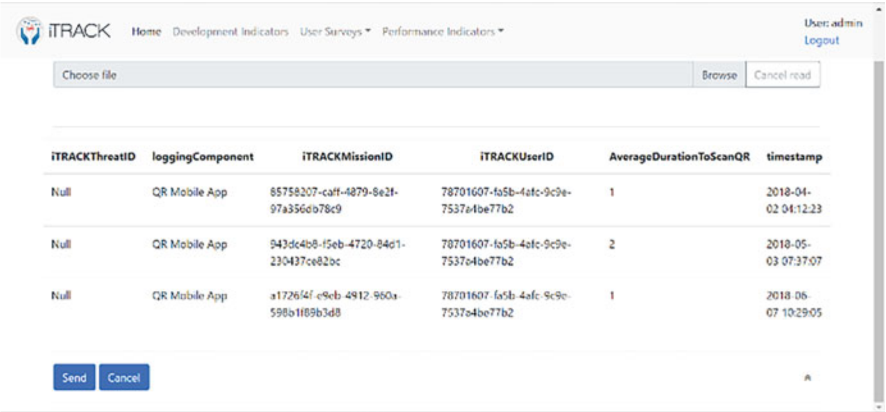

Fig. 17 The page of the performance indicators upload

any lines with formatting errors, they will not be parsed. The administrator can then upload the logs to the server of the iTRACK reporting and evaluation system. The system saves only new records and ignores any repetitions.

Performance Indicators Show Page

As described earlier, the tasks are performed by individuals. These individuals could be part of one team or grouped into different teams. Therefore, if an indicator is on the highest resolution (i.e., measuring the indicator's value for one individual working only on one task), it could be scaled up to this individual working through an entire phase or even the whole mission. The same principle applies when scaling up from an individual to a team or the whole organization. In the system, to view performance indicator results, the iTRACK reporting and evaluation system allows a user with administrative privileges, as shown in Fig. 18, to select:

- The user(s) to whom the performance indicator values belong, otherwise select an entire team

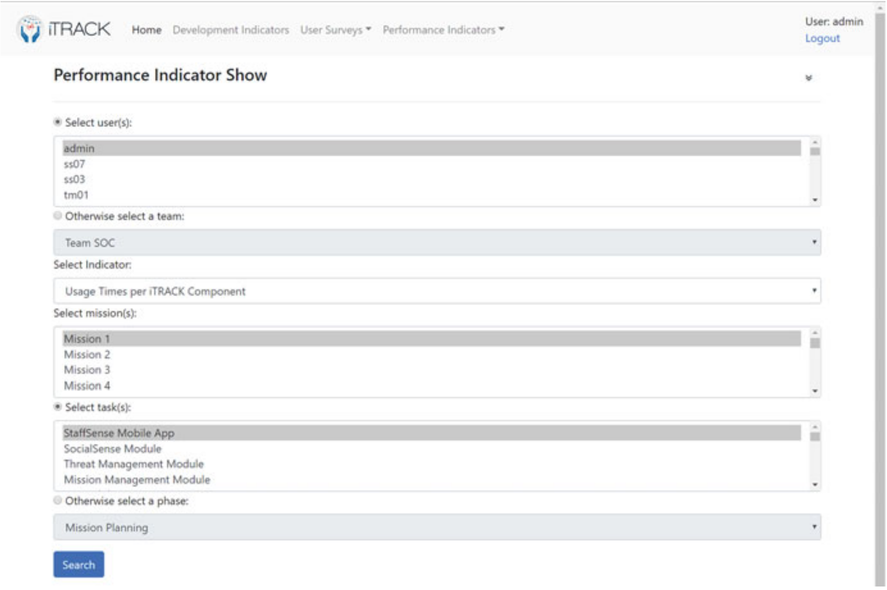

Fig. 18 The page of the performance indicators (select indicator)

- The indicator itself
- The mission(s) in which the performance indicator values have been captured
- The task(s) in which the performance indicator values have been captured, otherwise a whole phase

As shown in Fig. 19, the system shows detailed results concerning the iTRACK performance indicator for the selected parameters. To facilitate human readability, the results include textual comments about the indicator values, including some essential descriptive statistics like the general trend of the indicator, maximum value and its date, minimum value, and its date. In addition, the system presents a chart plotting the values of the indicator, including the linear trend.

SOPs/Policies Entry Page

The iTRACK technology is supposed to help humanitarians act according to the SOPs and policies. A logged-in user can fill the SOPs/policies compliance survey for a mission she/he is the leader of (otherwise, if she/he is an administrator), as shown in Fig. 20.

The user will be able to fill out a survey for the selected mission to assess the compliance of the staff of this mission with the SOPs and policies. Figure 21 shows a snapshot from the SOPs/Policies Entry page with SOPs and policies checkboxes list. The figure also shows an example of an error message that will appear if the

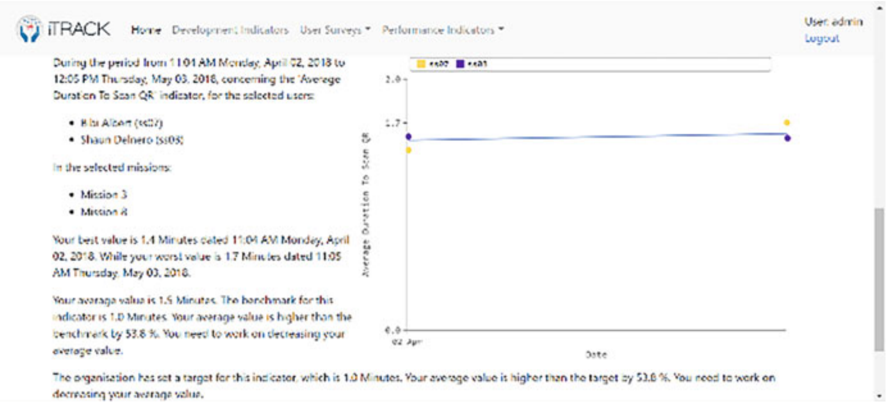

Fig. 19 The page of the performance indicators (human-readable textual results)

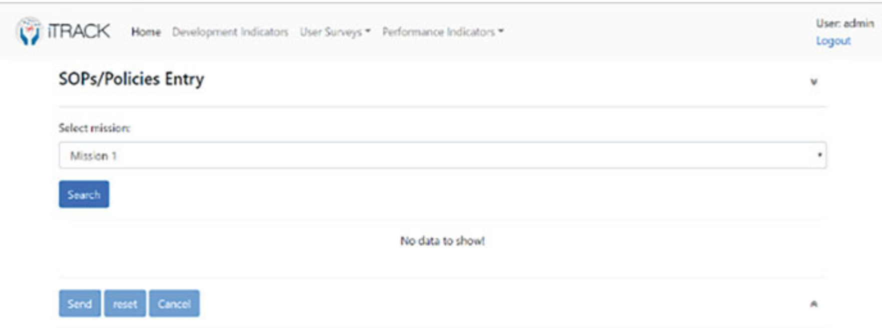

Fig. 20 The page of the SOPs/policies input (select mission)

user tries to check an SOP dependent on other SOPs that have not been checked yet or uncheck an SOP dependent on other SOPs that are still checked.

SOPs/Policies Show Results Page

A logged-in user with the correct permissions to view the SOPs/policies survey results for particular mission(s) can use the SOPs/Policies Show Results page shown in Fig. 22. The page allows the user to select a mission or more to view the results. The system accordingly shows the detailed results concerning the compliance with SOPs/policies of the selected mission(s), as shown in Fig. 23. The results are grouped under SOPs/policies tags. To also facilitate human readability, these results contain textual comments showing the SOPs/policies the team has not complied with. In addition, the system presents a chart showing the values of the SOPs/policies compliance indicator.

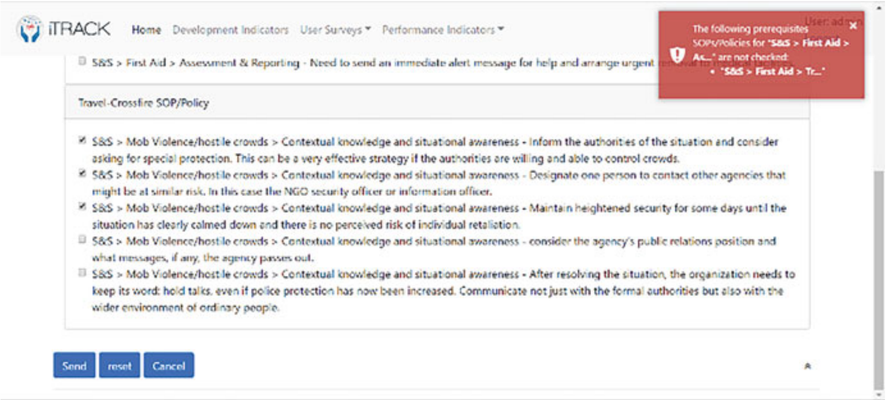

Fig. 21 The page of the SOPs/policies compliance entry (fill the survey with error messages)

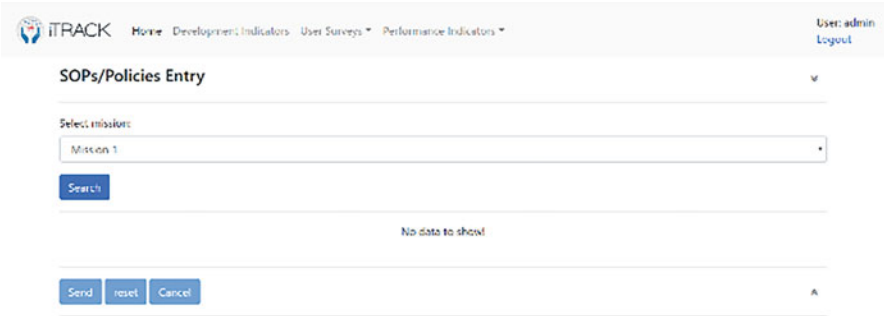

Fig. 22 The page of the SOPs/policies results (select mission(s))

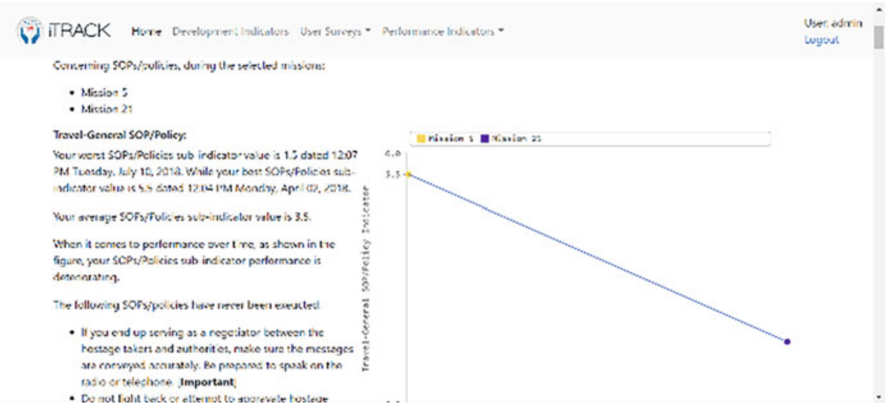

Fig. 23 The page of the SOPs/policies results (human-readable textual results)

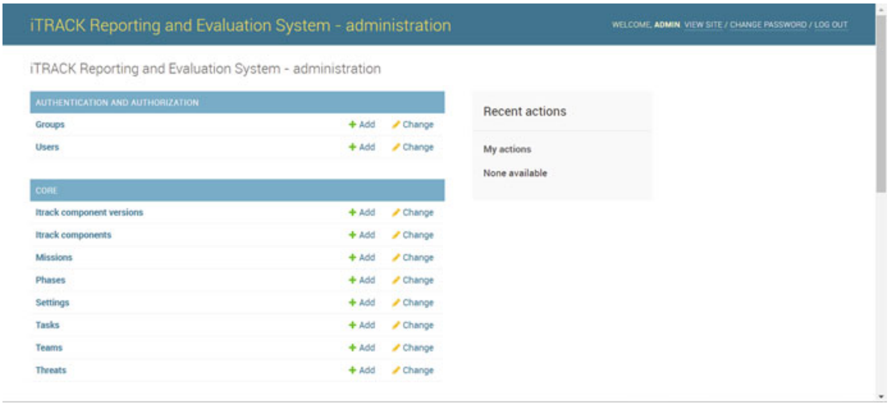

Fig. 24 The admin view for logged-in users

Administration View

The administration view could be viewed as a database management system for the underlying database of the iTRACK reporting and evaluation system. In this view, a user with proper administrator credentials can add, edit, and remove records from the different tables related to all indicators and results shown in all views of the *User view* mentioned earlier. This will keep the database updated with new and correct records. Figure 24 shows the page that will appear by calling the *Admin view* after passing the username and password authentication page.

As an example of the related admin pages, the previously mentioned set of performance indicators was added to the iTRACK reporting and evaluation system. Nonetheless, an administrator can add a new/edit/delete one or more performance indicators. As shown in Fig. 25, a performance indicator can be defined by:

- Adding an indicator name
- Adding the indicator's unit of measurement
- Deciding if the indicator is an average or an absolute value
- Deciding if the indicator uses a normal or inverted scale
- Deciding if the indicator is related to user performance or related to the performance of a technical component (e.g., the "Threat Detection")
- Adding the indicator's whereabouts (which is the name used for this indicator in the log file generated by the software components)

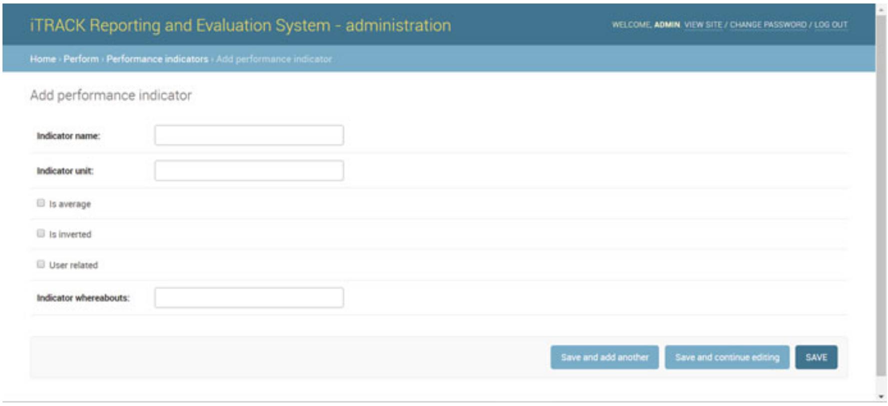

Fig. 25 The page to add a new performance indicator in the admin view

Summary and Discussion

Evaluation and testing are significant steps in developing software, but they are critical if innovation is used in highly sensitive contexts such as humanitarian conflicts. It is a vital phase in quality assurance of the system in terms of assessing the system quality and sophistication from diverse viewpoints. Nonetheless, an integrated evaluation framework that combines technology, functionality, and usefulness tests does not exist. This chapter presents metrics that were developed to help measure the quality and usefulness of a system and apply them to the case of the iTRACK system, a tracking and monitoring system for humanitarian conflicts.

This chapter reviewed the adequate evaluation methods and metrics to compile this integrated evaluation framework to assist in measuring the quality and usefulness of the iTRACK system. We have indicated that the software system quality is assessed in terms of software testing. We have introduced different software testing methods and levels used in software testing in general.

The usability of the iTRACK system is assessed separately, either via the system usability testing directly with users or via questionnaires administered to them. Moreover, for users to find any system useful, this system should solve a problem they face, fill a need, or offer them something. System usefulness is about helping accomplish job tasks quicker; improving job performance, productivity, and effectiveness; and making it easier to do the job. To measure the usefulness of the iTRACK system, we have proposed several performance indicators, in addition to subjectively recognizing the users' opinions about the usefulness of the system.

The iTRACK integrated system evaluation framework has been reviewed by several iTRACK project partners that belong to academia and software development, and their notes were taken into consideration in the final version. Figure 26 shows the pillars and details of the final framework.

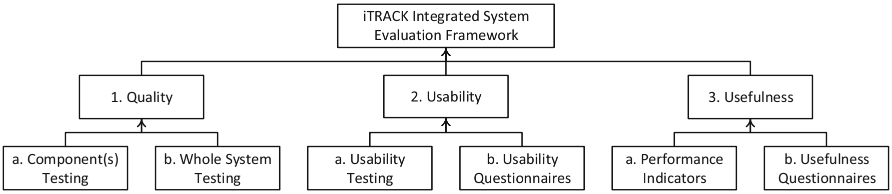

Fig. 26 iTRACK integrated system framework

The chapter also presents the iTRACK reporting and evaluation system that implements the proposed framework. A detailed look at graphical user interface design and functionalities was provided. The iTRACK reporting and evaluation system was developed with extensibility in mind. Extensibility is in terms of the system's capability of allowing its administrators to add new development indicators, performance indicators, surveys, SOPs, etc.

In April 2018, the iTRACK project conducted a simulated environment exercise. This exercise is an example of applying the iTRACK integrated system evaluation framework, as it was the first iTRACK system testing with users. During this exercise, participants tested the ready iTRACK system components. The participants were asked to complete specific tasks using the iTRACK system. The suitable usability and usefulness metrics and questionnaires proposed in this chapter were used during the exercise. The iTRACK reporting and evaluation system was used during the exercise. Some development results, like code coverage, were included in the iTRACK reporting and evaluation system as examples of the development indicators. The results of the questionnaires collected during the exercise were included in the iTRACK reporting and evaluation system as examples of users' surveys as well. Finally, some performance indicators were randomly generated for presentation purposes instead of actual results for privacy reasons. Results of the mission's SOPs/policies surveys were randomly generated and included in the system for presentation purposes as well.

For future work, the framework still requires more testing with the iTRACK system as well as other systems. For the iTRACK, the selected set of indicators and surveys was reviewed by the iTRACK partners as mentioned above. However, other systems will inevitably require other indicators. Our integrated framework and our reporting and evaluation system implementation facilitate extensibility in that sense by design. Accordingly, new development indicators, performance indicators, surveys, etc., could be easily added to the framework and the reporting and evaluation system based on the choices and needs of the target system. The reporting and evaluation system is available as an open source to facilitate further design changes or specific project adaptation requirements.

Acknowledgments This work is carried out as part of the iTRACK project, funded by the European Union's Horizon 2020 research and innovation program under grant agreement No 700510.

References

Black Box Testing, http://softwaretestingfundamentals.com/black-box-testing/. Last accessed 16 Mar 2018.

Boloix, G., & Robillard, P. N. (1995). A software system evaluation framework. *Computer, 28*, 17–26. https://doi.org/10.1109/2.476196

Borsci, S., Federici, S., & Lauriola, M. (2009). On the dimensionality of the system usability scale: A test of alternative measurement models. *Cognitive Processing, 10*, 193–197. https://doi.org/10.1007/s10339-009-0268-9

Brooke, J. (1996). SUS-A quick and dirty usability scale. In *Usability evaluation in industry* (Vol. 189, pp. 4–7). Taylor & Francis.

Corona, A. *What is usability and why should i care?*https://digitalconference.yale.edu/what-usability-and-why-should-i-care. Last accessed 14 Feb 2019.

Data, N. *13 essential software development metrics to ensure quality*, https://blog.usenotion.com/13-essential-software-development-metrics-to-ensure-quality-219cfc264ed1. Last accessed 19 Nov 2018.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly, 13*, 319–340. https://doi.org/10.2307/249008

Davis, F. D. (1993). User acceptance of information technology: System characteristics, user perceptions and behavioral impacts. *International Journal of Man-Machine Studies, 38*, 475–487. https://doi.org/10.1006/imms.1993.1022

EN_Tech_Direct. *Usability test report for mobile applications part ii: When the perfect combination of smart phone & music application*, http://www.technical-direct.com/en/usability-test-report-for-mobile-applications-part-ii-when-the-perfect-combination-of-smart-phone-music-application/. Last accessed 19 Feb 2018.

Gray Box Testing, http://softwaretestingfundamentals.com/gray-box-testing/. Last accessed 17 Mar 2018.

Humanitarian outcomes: Aid worker security database, https://aidworkersecurity.org/. Last accessed 14 Feb 2019.

IEEE Computer Society (Eds.). (1991). *IEEE standard glossary of software engineering terminology*. Institute of Electrical and Electronics Engineers.

Issues, https://docs.gitlab.com/11.6/ee/user/project/issues/index.html. Last accessed 19 Nov 2018.

ISTQB Glossary, http://glossary.istqb.org/. Last accessed 25 Apr 2018.

iTRACK Project – iTRACK, http://www.itrack-project.eu/. Last accessed 29 Jan 2018.

iTRACK: D8.1–Market analysis, https://www.itrack-project.eu/media/articles/files/iTRACK_WP8_D8.1_Market_Analysis_and_technology_foresight_FINAL.pdf. Last accessed 30 Jan 2022a.

iTRACK: D9.5–Final conference brief, https://www.itrack-project.eu/media/articles/files/D9.5%20%E2%80%93%20Final%20Conference%20Brief.pdf. Last accessed 30 Jan 2022b.

Jovanović, I. (2009). Software testing methods and techniques. *The IPSI BgD Transactions on Internet Research, 5,* 30–41. http://tir.ipsitransactions.org/2009/January/Paper%2006.pdf

Laugwitz, B., Schrepp, M., & Held, T. (2006). Konstruktion eines Fragebogens zur Messung der User Experience von Softwareprodukten. In H. M. Heinecke & H. Paul (Eds.), *Mensch und Computer 2006*. Oldenbourg Wissenschaftsverlag. https://doi.org/10.1524/9783486841749.125

Laugwitz, B., Held, T., & Schrepp, M. (2008). Construction and evaluation of a user experience questionnaire. In A. Holzinger (Ed.), *HCI and usability for education and work* (pp. 63–76). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-89350-9_6

McKay, J., Hamburg, M., & ISTQB Glossary Working Group (Eds.). (2016). *Standard glossary of terms used in software testing: Version 3.1*. International Software Testing Qualifications Board.

Müller, T., & Friedenberg, D. (2011). *ISTQB WG foundation level: ISTQB certified tester-foundation level syllabus*. International Software Testing Qualifications Board.

Nielsen, J. (1993). *Usability engineering*. Morgan Kaufmann.
Nielsen, J. *Usability 101: Introduction to usability*, https://www.nngroup.com/articles/usability-101-introduction-to-usability/. Last accessed 14 Mar 2018a.
Nielsen, J. *Severity ratings for usability problems: Article by Jakob Nielsen*, https://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/. Last accessed 1 Apr 2018b.
Noori, N. S., Wang, Y., Comes, T., Schwarz, P., & Lukosch, H. (2017). Behind the scenes of scenario-based training: Understanding scenario design and requirements in high-risk and uncertain environments. In T. Comes, F. Bénaben, C. Hanachi, M. Lauras, & A. Montarnal (Eds.), *Proceedings of the 14th ISCRAM conference* (pp. 948–959). Albi, France.
Perlman, G. *User interface usability evaluation with web-based questionnaires*, http://garyperlman.com/quest/. Last accessed 7 Mar 2018.
R, Python Duel As Top Analytics, Data Science software – KDnuggets 2016 Software Poll Results, https://www.kdnuggets.com/2016/06/r-python-top-analytics-data-mining-data-science-software.html. Last accessed 5 Dec 2017.
Sandvik, K. B., Jacobsen, K. L., & McDonald, S. M. (2017). Do no harm: A taxonomy of the challenges of humanitarian experimentation. *International Review of the Red Cross, 99*, 319–344.
Sauro, J. (2015). SUPR-Q: A comprehensive measure of the quality of the website user experience. *Journal of Usability Studies, 10*, 19.
Sauro, J. *Measuring usefulness*, https://measuringu.com/usefulness/. Last accessed 30 Mar 2018a.
Sauro, J. *10 essential usability metrics*, https://measuringu.com/essential-metrics/. Last accessed 8 Mar 2018b.
System usability scale. (2017). https://en.wikipedia.org/w/index.php?title=System_ usability_scale&oldid=805142158
Types of software testing for dummies, https://www.360logica.com/blog/types-of-software-testing-for-dummies/. Last accessed 6 Mar 2018.
Usability and user experience surveys, http://edutechwiki.unige.ch/en/Usability_and_user_experience_surveys. Last accessed 7 Mar 2018.
Van de Walle, B., & Comes, T. (2014). Risk accelerators in disasters. In *International conference on advanced information systems engineering* (pp. 12–23). Springer.
Van de Walle, B., & Comes, T. (2015). On the nature of information management in complex and natural disasters. *Procedia Engineering., 107*, 403–411.
W3Techs. *W3Techs – Extensive and reliable web technology surveys*, https://w3techs.com/. Last accessed 26 Nov 2016.
What is a White Box Testing? http://www.softwaretestingclass.com/white-box-testing/. Last accessed 17 Mar 2018a.
White Box Testing, http://softwaretestingfundamentals.com/white-box-testing/. Last accessed 16 Mar 2018b.