TU Delft

Delft University of Technology

# Evaluation Framework for the iTRACK Integrated System

Abdelgawad, Ahmed; Comes, Tina

**Publication date**
2019

**Document Version**
Final published version

**Published in**
ISCRAM 2019 - Proceedings

**Citation (APA)**
Abdelgawad, A., & Comes, T. (2019). Evaluation Framework for the iTRACK Integrated System. In Z. Franco, J. J. Gonzalez, & J. H. Canos (Eds.), *ISCRAM 2019 - Proceedings: 16th International Conference on Information Systems for Crisis Response and Management* (pp. 1160-1172). (Proceedings of the International ISCRAM Conference; Vol. 2019-May). ISCRAM.

**Important note**
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

**Copyright**
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

*This work is downloaded from Delft University of Technology.*
*For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.*

# Evaluation Framework for the iTRACK Integrated System

**Ahmed A. Abdelgawad**
Centre for Integrated Emergency Research,
University of Agder, Norway
ahmedg@uia.no

**Tina Comes**
Faculty Technology, Policy & Management,
TU Delft, Netherlands /
Centre for Integrated Emergency Research,
University of Agder, Norway
tina.comes@uia.no

## ABSTRACT

Evaluation and testing are major steps in the development of any information system, particularly if it is to be used in high-risk contexts such as conflicts. While thus far there are various approaches for testing against technology requirements; usability or usefulness, there is a lack of a comprehensive evaluation framework that combines the three elements. The lack of such a framework and commonly agreed standards constitutes a barrier for innovation, and at the same time imposes risks to responders if the technology is introduced without proper testing. This paper aims to close this gap. Based on a reviewing of evaluation methods and measurement metrics, we design a comprehensive evaluation framework including common code quality testing metrics, usability testing methods, subjective usefulness questionnaires, and performance indicators. We demonstrate our approach by using the example of an integrated system for the safety and security of humanitarian missions, and we highlight how our approach allows measuring the system's quality and usefulness.

## KEYWORDS

Evaluation Framework, Software Testing, Software Usability, Software Usefulness, Humanitarian Disaster.

## INTRODUCTION

Those who try to provide aid to the most vulnerable populations in the Middle East are increasingly risking their own lives and safety. According to the Aid Worker Security Database (Humanitarian Outcomes 2019), the number of humanitarian workers that fall victim to attacks continues to rise. Organizations in the field are confronted with mounting tensions as they seek to maintain access to populations in need. Thus, there is a new role for technology to support operations. Yet, these innovations and particularly the use of ICT in conflict can cause severe risks, ranging from privacy violations to threatening the lives and safety of those it is designed to protect.

Evaluation and testing is a major step in the development life-cycle of any software system, and it is a vital phase in quality assurance of the system (Jovanović 2006). Software system evaluation frameworks aim at assessing the system quality and sophistication from diverse viewpoints (Boloix and Robillard 1995). Nonetheless, there is no comprehensive evaluation framework that combines technology, functionality and usefulness tests for humanitarian conflict disasters. Such a test requires that the standards and problems of humanitarian innovation and experimentation are met (Sandvik et al. 2017) and the context of the problem is considered. In conflicts, a major challenge is dealing with sensitive information and organizational barriers to information sharing (Van de Walle and Comes 2015) and to evaluate risks as they emerge (Van de Walle and Comes 2014). The lack of such a framework and commonly agreed standards constitutes a major barrier for innovation, and at the same time may impose risks to responders if the technology is introduced without proper testing.

The goal of this paper is to compile a comprehensive evaluation framework for integrated systems in humanitarian conflicts, based on a review of evaluation standards and metrics. The framework should assist in measuring the **quality** and **usefulness** of a system, from the individual components' performance to the overall system. Software system **quality** is defined by the Institute of Electrical and Electronics Engineers IEEE 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). The International Software

Testing Qualifications Board ISTQB defines the quality in general as "the degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations" ("ISTQB Glossary" 2015), whereas it defines the 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 Glossary" 2015). In total, the **quality** of software is about meeting the **specified requirements** as well as **user satisfaction**. The former is achieved via testing the software system components individually or together, or the whole system against the requirements in terms of specifications, use cases, design documents, etc., while the latter is achieved via 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 2011). Based on Fred Davis' usefulness construct, system usefulness is about helping users in accomplishing job tasks quicker, improving job performance, productivity, and effectiveness, in addition to making the job easier to do in general. Figure 1 shows the pillars of our suggested integrated system evaluation framework.

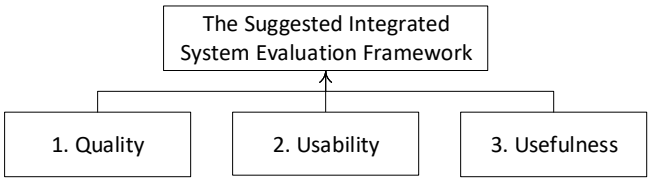


**Figure 1. Suggested Integrated System Framework**

The rest of this paper is organized as follows: the next section provides an overview of our methodology. The 'Results' section describes the evaluation methods reviewed and the evaluation methods selected for OUR case study in the context of the EU H2020 project iTRACK (https://www.itrack-project.eu). We conclude with a 'Summary' section.

## METHODOLOGY

To achieve the goal of this paper, we surveyed relevant sources for "software testing methods" and "technology usefulness instruments" to collect quality and the 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, the United Nations Development Program UNDP, etc. We used a variety of keywords like: "software testing", "software evaluation", "information system testing", "information system evaluation", "software quality", "information system quality", sometimes even just using "software" and searched for relevant material in results. This search was not fruitful. To mitigate the situation, we have used the same search keywords mentioned above and broadened our search circle to include sources like:

- 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), and
- Other material available on the Internet in general.

The results of this investigation were organized under the three pillars of our intended framework: quality, usability, and usefulness.

## RESULTS

The quality of software, as indicated previously, is about meeting the specified requirements as well as 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 performance indicators of an individual, a team, or an organization as a result of using the system, as well as subjectively by explicitly asking the users to provide their opinions on the usefulness of the system.

Our literature survey results were compiled under the first two main subsections of this section: '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. Grey-Box (Jovanović 2006). The software testing method will be decided based on the testers access to the internal structure of the software system under test (its source code): (1) **Black-Box Testing Method** (aka. Specifications-Based, or Behavioral testing) is a software testing method in which no need to access the source code of the tested item. ("Black Box Testing" 2017). (2) **White-Box Testing Method** (aka. 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) ("What is a White Box Testing?" 2012; "White Box Testing" 2010). (3) **Grey-Box Testing Method** is a combination of the Black-Box and White-Box software testing methods ("Gray Box Testing" 2010).

### *SOFTWARE TESTING LEVELS*

Software testing is being conducted on four different levels: Unit, Integration, Functional, and Acceptance.

**Unit Testing Level** (aka. Component, Module, Program, or Structural testing)[1] ("Types of Software Testing for Dummies" 2013) is a typical Whit-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" (Müller et al. 2011; "Types of Software Testing for Dummies" 2013). 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 (Müller et al. 2011; "Types of Software Testing for Dummies" 2013).

**System Testing Level** is a system testing that is concerned with the complete functionality and behavior of the whole system (Müller et al. 2011). The environment where this testing level is conducted should resemble the production environment to reduce the environment-specific failures (Müller et al. 2011). System testing level "may include tests based on risks and/or 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 et al. 2011). This testing level inspects both functional and non-functional requirements and could be conducted by an independent tester (Müller et al. 2011).

Figure 2 shows the relationship between the testing methods and the testing levels.

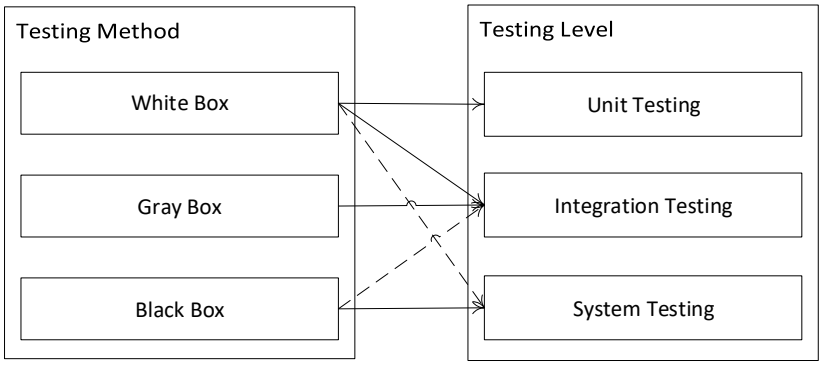


**Figure 2: Testing Levels and Methods**

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

### 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). Full documentation is available on the project website https://www.itrack-project.eu. Successive versions 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." (Judy McKay et al. 2016).

In the iTRACK development environment, an Integration testing for mainly the server-side components was carried out. In a simulation exercise in April 2018, another Integration testing including the client-side components was performed in addition to a System level testing to evaluate end-to-end workflows.[2] Before the final deployment, another System-level testing is planned. After deployment other metrics like the numbers and rates of bugs and issues reported, fixes; enhancements; improvements and new features released, issues reopened, etc.(for others, please check (Data 2017; "Issues" 2018)) can be used to indicate the quality of the iTRACK system.

## SOFTWARE USABILITY

Usability Testing Level (aka. 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 users to provide feedback before the system release to the market, which is called Beta testing (Müller et al. 2011; "Types of Software Testing for Dummies" 2013). "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 et al. 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 2012). Usability can be measured objectively and subjectively. Objectively by asking users to complete certain tasks and observe them. Subjectively by asking users to fill 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 2014). Observations are in terms of how users use the software. Then the developers will be able to know the correct needed features and understand the issues facing the users while using the software. Accordingly, developers can make improvements based on these observations.

### 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:

How easy it is for the user to learn using the system (EN_Tech_Direct 2012; Nielsen 2012). Learnability could be measured by:

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

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.

[2] For a full documentation of the test results, see: https://www.itrack-project.eu/media/articles/files/iTRACK_D7.3%20–%20Results%20of%20integrated%20system%20evaluation.pdf

- Efficiency:

How quickly it is for the user to accomplish tasks after learning the system (EN_Tech_Direct 2012; Nielsen 2012). Efficiency could be measured by finding the total time saved between the first and the last times doing a certain task.

- Effectiveness:

How well the user achieves her/his goals using the system (EN_Tech_Direct 2012). 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).

e.g.:

| | **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 2011).

e.g.:

| | **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 which the users encounter during the test. The moderator/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 2011).

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 1994).

- 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 2011). Afterward, the moderator/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 2011). Errors are detected via the observer's notes, for example, "user entered last name in the first name field" (Sauro 2011).

- 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 2011).

| | **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 2011). 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 2011).

| **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 2011). 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 standardized average of measures of effectiveness, efficiency of satisfaction and is typically composed of 3 metrics: **completion rates**, **task-level satisfaction** and **task time**" (Sauro 2011).

### *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 surveys" 2016).

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 2015). 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<br>○ Perceived ease of use. | (Davis 1989) |
| Software Usability Scale (SUS) | 5-points scale | 10 | ○ Usability and<br>○ Learnability. | (Borsci et al. 2009; Brooke 1996; Sauro 2015; "System usability scale" 2017) |
| Standardized User Experience Percentile Rank Questionnaire (SUPR-Q) | 11-points scale | 8 | ○ Usability,<br>○ Trust,<br>○ Appearance, and<br>○ Loyalty. | (Sauro 2015) |
| User Experience | 7-points scale | 26 | ○ Attractiveness, | (Laugwitz et al. 2006, |

| Questionnaire title | Questionnaire type | Number of items | Sub-scales/Construct | Reference |
|---|---|---|---|---|
| Questionnaire (UEQ) | | | ○ Perspicuity,<br>○ Efficiency,<br>○ Dependability,<br>○ Stimulation, and<br>○ Novelty. | 2008) |

### *THE ITRACK USABILITY AND USER EXPERIENCE TESTING*

The iTRACK system consists of several packages that have 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 iTRACK system component. Each feature to be tested is supplied 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 select usability task example, before doing this task, the participants should answer the following question:

**Before Task**

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

After finishing the task, the participants should log the time they took to complete this task 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 (S**uccess, **F**ailure and **P**artial **S**uccess**)** | | | | | | | | ○ S<br>○ PS<br>○ F |
| How difficult did you find this task? | Very easy | ○ | ○ | ○ | ○ | ○ | ○ | ○ Very difficult |
| - Please explain your choice: | | | | | | | | |

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

As indicated in the review, many questionnaires could be used to subjectively measure different constructs. Usually, users' time is limited and filled with several activities. To use this limited time efficiently, our team has selected only Davis' Perceived Usefulness and Ease of Use questionnaire, and UEQ questionnaires to be administered to the users as subjective usability measures. Davis' Perceived Usefulness and Ease of Use questionnaire is short and assesses the usefulness in addition to the ease-of-use, while UEQ gives more insights into the experience of the user. These questionnaires are to be administered to users for each of the iTRACK system components individually so that users can understand the text of 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, 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). Davis found that in predicting the actual system use, system usefulness is 1.5 times more important than ease of use or usability (Davis 1993; Sauro 2011).

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 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 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 a full phase or a whole mission as well. The same principle applies when the indicator is scaled up from an individual to a team or an organization. Figure 3 shows indicator measurement levels granularity that we have used while composing the performance indicators in the following subsections.

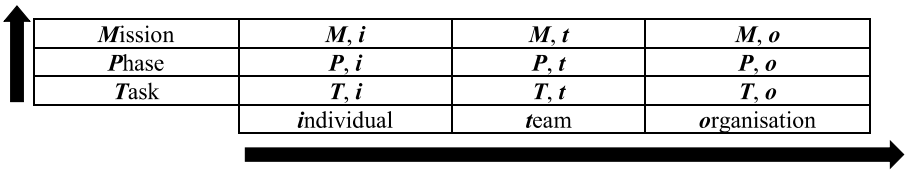


**Figure 3: Indicator Measurement Levels Granularity (Arrows Go towards Higher Levels of Aggregations)**

### *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:

$$ui_t = \frac{\sum_{i \in t} ui_i}{|t|}$$

$ui_t$ 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:

$$ui_o = \frac{\sum_{i \in o} ui_i}{|o|}$$

$ui_o$ 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 the average. Indicators, like replying to the message or performing an action because of the message content, could be insightful. Yet, 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 $\mathrm{PT}_1$, the time passed between receiving 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 $\mathrm{rmt}_{total}^{\mathrm{PT}_1} = \sum_{x\in \mathrm{PT}_1} \mathrm{rmt}_x^{\mathrm{PT}_1}$ and the average is $\mathrm{rmt}_{average}^{\mathrm{PT}_1} = \frac{\sum_{x\in \mathrm{PT}_1} \mathrm{rmt}_x^{\mathrm{PT}_1}}{|\{x: x\in \mathrm{PT}_1\}|}$.
- For an individual, the total reaction time to all messages during all tasks of the whole planning phase is $\mathrm{rmt}_{total}^{P} = \sum_{x\in P} \mathrm{rmt}_x^{P}$ and the average is $\mathrm{rmt}_{average}^{P} = \frac{\sum_{x\in P} \mathrm{rmt}_x^{P}}{|\{x: x\in P\}|}$. Similarly, $\mathrm{rmt}_{total}^{E}$ and $\mathrm{rmt}_{average}^{E}$, and $\mathrm{rmt}_{total}^{R}$ and $\mathrm{rmt}_{average}^{R}$ can be calculated.
- For an individual, the total reaction time to all messages during the whole mission is $\mathrm{rmt}_{total}^{mission} = \sum_{x\in M} \mathrm{rmt}_x^{M}$ or $\mathrm{rmt}_{total}^{P} + \mathrm{rmt}_{total}^{E} + \mathrm{rmt}_{total}^{R}$, and the average is $\mathrm{rmt}_{average}^{mission} = \frac{\sum_{x\in M} \mathrm{rmt}_x^{M}}{|\{x: x\in M\}|}$.

In case 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 respectively.

### *TIME-SAVING USING THE ITRACK SYSTEM*

This indicator requires two different entities (two individuals, two teams, or two organizations) executing 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 be also 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 $\mathrm{PT}_1$. 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 $\mathrm{PT}_1$. 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 $\mathrm{ts}_i^{E}$, and the individual's time saved during the response and recovery phase $\mathrm{ts}_i^{R}$. 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 $\mathrm{PT}_1$, 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 $\mathrm{PT}_1$, 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 that is related to specific tasks like loading trucks, completing deliveries, etc. can be separately considered as independent indicators on their own.

### *COST-SAVING USING THE ITRACK SYSTEM*

Cost could be calculated as the real cost of executing the task(s), phase(s), or mission(s) per an individual, team, or organization, which is difficult 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 indicator requires two entities (individual/team/organization) executing 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 be also 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 tasks related to specific tasks like loading trucks, completing deliveries, etc. can be separately considered as independent indicators on their own.

### *THE ITRACK USEFULNESS SUBJECTIVE EVALUATION*

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

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

To subjectively measure the usefulness of the iTRACK system or one 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 et al. 2011). Taking into consideration the limited time of the users testing the iTRACK system, another reason to select Davis' is that it is shorter than the others.

## SUMMARY

Evaluation and testing are major steps in the development of any software, but they are of particular importance if innovation is to be 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, a comprehensive evaluation framework that combines technology, functionality and usefulness tests does not exist. The paper developed metrics that helps in measuring the quality and usefulness by using the case of the iTRACK system, a tracking and monitoring system for humanitarian conflicts.

This paper reviewed the adequate evaluation methods and measurement metrics, to compile this comprehensive 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 that are 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 the system useful, the system should solve a problem they are facing, or fill a need or offer them something. System usefulness is about helping in accomplishing job tasks quicker, improving job performance, productivity, and effectiveness, in addition to making it easier to do the job. To measure the usefulness of the iTRACK system, we have suggested several performance indicators, as well as by subjectively recognizing the users' opinion about the usefulness of the system. Figure 4 shows the pillars and details of the suggested integrated system evaluation framework.

Finally, 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.

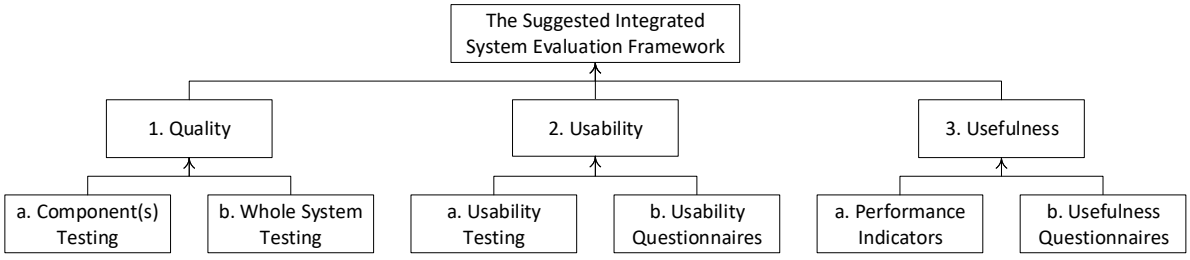


**Figure 4. The Detailed Suggested Integrated System Framework**

The iTRACK project conducted a simulated environment exercise in April 2018. 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, a set of participants tested the ready iTRACK system components. The participants were asked to complete certain tasks using the iTRACK system. The suitable usability and usefulness metrics and

questionnaires suggested in this paper were used during the exercise. As a final note, an independent reporting and evaluation software system applying these metrics for the iTRACK is under development.

## 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

Corona, A. (2014). "What is Usability and Why Should I Care?" *Yale Digital Conference*, <https://digitalconference.yale.edu/what-usability-and-why-should-i-care> (Feb. 14, 2019).

Humanitarian Outcomes. (2019). "Aid Worker Security Database." <https://aidworkersecurity.org/> (Feb. 14, 2019).

"Black Box Testing." (2017). *Software Testing Fundamentals*, <http://softwaretestingfundamentals.com/black-box-testing/> (Mar. 16, 2018).

Boloix, G., and Robillard, P. N. (1995). "A software system evaluation framework." *Computer*, 28(12), 17–26.

Borsci, S., Federici, S., and Lauriola, M. (2009). "On the dimensionality of the System Usability Scale: a test of alternative measurement models." *Cognitive Processing*, 10(3), 193–197.

Brooke, J. (1996). "SUS-A quick and dirty usability scale." *Usability evaluation in industry*, 189(194), 4–7.

Data, N. (2017). "13 Essential Software Development Metrics to Ensure Quality." *The School of Little Data*, <https://blog.usenotion.com/13-essential-software-development-metrics-to-ensure-quality-219cfc264ed1> (Nov. 19, 2018).

Davis, F. D. (1989). "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology." *MIS Quarterly*, 13(3), 319–340.

Davis, F. D. (1993). "User acceptance of information technology: system characteristics, user perceptions and behavioral impacts." *International Journal of Man-Machine Studies*, 38(3), 475–487.

EN_Tech_Direct. (2012). "Usability Test Report for Mobile Applications Part II: When the Perfect Combination of Smart Phone & Music Application." *Technical Direct*, <http://www.technical-direct.com/en/usability-test-report-for-mobile-applications-part-ii-when-the-perfect-combination-of-smart-phone-music-application/> (Feb. 19, 2018).

"Gray Box Testing." (2010). *Software Testing Fundamentals*, <http://softwaretestingfundamentals.com/gray-box-testing/> (Mar. 17, 2018).

IEEE Computer Society (Ed.). (1991). *IEEE standard glossary of software engineering terminology*. Institute of Electrical and Electronics Engineers, New York.

"Issues." (2018). *GitLab*, <https://docs.gitlab.com/11.6/ee/user/project/issues/index.html> (Nov. 19, 2018).

"ISTQB Glossary." (2015). *ISTQB Glossary*, <http://glossary.istqb.org/> (Apr. 25, 2018).

Jovanović, I. (2006). "Software testing methods and techniques." *The IPSI BgD Transactions on Internet Research*, 30.

Judy McKay, Matthias Hamburg, and ISTQB Glossary Working Group (Eds.). (2016). *Standard Glossary of Terms used in Software Testing: Version 3.1*. International Software Testing Qualifications Board.

Laugwitz, B., Held, T., and Schrepp, M. (2008). "Construction and Evaluation of a User Experience Questionnaire." *HCI and Usability for Education and Work*, A. Holzinger, ed., Springer Berlin Heidelberg, Berlin, Heidelberg, 63–76.

Laugwitz, B., Schrepp, M., and Held, T. (2006). "Konstruktion eines Fragebogens zur Messung der User Experience von Softwareprodukten." *Mensch und Computer 2006*, H. M. Heinecke and H. Paul, eds., OLDENBOURG WISSENSCHAFTSVERLAG, München.

Sauro, J. (2011). "Measuring Usefulness." *MeasuringU*, <https://measuringu.com/usefulness/> (Mar. 30, 2018).

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

Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann, Amsterdam, the Netherlands.

Nielsen, J. (2012). "Usability 101: Introduction to Usability." *Nielsen Norman Group*,

<https://www.nngroup.com/articles/usability-101-introduction-to-usability/> (Mar. 14, 2018).

Noori, N. S., Wang, Y., Comes, T., Schwarz, P., and Lukosch, H. (2017). "Behind the Scenes of Scenario-Based Training: Understanding Scenario Design and Requirements in High-Risk and Uncertain Environments." *Proceedings of the 14th ISCRAM Conference*, T. Comes, F. Bénaben, C. Hanachi, M. Lauras, and A. Montarnal, eds., Albi, France, 948–959.

Perlman, G. (2015). "User Interface Usability Evaluation with Web-Based Questionnaires." <http://garyperlman.com/quest/> (Mar. 7, 2018).

Sandvik, K. B., Jacobsen, K. L., and McDonald, S. M. (2017). "Do no harm: A taxonomy of the challenges of humanitarian experimentation." *International Review of the Red Cross*, 99(904), 319–344.

Sauro, J. (2011). "10 Essential Usability Metrics." *MeasuringU*, <https://measuringu.com/essential-metrics/> (Mar. 8, 2018).

Sauro, J. (2015). "SUPR-Q: A Comprehensive Measure of the Quality of the Website User Experience." 10(2), 19.

Nielsen, J. (1994). "Severity Ratings for Usability Problems: Article by Jakob Nielsen." *Nielsen Norman Group*, <https://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/> (Apr. 1, 2018).

"System usability scale." (2017). *Wikipedia*.

"Types of Software Testing for Dummies." (2013). *The Official 360logica Blog*, <https://www.360logica.com/blog/types-of-software-testing-for-dummies/> (Mar. 6, 2018).

"Usability and user experience surveys." (2016). *EduTech Wiki*, <http://edutechwiki.unige.ch/en/Usability_and_user_experience_surveys> (Mar. 7, 2018).

Van de Walle, B., and Comes, T. (2014). "Risk accelerators in disasters." *International Conference on Advanced Information Systems Engineering*, Springer, Cham, Switzerland, 12–23.

Van de Walle, B., and Comes, T. (2015). "On the nature of information management in complex and natural disasters." *Procedia Engineering*, 107, 403–411.

"What is a White Box Testing?" (2012). *Software Testing Class*, <http://www.softwaretestingclass.com/white-box-testing/> (Mar. 17, 2018).

"White Box Testing." (2010). *Software Testing Fundamentals*, <http://softwaretestingfundamentals.com/white-box-testing/> (Mar. 16, 2018).