



CHALMERS
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG

Gamification of Traceability Management Tools

Master's thesis in Software Engineering

Carl-Oscar Persson & Emil Sundklev

MASTER'S THESIS 2018

Gamification of Traceability Management Tools

Carl-Oscar Persson, Emil Sundklev



Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2018

© Carl-Oscar Persson, 2018.

© Emil Sundklev, 2018.

Supervisor: Grisha Liebel and Salome Maro, Department of CSE

Examiner: Regina Hebig, Department of CSE

Master's Thesis 2018

Department of Computer Science and Engineering

Chalmers University of Technology and University of Gothenburg

SE-412 96 Gothenburg

Telephone +46 31 772 1000

Department of Computer Science and Engineering
Gothenburg, Sweden 2018

Abstract

Background: Traceability is a desired quality of software. However, successfully implementing it can be expensive. The existing traceability management tools meant to simplify the process are often considered unengaging, further complicating the task of creating and maintaining a high standard of traceable systems.

Objective: In this study we examine whether gamification features can be used to extend the verification aspects of traceability management tools. Our intent is to render the existing tool Capra more engaging and to examine to which extent it is affected.

Method: Our methodology is built around a Design Science Research framework and incorporates various data collection methods. To produce viable data sets to analyze we have conducted an experiment along with three different surveys. In the aforementioned experiment, two groups of 12 participants are tasked to verify the trace links of the same trace matrix. The first group was assigned the traceability management tool Capra extended with a level and a badge feature, while the control group used Capra with no additional features.

Results: The results showed that there was no significant difference between the groups' speed and correctness. The level and badge features were perceived positively by the majority of the participants while some pitfalls and improvements were pointed out. Upon testing the results they proved mostly to be insignificant, with the exception of the user's perceived enjoyment, as such further research would be required in order to confirm many of the indications presented in this study.

Conclusion: In conclusion the study indicates the need for further research into the field as the result raises several questions regarding traceability, gamification and their interaction. More extensive studies need to be conducted to investigate these indications with larger sample sizes, different gamification features and alternative traceability management tools.

Keywords: Software engineering, gamification, traceability, traceability management tool.

Acknowledgements

The authors of this study would like to extend their gratitude to our supervisors, Grisca Liebel and Salome Maro who were invaluable to our work process. We would also like to thank all who participated in the experiment along with those who offered support during this undertaking.

Carl-Oscar Persson & Emil Sundklev, Gothenburg, June 2018

Contents

List of Figures	xiii
List of Tables	xv
1 Introduction	1
2 Theory	3
2.1 Traceability	3
2.1.1 Traceability terminology	3
2.1.1.1 Trace	3
2.1.1.2 Trace artifacts	4
2.1.1.3 Trace link	4
2.1.2 Traceability processes	4
2.1.2.1 Creating Traces	4
2.1.2.2 Vetting traces	5
2.1.2.3 Maintaining traces	5
2.1.2.4 Domain background	5
2.2 Gamification	6
2.2.1 Reported benefits of gamification	7
2.2.1.1 In an educational setting	7
2.2.1.2 In an industrial production setting	7
2.2.2 Kaleidoscope	8
3 Methods	9
3.1 Design science research	9
3.2 Awareness of the problem	10
3.2.0.1 Lack of motivation	10
3.2.1 Identifying viable gamification elements	11
3.3 Suggestions	12
3.3.1 Suggested game elements to implement	12
3.3.1.1 Points	12
3.3.1.2 Progress bar	12
3.3.1.3 Levels	13
3.3.1.4 Badges	13
3.3.1.5 Leader board	14
3.4 Development	14

3.4.1	Gamified system: Capra	15
3.5	Evaluation	15
3.5.1	Experiment design	15
3.5.1.1	Participants	16
3.5.1.2	Experiment equipment	16
3.5.1.3	Traceable dataset: Medfleet	16
3.5.1.4	Capra/MedFleet specific candidate links	17
3.5.1.5	Introduction to Capra and its features	17
3.5.1.6	Execution of the experiment	17
3.5.2	Data collection and analysis	18
3.5.2.1	Automatic data collection	18
3.5.2.2	Survey	18
3.5.2.3	System usability scale	19
3.6	Ethical concerns with gamification	19
3.7	Validity threats	20
3.7.1	Construct validity	20
3.7.1.1	Initial pilot survey misinterpretations	20
3.7.1.2	Experiment survey misinterpretations	20
3.7.1.3	Vetting misinterpretations	20
3.7.1.4	MedFleet domain knowledge	20
3.7.2	Internal validity	21
3.7.2.1	Personal preferences and personality conflicts	21
3.7.2.2	Inaccurate data collection	21
3.7.2.3	Poor UI design	22
3.7.3	External validity	22
3.7.3.1	Generalizing the results	22
3.7.4	Reliability	23
3.7.5	Unforeseen events during the experiment.	23
3.7.5.1	Previous experience	23
3.7.5.2	Loss of video	23
3.7.5.3	Loss of JSON	23
3.7.5.4	Varying group sizes	23
4	Results	25
4.1	Implementation	25
4.1.1	Modifications to existing classes	26
4.1.2	The Added Files	26
4.1.3	Libraries Used	27
4.2	Gamification features	27
4.3	Experiment results	28
4.3.1	Background	29
4.3.2	Post-experiment questions	31
4.3.2.1	General questions about the experiment	31
4.3.2.2	System usability scale	32
4.3.2.3	Gamification group on levels and badges	34
4.3.2.4	Control group on levels and badges	35

4.3.2.5	Answering the research questions	36
4.3.3	Vetting task results	36
5	Discussion	39
5.1	The effects of gamification on traceability management tools	39
5.2	Cost of speed and competition	39
5.3	Inaccurate acceptations and accurate rejections skew	40
5.4	Are traceability management tools worth using?	42
5.5	SUS result	42
5.6	Insignificant result	42
6	Conclusion	45
	Bibliography	47
A	Appendix 1 - pilot survey	I
B	Appendix 2 - GG survey	XI
C	Appendix 3 - CG survey	XXV
D	Appendix 4 - SUS statements	XXXIX
E	Appendix 5 - Experiment Instructions	XLI

List of Figures

2.1	The Kaleidoscope of Effective Gamification [13]	8
3.1	Design Science Research Process Model (DSR Cycle) [32]	10
4.1	Level feature	28
4.2	Badge feature	28
4.3	Q1: Software development experience	29
4.4	Q2: Experience in using Eclipse	30
4.5	Q3: Experience with software traceability	30
4.6	Q4: Experience with systems similar to MedFleet	32
4.7	Q5: Understanding of the MedFleet system	33
4.8	Q6: Enjoyment of the vetting process	33
4.9	Q7: Motivation to complete the task	33

List of Tables

4.1	SUS score	34
4.2	GG responses regarding the <i>levels</i> and <i>badges</i> feature	35
4.3	CG responses regarding the <i>levels</i> and <i>badges</i> feature	36
4.4	GG results from the vetting task	37
4.5	CG results from the vetting task	37

1

Introduction

In the software industry, traceability is an often desired quality of software, as it aids both the developer and manager in tracing software artifacts to documentation. It can also be required by certain certificates, such as a level 3 certificate from the Capability Maturity Model Integration (CMMI) and there are also certain agencies such as the European Aviation Safety Agency [2] who introduces regulations for safety critical systems requiring traceability. Despite being a desired quality and also sometimes required, traceability management tools are considered unengaging [2] and poorly justified trace links could potentially lead to more problems than solutions than having no trace links at all [3].

When attempting to engage users, the concept of gamification has shown to have a positive motivational effect on its users [12, 15, 10]. In some fields gamification has been shown to reduce the rates of failure and assist in the learning process[19].

Traceability management tools are often unengaging and gamification can have a positive effect on its users in such cases, suggesting a potential beneficial interaction between the two [14]. Because of this promising interaction between gamification and traceability, there is a need to study this concept in detail. However, to our knowledge no previous study exists on this specific topic, giving it further value to the research and industrial community.

Therefore, the purpose of this study was to examine how gamification principles could be applied to the vetting process of a traceability management tool, and how it could affect some of the issues it currently faces. Towards this purpose, this study aims to answer the following research questions:

- **RQ1:** Which gamification elements can be used for extending traceability management tools?
- **RQ2:** To what extent can gamification elements increase motivation for traceability link vetting?
- **RQ3:** What are the disadvantages of adding gamification elements to traceability management tools?

The study consisted of one iteration of the design science research methodology, alongside an experiment involving the traceability management tool Capra and a set of compatible trace matrices. The experiment was carried out with two groups of 12 participants each. The first group used a gamified version of Capra with the extension of a level and a badge feature, while the second control group used the

standard version of Capra.

The results show that that the gamification elements had little impact on the results of the vetting task, while showing it had the potential to engage and motivate the users of such a task.

The findings of this study act as a basis for future research focused on the interaction between gamification and traceability. In the long-term our findings could help software companies to determine whether implementing gamification into their own traceability systems is to be considered.

This study is divided into 4 chapters, the first *Theory* chapter provides domain background on traceability and gamification. The second chapter, *Methods* describes the process of design science research and how the study was carried out in the different phases of it. The *Result* chapter presents the results from the experiment and the meaning of the results are discussed in the *Conclusion* chapter.

2

Theory

Within the theory section, we describe the theoretical background for our study, the relevant literature we have compiled and how it relates to our study. We also describe the various traceability artifacts, concepts of gamification and guidelines for successful gamification.

2.1 Traceability

Traceability can be defined as "The potential for traces to be established and used" [7]. Within the software industry traceability entails the ability to trace artifacts such as requirements, source code and the tests of a system. The result of traceability is called a *trace*, which consists of 3 elements, a *source artifact*, a *trace link* and a *target artifact* [7]. An example of a trace could be a requirement (source artifact), the source code which was developed for the sake of fulfilling the requirement (target artifact) and the association between these two artifacts (trace link). One of the reasons for traceability being a desired quality of software development, is that being able to use such a trace and going from a requirement to its implemented source code or vice versa, aids both the developers and managers in comprehending and maintaining a system.

2.1.1 Traceability terminology

This section provides descriptions on some of the terminology used throughout this paper when discussing traceability. The terminology presented below is based on the work presented in *Software and Systems Traceability* written by Cleland-Huang et al. [7]. Further details can be found there.

2.1.1.1 Trace

A trace can be interpreted in two different ways depending on the context. The first way is the combination of the three elements described earlier in this section which is a *source artifact*, a *target artifact* and a *trace link*. Together they form a *trace*. The second way of interpreting a *trace* is to treat it as a verb, which is to pursue a *trace link* between a *source artifact* to a *target artifact* or if required the other way around.

2.1.1.2 Trace artifacts

Trace artifacts can be any type of unit of data within a software system or documents related to it that can be traced. It can be any unit ranging from a package, source file, class or operation. It can also include individual requirements as well as an entire requirements document.

Trace artifact - This could be anything from a UML diagram to a Java class operation, either individual ones or a group of them and are candidates to be either a *source artifact* or a *target artifact*

Trace artifact type - Any trace artifacts that can be considered to be of the same type, like requirements, source files or tests.

Source artifact - This is one of the three elements of a *trace* where the *trace link* starts from.

Target artifact - This is one of the three elements of a *trace* where the *trace link* ends at.

2.1.1.3 Trace link

A trace link is one of the three elements of a *trace* and is the connection between the *source artifact* and the *target artifact*. A trace link can also have a specified direction, which can be the *primary trace link direction*, the *reverse trace link direction* or both of them. The direction is usually mentioned when you have the intention to traverse a trace in order to find what you are looking for. For example, if you want to know which source code (source artifact) a test (target artifact) is testing, you would have to go in the reverse direction of the trace link.

Primary trace link direction - This direction indicates that you traverse the trace starting from the *source artifact* and end at the *target artifact*

Reverse trace link direction - This direction indicates that you traverse the trace starting from the *target artifact* and end at the *source artifact*

Bidirectional trace link - This is an indication that a trace link can be of both a *primary trace link direction* and a *reverse trace link direction*.

Candidate trace link - A trace link between two artifacts which has not yet been validated to be correct by one or several peers.

2.1.2 Traceability processes

A traceability process undertakes a specific task to perform, such as the task of creating a trace link between two artifacts, vetting a trace link which has been created by someone else or maintaining old traces which are out of date.

2.1.2.1 Creating Traces

This process can be considered to be the most basic process of traceability, as you cannot have any traces if you do not create them. Creating a trace is usually done manually by a developer of a specific system, but it can also be done automatically, Traces can be generated automatically by detecting key words in a requirements document and making a connection to a source file by looking at the definitions of

an operation or a variable name including the key words or synonyms to the key words. Automatically generating traces is a research topic on its own, as it would save both time and money in the long run to implement it [8]. More research is required on this specific topic in order to increase the accuracy and correctness of the *candidate trace links*. So in the mean time there is still a need to manually validate the traces [6].

2.1.2.2 Vetting traces

Vetting traces is the process of peer validation of any trace created, to account for human error among other potential issues. It works as follows; you are presented with one or several *candidate trace links* which needs to be validated by either accepting or rejecting them. Trace vetting is necessary for both manual and generated traces, since neither method is without fault.

2.1.2.3 Maintaining traces

A software system in use is constantly evolving, and such any trace created for it needs to be maintained in order for it to actually represent the current system in use. This process includes updating an existing trace by changing the *source artifact*, the *target artifact* or the *trace link*, removing the trace entirely or simply validating whether the current trace is still up to date.

2.1.2.4 Domain background

One of the main issues surrounding traceability is that the people who are supposed to create or maintain traces have little to no motivation in doing so [2, 14]. There are several reasons for this, one of them being that the people who programmed a piece of code would normally be the one creating the necessary traces connected to that piece of code. In the end this person would not really benefit from the traces that he created, since the information from the trace is to him already known. The task of creating the trace could therefore be considered unnecessary by the people creating them. Other common issues surrounding traceability are that it is time-consuming, tedious and generally not considered to be an engaging task [3, 4].

There are several reasons for why you would want to incorporate traceability as part of your software development process. It provides means of being able to perform coverage analyses, navigating between artifacts and provides a means to justify how and why your system is built in a certain way. Some companies use traceability as a way of maturing. For instance, to get a level 3 certificate for Capability Maturity Model Integration (CMMI), integrating traceability is a necessity. If working on a safety-critical system, there are agencies such as the European Aviation Safety Agency [2] and the USA Federal Aviation Authority [1] that have introduced regulations which force you to apply traceability to the development process.

Cleland-Huang et al. [1] pinpoint areas of improvement where resources and effort need to be placed in order to achieve what the authors called "the grand challenge

of ubiquitous traceability” [1]. Solving this challenge would mean that traceability is always present in the software engineering process, but with more or less zero effort. What they present in the study are seven areas of research and each of which has its own set of directions of where research needs to be focused on. When these research areas have been addressed, ubiquitous traceability can be achieved. One of the research areas presented is trace integrity, which concerns the correctness and quality of any trace link which has been created and how to validate that quality. One of the directions concerning trace integrity is ”improving integrity through human feedback” [1], which is the focus of this study. One way of addressing this is by developing and improving tools which helps the analysts who undertake the task of link vetting. The results of this study will contribute towards achieving ubiquitous traceability.

Cuddeback et al. [6], present in their study the results collected from a few of their previous studies which have been looking at the accuracy of vetting candidate trace links. Their study focuses on looking at how the analyst, the person who undertakes the vetting task, performs while vetting candidate links both manually and with the use of tools. The authors found that ”human analyst fallibility in trace validation tasks is both unavoidable and predictable” [6], and they presented four courses of action which they deemed to be potential solutions, given the current state of analysts behaviour and performance.

The first course of action presented was to remove the analyst completely, which would only be viable if there is an automatic generation of candidate links including perfect accuracy for each link, but this is not yet available and will most likely be the case for a long time.

The second course of action considered is to place a type of firewall security on an analyst, preventing them from rejecting or accepting a candidate link, for example if the analyst has a history of low accuracy. At the same time the authors mention that this would most likely be considered a toxic behaviour and probably not a suitable way of motivating an analyst to improve his vetting capabilities.

The third course of action is to train your analysts to become better at making decisions when vetting candidate links, which is a good approach in the long run, but a short term approach is still necessary.

The final course of action which is of most relevance for this study, is to embrace the analyst. This means that the analyst must be considered to be a core part of the process, and that mistakes are bound to happen. What you want is to find new ways to improve the process in a way that enables the analyst to produce better results. The authors end their study by emphasizing the fact that more research needs to be conducted on this topic and to conduct isolated experiments which look at one variable at a time [6], which is the aim of this study.

2.2 Gamification

Gamification can be described as “the use of game design elements in a non-game context“ [11]. Applying gamification is a potential solution for when you have the intention of increasing the motivation or user-activity of either your customers or

employees. The most usual application of gamification is to add elements such as points, levels, leader boards, achievements and badges [21, 12, 10].

Gamification has been implemented in many different fields and environments, software engineering included. This is shown in the paper *Gamification in software engineering—A systematic mapping* by Pedreira et al. [12], and it is part of a project called GOAL (Gamification on Application Lifesycle). It was carried out as a systematic mapping in order to understand the current status of gamification within the academic domain of software engineering. The authors were looking for process areas in which gamification principles were applied, e.g. development or testing. Areas mapped in this study were gamification concepts involving reward system, e.g. badges or point. From the results the authors came to the conclusion that “the existing research on gamification applied to SE is very preliminary or even immature” [12] and also stated that the effects gamification has on SE needs further research. The authors also points out that there is a big gap when considering the areas of software engineering where gamification has been applied to. Traceability is one area which was not mentioned in the study and which is lacking research for applying gamification to it and what potential effect it could have.

2.2.1 Reported benefits of gamification

Several benefits of using gamification have been reported in academic literature. In the following, we cover them based on their area of application.

2.2.1.1 In an educational setting

Gamification sees different applications across different domains, for example within an educational setting. Enveloping the material to be learned into a narrative context can increase student motivation and engagement [16]. Competitive elements such as leaderboards can be helpful, and badges provide tangible indications of one’s progress within the course [18, 17].

Another example showed that students who recieved feedback on their work through a competitive lens had lower rates of failures and learned more [19]. While there are some clear benefits, it is not without its downsides, one study showed that students participating in a gamified course attended less class activities and had worse results on written assignments [20].

2.2.1.2 In an industrial production setting

Work done by humans is quite often prone to errors and varying speeds. In an industrial production setting this can involve potential failures at many different steps of the process.

A study has shown that in a production setting, gamification applied to the work process has been able to improve its speed whilst decreasing the accuracy of the operator. It should be noted however that the accuracy only deteriorates if no



Figure 2.1: The Kaleidoscope of Effective Gamification [13]

quality feedback is shown to the operator. Where as if the operator gets consistent feedback, the accuracy remains the same [30].

2.2.2 Kaleidoscope

While gamification finds frequent use within software engineering, there are few coherent strategies or guidelines for creating effective gamified systems. One of the few guidelines is the "Kaleidoscope of Effective Gamification" [13]. The Kaleidoscope draws inspiration from existing Game Design elements such as the mechanics dynamics aesthetics (MDA) framework and the motivational model of video game engagement [13].

Kaleidoscope explains an effective gamified system as five separate layers, which in turn identify the extrinsic and intrinsic motivations of the user, the challenges the user has to overcome and the gameplay experiences which it would lead towards. It also describes the game design patterns and mechanics involved and how it links to the perceived "fun" of the user [13].

These guidelines were useful as we designed the gamification elements, they allowed us to relate our ideas to an already established framework, instead of designing the gamification elements ad-hoc.

3

Methods

This chapter describes each phase of the design science research approach and what was done and learned from each phase. It also includes the experiment setup, ethical concerns with gamification and finally the identified validity threats of the study.

3.1 Design science research

Design science research (DSR) can be described as designing and creating artifacts in order to solve existing and known problems in a given field [31, 32, 33]. Since this study is concerned with the software engineering field, an artifact which is being created during the process of DSR could for example be "algorithms, human/computer interfaces, and system design methodologies or languages" [32]. During the process of DSR, a study goes through five different phases in an iteration, an iteration which can be cycled through as many times as deemed necessary for your research. The phases of one iteration are called *Awareness of the Problem*, *Suggestions*, *Development*, *Evaluation* and finally *Conclusion*. These phases are presented in Figure 3.1 including what type of output each phase is expected to produce. The model as a whole represents one iteration for the DSR process, from start to end.

For this study we have chosen to apply the design science research approach since we want to investigate how different gamification elements can improve traceability. DSR allows us to create and evaluate different gamification elements in the form of software artifacts. The identified problem surrounding traceability and in need of a solution is the lack of motivation or enjoyment in completing a task associated with traceability. The proposed solution to this problem is a gamified version of the traceability management tool Capra. For evaluating our implemented game elements, we have conducted an experiment which consisted of two groups of 12 participants in each group. One of the groups used the standard version of Capra and the other group used the gamified version.

In the following sections we describe how we worked during each phase of the DSR cycle and present the knowledge we were able to gather during these phases. Our study consists in total of one iteration and the end result of this iteration present an increased understanding of how gamification can have an impact on traceability and what future iterations should consider to improve even more.

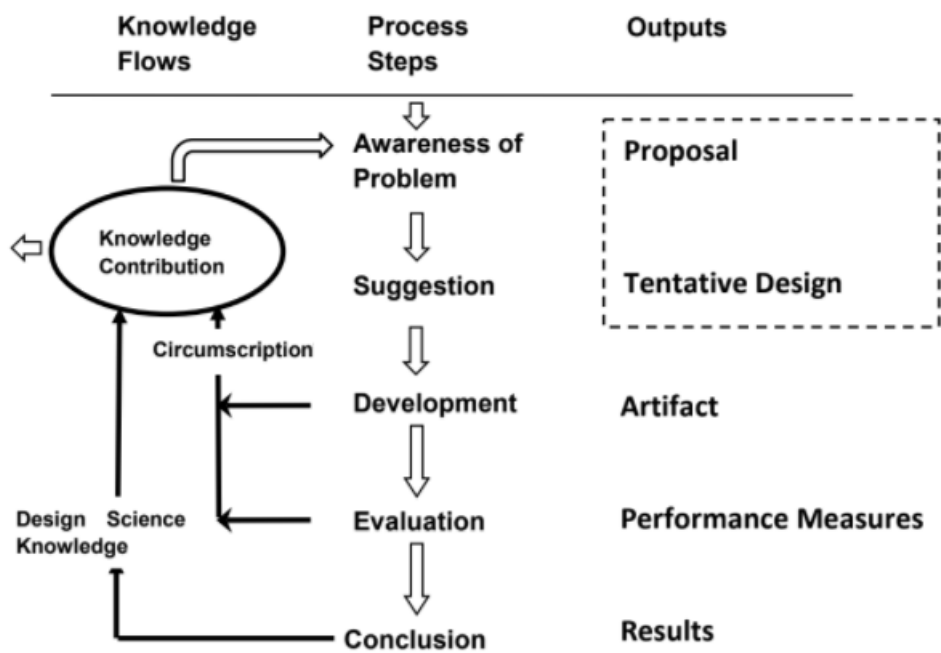


Figure 3.1: Design Science Research Process Model (DSR Cycle) [32]

3.2 Awareness of the problem

Our study delves into several of the issues faced by modern traceability management tools, but focuses predominantly on examining to what extent gamification can affect these issues.

3.2.0.1 Lack of motivation

Motivation is commonly cited as a major issue for traceability management tools [2], it is intriguing then that gamification can sometimes be applied to a system to increase motivation in the userbase [9, 10]. Our study therefore strives to determine to what extent gamification can affect a users motivation for using a traceability management tool.

Many studies which are focused on gamification consider how and if a gamified implementation has had an impact on intrinsic and extrinsic motivation [22, 21, 15]. Intrinsic motivation can be described as being motivated to perform a task simply by the fact that you enjoy doing it, while extrinsic motivation can be described as completing a task if you have some sort of incentive after completing it [36].

When attempting to motivate people, it is generally considered best practice to aim at increasing intrinsic motivation, since being motivated to do something because you enjoy it is more sought after than being motivated by extrinsic rewards [36].

It would have been ideal to examine the impact gamification would have had on

intrinsic and extrinsic motivation, but given that the experiment was done in one session per participant it would be near impossible to conclude whether the gamification features would have had any impact on either one. Achieving such results would require a longer study which would need to include follow up interviews with users of a gamified system in an industrial setting. This is why the scope of this study is focused on motivation in general and not specifically for intrinsic or extrinsic motivation.

3.2.1 Identifying viable gamification elements

Whilst designing the gamification elements we sent out a survey to 14 participants of a previous study involving Capra. The purpose of this survey was two-fold, first to gather their own opinions on Capra, particularly how it relates to gamification aspects, such as if the system was enjoyable to use. The second focus of the survey was to elicit opinions regarding potential gamification elements to be implemented and whether our own suggestions seemed like beneficial additions. This allowed us to design the gamification elements with more perspectives available to us, potentially revealing flaws and design decisions previously unconsidered. In the survey we inquired about the following four game elements: *progress bar*, *levels*, *leader board* and *badges*.

In the survey we had the same five questions for each of the four game elements we asked about. The questions we asked the subjects were as follows:

1. Do you believe this functionality would make the task of verifying traceability links more satisfying?
2. Do you believe this functionality would help in decreasing the time spent on each link?
3. Do you believe this functionality would make you focus less on verifying the links correctly and focus more on verifying as many as possible?
4. Given this functionality, do you believe you would skip the more advanced links and go for easier ones that could potentially fill up the bar quicker?
5. Do you think a progress bar could have any negative effects? If yes, which ones?

Note that some of the text in question 4 and 5 would slightly vary depending on the game element in question. We asked these questions in order to get an overview of how the game elements were perceived and if anyone would think it would be helpful or distracting when vetting candidate links. Available answers for each question were 1 (Strongly disagree) to 5 (Strongly agree), with 3 being a neutral activity. For each game element we asked about, there was an optional text field in which they could write any additional thoughts they had or if they wanted to clarify their responses in any way. The survey had a total of 11 responses and all of the responses can be found in Appendix A

3.3 Suggestions

In this phase, we used the knowledge gathered from the previous phase in order to come up with concrete suggestions on how to solve the problem.

What we learned from the previous phase, was that users assigned traceability tasks are usually not very motivated to complete it. On top of that we learned that one possible solution to this problem would be to apply gamification.

3.3.1 Suggested game elements to implement

In this section we describe the different game elements that were considered for the gamified version of Capra, our reasoning behind it and what the responses from our survey indicated about each game element. With the exception of the progress bar, they are all commonly used gamification elements [10], thus we deemed them viable candidates to be implemented.

3.3.1.1 Points

Points were initially a gamification feature we sought to include. But the idea was later turned down, for a number of reasons. One of these was that points allow the user to gain an understanding of their current progress, something which the proposed levels and progress bar gamification elements already fulfilled. Therefore, we decided to not ask about points in the survey we sent out. Given the positive response for levels in the survey, levels were chosen over points since their functionality overlapped.

3.3.1.2 Progress bar

The progress bar is not a traditional gamification element, but the progress bar we had in mind is more accurately described as a re-interpretation of the levels system. The progress bar fills up with each vetted link, translating onto the UI how many trace link candidates that remain after each verification, but resets at the end of the session. While vetting large systems, there might be enough candidate links that they cannot all be verified during a single work day. This can make the task seem insurmountable. The progress bar attempts to solve this by setting a reasonable limit as to how many traceability links are required to fill the bar, it also resets upon the next session and the progress bar can be filled anew. In essence, it is a levels system that resets at the end of each work session.

The majority of the responses for question 1 were positive about having a progress bar added to the system and one of the comments said "It is usually nice to see progress being made as you work". The responses for questions 2-4 were quite split and did not indicate a significant majority for the progress bar having a direct positive or negative impact on the system when thinking about time, correctness and "slacking" (Question 4).

The progress bar was a concept we considered, but in the end decided to not include because we could not justify its inclusion, since it has no proven effectiveness in the studies examined [10, 27, 29].

3.3.1.3 Levels

Levels are milestones reached by a user, and in the context of this study vetting a certain amount of links would be considered a milestone. A users level should reflect the users experience in the given game. Each level requires an increasing amount of vetted links to be reached, with the intention making the gaps of levels less noticeable between veteran and new users of the system. The new user will increase in levels quickly to begin with, but the progress will gradually slow down as you reach higher levels. This is more or less a universal way of implementing levels in any given game. That is because an initially slow pace of level progression might discourage new users to continue playing when comparing to others. But as the experiment only takes place during a single session and with no ability to compare your progress against others there is little point in varying the rate at which you gain levels.

Similarly to the progress bar, levels were positively received by the majority of the responses for question 1. Responses for question 2 and 4 were also quite split, but for question 3 the responses indicated that people might focus more on vetting more links in total instead of focusing on vetting them correctly.

Given that levels were well received by the responses in the survey and that it has been shown to have a positive effect in other studies [21, 23, 24], we decided to include levels in the gamified version of Capra. We are aware about the potential that people might be discouraged to verify links correctly in order to level up faster, or attain badges at a quicker rate. Whether the gamification elements promote undesired user input was determined in the post-experiment survey.

3.3.1.4 Badges

Badges are awarded upon specific actions or conditions set within the system, such as reaching a set level or in the context of this study by rejecting your first link. These are intended both as a motivational method as well training wheels to guide the user through the functionality of the system. The conditions required to receive a badge can be customized, such that it is designed to be acquired once the user has mastered certain functionality of the system, forming a rudimentary tutorial system.

Badges seemed to have the most favorable responses compared to the other game elements with only one negative response when asked if the badges would make the system more satisfying to use. The majority thought it would make the vetting task more satisfying and that it would have little to no effect to the time spent on each link. There were split responses when asking about focusing on the correctness of each link. Finally, the majority of the responses indicated that people would most

likely not skip the more advanced links in favor of the easier ones.

Since the badges were the game element which were favored in the survey and has shown to have a positive effect on user behaviour in other studies [25, 26], we decided to include it as well.

3.3.1.5 Leader board

One of the gamified elements that we decided to include in the survey was the leader board. Our thoughts on adding a leader board to Capra would simply display the top five players, based on their level. Our reasons for potentially including the leader board is to reach out the participants who possibly have a competitive personality. This has the potential to motivate them to complete their task by giving them the incentive of being placed on the leader board [23, 24] and also adding the social factor of comparing results with coworkers, which have been shown to improve the process further [28]. Any of the participants who would not have been interested in reaching the top of the leader board would hopefully not be affected by the presence of it.

The leader board would be shown to every individual participant as they start their task, and be filled with five fake accounts, each with a corresponding level. Ideally we would have liked to compare the results of each participant to one another, but this would become troublesome for the first couple of participants, as there would not be any previous participants to compare them against. With the use of fake accounts and levels, we can give each participant the same scenario and so their experience would be similar as well. We would have to design the levels of each fake account so that more or less every participant would be able to reach the bottom of the leader board, and some of them might even reach the top if they are efficient enough.

The leader board was the game element which had the most split responses when asking about it making the vetting task more satisfying, the amount of people that thought the task would become more satisfying, roughly the same amount of people thought it would make it less satisfying. The responses indicated that it would have little to no effect on time spent on each link. The majority of people responded that they would focus less on correctness and also skipping the more difficult links in favor of the easier ones.

Due to not being able to guarantee that our users would have been competitive minded, we choose to exclude the leader board from the gamified system, since otherwise we risk giving the non-competitive minded a feature they have no need of or that could affect them detrimentally.

3.4 Development

In this section we describe the planned implementation of the solutions presented in the Suggestions section and present the traceability management tool to be used

in the experiment.

3.4.1 Gamified system: Capra

To carry out our experiment we needed a system to gamify and we choose the traceability management tool Capra. We decided to gamify this particular system for a number of reasons. The primary one being that we are in contact with the creator of the system and should we run into unexpected problems we can request assistance. We also have some previous experience using Capra and found that it included all rudimentary functionality required for our study, including the ability to vet traceability links and the creation of said links.

The system used in the experiment extends upon Capra by adding gamification elements to it. The nature of these gamification elements and why they are chosen over others is determined by two primary factors.

- The gamification theories presented in earlier chapters.
- The survey that was sent out to receive suggestions and opinions from previous Capra users as to what elements could be the most useful.

These gamification elements are implemented as part of the UI, extending the Eclipse Widget API. Alternative solutions could have been using a screen overlay or directly drawing it onto the screen to avoid being limited by the Eclipse API. Other functionality were implemented such as a data collection framework that can monitor participant activity within the system and some means of storing the results of each participant in a JSON format.

The exact details of developing this system is presented in the result section.

3.5 Evaluation

The evaluation step is important for design science research, in it we examine the quality of the work done as well as describe the systems in place to ensure the quality of our work.

3.5.1 Experiment design

In this section we describe the experimental design, the systems used within the experiment and the conditions under which it operates.

We used the following hypotheses for our experiment design:

Null Hypothesis 1 *There's no significant difference between the amount of vetted links between the two groups.*

Null Hypothesis 2 *There is no significant difference between the vetting correctness of the two groups.*

3.5.1.1 Participants

The participants of this experiment consisted of students from an academic software engineering background and sometimes with experience as developers. A reason for this particular group of participants is because of the different perceptions a group of people can have on a system and its features, depending on their background and age. Another important aspect to consider is that when designing a system, you need to design it with the intended audience in mind and the difficulty of designing the system increases the more diverse audience you intend to satisfy. With that said, the game elements we implemented were only moderately related to the specific profiles of our future participants. Not optimizing our system to our specific subsets of users is due to the fact that realistic traceability management tools users are diverse and have few if any distinct common denominators we can take into consideration. After they have completed the experiment, we asked each one of them about their previous experience with programming and traceability tasks as a means of validating their background.

A sample size of 24 was decided upon, 12 participants for the experiment on the standard version of Capra and 12 participants on the experiment with the added gamification elements. The participants were assigned to one of these two groups randomly. The amount of participants was influenced by resource constraints. We can manage twenty-four participants in the allotted time of our study without compromising the quality of other aspects.

3.5.1.2 Experiment equipment

All participants used one of two laptops available to us, to ensure every participant operates under controlled and equivalent conditions. The individual participants might have individual experiences and preferences for equipment such as keyboard and computer mouse. These preferences were not taken into consideration unless the participant voiced a specific problem with the equipment. In this instance it was to be documented and included in the results, but no such exception occurred.

3.5.1.3 Traceable dataset: Medfleet

For our experiment we decided to use the MedFleet project and its artifacts to be vetted. The MedFleet project system is a service for coordinating a drone fleet designed to rapidly deliver medical supplies in mapped areas. The system ensures that a single medical supply order is not delivered by multiple drones and other behaviour. It includes much more functionality, but the intricate details of the system are irrelevant to our study, since we are concerned with traceable artifacts rather than system behaviour.

Projects with a nearly complete set of traceability artifacts are rare, but MedFleet is an exception. The existence of these artifacts mean that we do not have to determine what artifacts are correct or create our own traceability artifacts and the accompanying underlying system.

3.5.1.4 Capra/MedFleet specific candidate links

During the experiment, the participants encountered three different kinds of candidate links to either be accepted or rejected. That being *requirements to code*, *requirements to assumptions* and *requirements to faults*. Below you can find a short description and an example of each type of candidate links.

- **Requirements to code** A requirement and its associated code.
Requirement: "Mission instructions shall be sent to the ground station".
Code: "config.java"
- **Requirements to assumptions** A requirement and its associated assumptions.
Requirement: "While any Drone is active, the GUI should display all relevant data".
Assumption: "GPS accuracy cannot be guaranteed".
- **Requirements to faults** A requirement and its associated faults.
Requirement: "If drones are headed to a collision then the flight control shall reroute the drones so they do not collide".
Fault description: "GPS coordinates do not reflect actual position of the drone".
Fault effect: "Drones crash into each other".

Looking back at the *requirements to code* example, it is not very clear if the "config.java" file should be associated with this specific requirement since the only information you have to base your decision on is the name of the file. When unsure on whether to accept or reject a candidate link, Capra offers the option to open the java file to check the contents of it. This helps any user to get a detailed understanding of what is contained in this specific java file and therefore allows them to make an informed decision whether to accept or reject this candidate link.

3.5.1.5 Introduction to Capra and its features

Before the actual experiment began, the participant was given a written document describing all the different features that the participants will encounter during the experiment. There were two different documents given out, one for the standard Capra version, the other for the gamified version which also describes the different game element features that had been added to Capra and their purpose. If the participants stated any confusion regarding the introduction paper, we were available to answer questions and clarify as much as possible. The experiment instructions given to the participants in both groups can be found in Appendix E.

3.5.1.6 Execution of the experiment

After the participants had been introduced to Capra and its existing features, they were given 45 minutes to complete their task. In this task they used Capra and its features freely, and we encouraged them to accept and reject as many candidate

links as they can.

As they were participating in the experiment, they were allowed to ask questions regarding the operation of Capra and our gamified system in case they required assistance. Questions that were not related to the experiment or task were not answered until after the experiment's duration.

3.5.2 Data collection and analysis

This section describes the data being collected during and after the experiment and how it helped us in answering our research questions.

3.5.2.1 Automatic data collection

As the experiment carried on, an underlying system recorded some of the input such as the various timings of vetting actions. The timings allowed us to measure the speed of the subject, whereas the recorded actions allowed us to compare their answers to that of the other participants. The individual timings between each recorded action could potentially be compared to their earlier timings to provide an indication as to how quickly they are improving while using the system, but was not used in our study. The data can also be used to analyze which type of Candidate Link took the longest time to vet. the data primarily used that was recorded by this system was the total amount of vetted links and the correctness of said links.

This data was used for answering *RQ3*, as it allows us see if the added gamification elements had any sort of impact on correctness and speed when vetting.

3.5.2.2 Survey

After the experiment the participants were asked to answer a survey, which consisted of questions asking about their background and previous experiences with coding. The survey looked different depending on which version of Capra were used. Both of these versions including the results can be found in Appendix B and C. In these surveys we specifically asked about the gamification features we implemented, or if they think these would be a good addition to the vetting task for the non-gamification group.

For the participants who used the gamified version of Capra, we asked questions on how they perceived them and if they thought it improved the experience of using Capra.

The surveys aids in answering *RQ1* and *RQ2*, since answers from the surveys provided data on what gamification features actually had an impact on the participants performance from their own perspective as well as how motivated they felt. It was also used to answer *RQ3* since the participant were able able to voice concerns with the gamified elements, and thus we can identify some of the potential disadvantages.

3.5.2.3 System usability scale

As our final data collection method, we used the well established system usability scale (SUS) [34]. This survey contains 10 questions which gave us a quick and robust way of evaluating the usability of Capra, both with and without the added gamification features. This allowed us to compare the two versions of Capra, and indicated if the added gamification elements kept the usability of the system intact. The results served as a good base for understanding where improvements are required and also where improvements have been made, such as complexity and ease of use.

This aids in answering *RQ3*, since through SUS we can detect potential faults of the gamified version of Capra and compare it with the result of the non-gamified version.

3.6 Ethical concerns with gamification

Gamification can be used to build habits [37], habits themselves can be detrimental to a person if said habit involves activities with harmful side-effects. There is therefore a precedence for briefly investigating and clarifying the ethical implications of gamification and how we should approach them within our study.

Habits are built over time, but our experiment includes only a single session no longer than 45 minutes. By containing the experiment to a single session, we lessen the chance of a habit forming rather than if it took place over many days. It should also be noted that our experiment contains no narrative element that could be deemed controversial or harmful. The experiment neither involves interaction with humans that could be exploited in favor of gamification rewards and no further rewards can be gained post-experiment to reinforce habits.

Gamification can by nature be considered manipulative since it's a technique used to affect human behaviour through processes the user might be unaware of. Consent is rarely asked or even expected of a gamified system's users. In our case we're not using the manipulative nature of gamification to cause behaviour that could be harmful to the user, the experiment has clear constraints and the effects are unlikely to affect everyday life.

Even though the experiment only lasts for a short session, the aim of the experiment and this study is to be able to predict how the gamification elements could be positively implemented in a users daily work. If used on a daily basis, as previously mentioned habits are built and could be shown to have a negative impact on the users work. Future work needs to take the ethical concerns into consideration in order to prevent any potential harmful side-effects of the users behaviour.

3.7 Validity threats

This section discusses threats to validity for the study and any mitigation strategies taken in order to reduce the impact of said threats.

3.7.1 Construct validity

3.7.1.1 Initial pilot survey misinterpretations

One potential construct validity threat faced is the possibility of the gamification features survey questions presented in Section 3.3.2.1 being misinterpreted. Since natural language can have multiple meanings it is possible that we as authors have a different meaning to it that is not accurately portrayed to the readers. Our mitigation strategy for this was to run it through our two supervisors, whom have experience with assembling surveys and thus allowing us to run our writing through multiple perspectives and discern whether the meaning remains intact.

3.7.1.2 Experiment survey misinterpretations

Much like the pilot survey, the surveys filled out by the experiment participants risk being misinterpreted. If the exact meaning of a question is not concrete, experiment participants might answer the survey based on an entirely different scale than what was intended. Our mitigation strategy was to first allow an expert to provide feedback on the questionnaire in order to lower the risk of misinterpretation. We also encouraged the participants to ask questions when its meaning was unclear to them.

3.7.1.3 Vetting misinterpretations

One construct validity threat is the possibility of each experiment participant vetting with different underlying assumptions. One person's rejected candidate links might be accepted by another person, not necessarily because one might be misinformed, but their own subjective opinion as to what constitutes a correct link varies. Naturally we cannot have all participants conform to the same opinion and it would serve no purpose to do so, as the vetting process in a realistic environment involves the vetters having different opinions. But it is important that each participant has the same basic idea as to what the different tasks imply. To ensure each participant has the same understanding of the task, they are asked to read two pages of experiment instructions. These instruction explain the task so that everyone operates under the same basis. If questions regarding the tasks are brought up, they are answered by us as long as it does not directly solve a task for them.

3.7.1.4 MedFleet domain knowledge

None of the participants had any prior knowledge of the MedFleet system and a big majority of them did not have prior experience in systems similar to it either. This can in turn affect the end result of the correctness of the vetted links. The results showed that the average acceptance correctness rate were extremely low, and this

is most likely down to the fact that none of the participants had been part of the development of the system. In a real life setting, the people undertaking the task of vetting the candidate links would have had some prior knowledge and experience with the system in question and should therefore be able to make better decisions compared to the participants of this experiment. Such a setting is very difficult to reproduce and one way to mitigate this was to make sure that every participant were on the same level of knowledge on MedFleet, which in this case was basically level zero. The MedFleet system also came with the knowledge on what links that were correct, which enabled us to see the end result of the participants decision. This is not something that comes with every system and was the major reason for using MedFleet in the experiment.

3.7.2 Internal validity

3.7.2.1 Personal preferences and personality conflicts

Capra is currently available as an open source release but is still undergoing development for improving the system further. One threat that was considered is that if any of the participants had prior experience in using Capra itself, or any other traceability management tool, this could affect the results of this study in a few different ways. Capra, along with every other system imaginable, has a learning curve involved. A mitigation for this threat is that anyone that would have used Capras verification features prior to the experiment are not allowed to participate, since they would have an advantage compared to the other participants which is using Capra for the first time.

Another requirement for the participants is that they should not have had an extensive amount of experience in using other traceability management tools. The reason for this is because the intention of this study is to compare the two version of Capra and to what extent the added gamification elements can improve it. If any of the participants would have experience in using any other commercially available traceability management tool's verification, there is a risk that these participants can use the pre-existing knowledge from that system to perform better in Capra. While such a comparison would be interesting, it is unfortunately outside of the scope for this study.

3.7.2.2 Inaccurate data collection

The data collection framework was not thoroughly tested, while some manual testing was done there was no rigorous testing applied to its accuracy. Potential bugs might affect the produced data, thus a mitigation strategy is warranted. This mitigation strategy was to record the screen of the participant as the experiment carries on. In the case of anomalous data it would be tested against the video recording of the screen.

3.7.2.3 Poor UI design

There is a possibility that the UI design implemented for the experiment is lacking in several qualities that could impact the outcome of the experiment. Our intention is to present to the user a UI that would fulfill all the basic needs of user. But however, since we lack formal interaction design training, mitigation strategies have been taken.

The System Usability Scale (SUS) [35] is a recognized standardized means of evaluating the usability of a system. SUS was integrated into our post-experiment survey to determine whether the UI we had designed fulfilled their usability expectations. This does not solve the issues our UI might have, but it would highlight them so they could be presented in the system.

The initial pilot survey also allowed us to determine which gamification features would likely not be a good fit based on the opinions of people who were familiar with the system.

Along with these two other mitigation strategies we relied heavily on the feedback of others throughout the development process to ensure we were not limited to our own perspectives of the UI.

In addition to this our gamification elements are very simplistic and were emulation of gamification elements designed by others already proficient in UX design and gamification.

3.7.3 External validity

3.7.3.1 Generalizing the results

The scope of this study is quite compact and clear, and it is important to discuss to what extent the results presented can be generalized. The study is targeted at improving traceability tools with the use of gamification. The results present how a few specific types of gamification elements (Levels and badges) can have an impact on a specific traceability task (Vetting), when tested in a controlled experiment. These results can simply be used as a basis for future research concerning traceability and gamification. In order to use these results and to make a generalization of it, more game elements needs to be tested and for different kinds of traceability tasks. It also needs to be tested on participants which would have a different background and experience than the participants that took part in our experiment. Finally, the results of this study and any future studies covering the same topic area, can be used as a basis for testing a gamified traceability tool in an actual industrial setting.

3.7.4 Reliability

The conducted experiment should be easy to reproduce by other researchers, since the design of the experiment is described in detail and all the surveys and responses are located in the appendices. Capra is open source and MedFleet should be possible to get access to by getting in contact with the authors. A full reproduction of the experiment should show the same results if the same amount and type of participants are used (12 per group, mostly students and junior developers).

A bigger sample size and participants with better knowledge in traceability or software development in general could potentially show a different but interesting result. The result might also differ if similar gamification features would be implemented in another traceability management tool than Capra, and with the use of another system than MedFleet.

3.7.5 Unforeseen events during the experiment.

During the experiment a number of unplanned for events occurred. They are detailed here, their consequences and solutions explained.

3.7.5.1 Previous experience

We were informed pre-experiment by one participant that they had previous experience with Capra. This naturally allows the participant to learn the system quicker since they are already familiar with Capra. But in this particular instance the system was unrecognizable, this was due to the version and functionalities of Capra utilized in our experiment was not available at the time they had previously used Capra. With this taken into account their previous experience should have no impact on their performance.

3.7.5.2 Loss of video

The video screen recording software license expired once mid-experiment. From that one session no video footage were recovered, but all the other artifacts remained intact so no data was lost.

3.7.5.3 Loss of JSON

For one participant the JSON file detailing the exact timing of each individual action was lost due to mismanagement of the file browser. It should be noted that their final results was recoverable along with the video files and thus we still retained all the data we needed.

3.7.5.4 Varying group sizes

The experiment duration itself is 45 minutes long. This is not including the pre-experiment information briefing/reading. To speed up our ability to perform the experiments, we on several occasions ran two participants at the same time, in these

3. Methods

instances they were not allowed to collaborate, they only shared the same locale and informational briefing.

4

Results

The first section in this chapter describes the extended functionality of Capra and the strategies we employed while implementing it with the gamification features. The second section explains the final implementation and functionality of the level and badge feature. In the third section the results from the experiments are separated into three sub sections, with the first sub section containing information about the participants background. The second sub section contains the responses from the post-experiment survey, and the third sub section contains the results from the vetting task.

4.1 Implementation

Three distinct features had to be implemented for our experiment to be executable. These were the two gamification features, Levels and Badges and finally a data collection system which collected the necessary data during the experiment in order to lower the amount of manual work required.

There were multiple considered means of gamifying Capra. The two implementations that were primarily considered was Eclipse's built in GUI library labeled Standard Widget Toolkit (SWT) and a screen overlay that was not tied to the Eclipse GUI.

The screen overlay would need to be built from scratch by us, since we could not find an appropriate library. Building the system from scratch would be time intensive and due to the limited time available to us, we decided to not go with this approach, and instead went with the Eclipse GUI extension.

Eclipse's built in GUI library had all the functionality necessary for the two planned gamification features, but it lacked visual polish. A custom system would have been more flexible in terms of aesthetics, however, the limitations of the built-in GUI did not greatly hinder the implementation of the rather simplistic features we had in mind.

Two aspects were affected by the chosen library's limitations. The first aspect affected was that the library could not render high-resolution images onto the GUI with altered heights and widths, if one looks closely at the pictures present in our gamification implementation, the resolution is quite low. The second affected aspect is that the level bar's color could only be determined by the operating system, thus

the colors available to our setup were yellow, red and green. The green color was chosen over the others because it was the only color that supported animation and is predominantly associated with progress bars in the west.

Installing Capra was a cumbersome process and involved many challenges, but with the aid of one of the creators we were able to successfully install it. The gamified elements were implemented after a short learning period in which we studied official code samples of proper SWT implementations. After this the development itself was simple, but tedious, since to align all GUI elements correctly one had to input specific measures which was done by trial and error until the correct result was attained. There was also an issue in which text could not be rendered over other UI elements normally, but was solved after multiple hours of investigating the issue. Over the course of the development time, the GUI was shown to our supervisors in exchange for feedback on it, with this in mind gamifying Capra was an iterative process. Along with these features we also introduced a rudimentary Automatic Data Collection framework to minimize the manual work in taking notes during the experiment.

The data collection framework records these types of data:

- Nanosecond intervals between each vetted link.
- Time of each vetted link.
- Number of vetted links
- Which artifact that was vetted.
- Which parent the artifact belong to.
- Experiment start time

Our final prototype implementation consists of 5 separate .Java files and modifications to a single pre-existing .Java file.

4.1.1 Modifications to existing classes

“TreeView.Java” is a class that manages the vetting actions available to the user, such as Accepting a trace link, Rejecting it and browsing a source file. The class was modified to call the new gamification classes whenever one its actions were taken and pass on the information of that decision.

4.1.2 The Added Files

Note that these files might not be appropriately named, as their functionality has changed over the course of the development. As of now we have only treated it as an internal prototype with a limited lifespan and is thus not optimized for readability.

BadgesView.Java

This class contains the instructions for SWT to construct the “Badges” window used during the experiment.

GamificationView.Java

This class contains the instructions for construct the “Levels” window used during the experiment.

Profile.Java

Data Structure used for handling the data of an individual participant.

ExperimentHandler.java

Records the actions done by the user and wraps it around a Profile data struct.

PointsManager.Java

Handles the underlying points system for the “Levels” feature and converts the results of the experiment as a serialized classes in a JSON format through the GSON library.

4.1.3 Libraries Used

GSON

Developed by Google, this open source library can be used to deserialize and serialize java objects along with much auxiliary functionality related to those features. It saw extensive use within our Data Collection Framework to cut down on development time when it came to storing information onto the hard drive in the recognized format JSON.

Standard Widget Toolkit

The SWT is a library which is used for implementing Eclipse GUI widgets. This library is compatible with Capra, as it is built on top of Eclipse. Given that SWT has already been used previously for Capra and since we did not intend to make visual elements that stood out compared to others on the screen it was a good fit.

4.2 Gamification features

The final gamification features were based on previously studied gamification theory and the results of the pilot survey sent out to examine the viability of the proposed features. It was decided on implementing the level and badge features. The leader board also had quite a good response from the pilot survey, but the decision was made to exclude it since it does come with the competitive element which is known to discourage some individuals and also pointed out by some of the responses from the survey. Also if adding to many features it might increase the difficulty in pin-pointing which feature which had a greater impact compared to the others.

The level feature can be seen in figure 4.1. What can be seen is the current level of the user in the green star, the total number of points accumulated and how much progress is left until the next level is reached. The leveling system was very simple, you got 10 points for both accepting or rejecting a candidate link and for each level

you required 100 points.



Figure 4.1: Level feature

The badge feature can be seen in figure 4.2. There was a total of three badges to be collected. The first is to accept a total of 20 links, the second to reject a total of 20 links and the third is to open a total of 25 source files. What can be seen is the logo of the badge, how many links have been accepted/rejected, how many source files have been opened and how much progress has been made for each badge. When the requirements of a badge are fulfilled, the logo turns into a green color as can be seen on the rejected links badge.

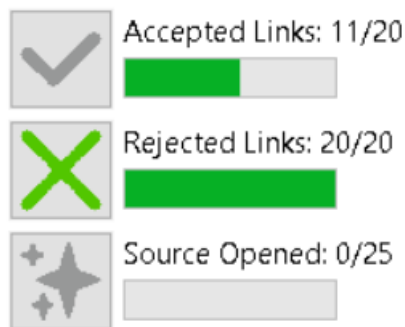


Figure 4.2: Badge feature

4.3 Experiment results

This section present the data collected during the execution of the experiment. It is divided into four sub sections with the first one containing the background info for the participants. The second sub section contains the results of the System Usability Scale. The third sub section contains the results from the post-experiment survey. Finally, the fourth sub section contains the correctness of the candidate links the participants vetted and the evaluation of the hypotheses. The experiment participants are divided into two separate groups, one using the gamified version of Capra (Gamification group or GG) and the second which used the standard version of Capra (Control group 2 or CG).

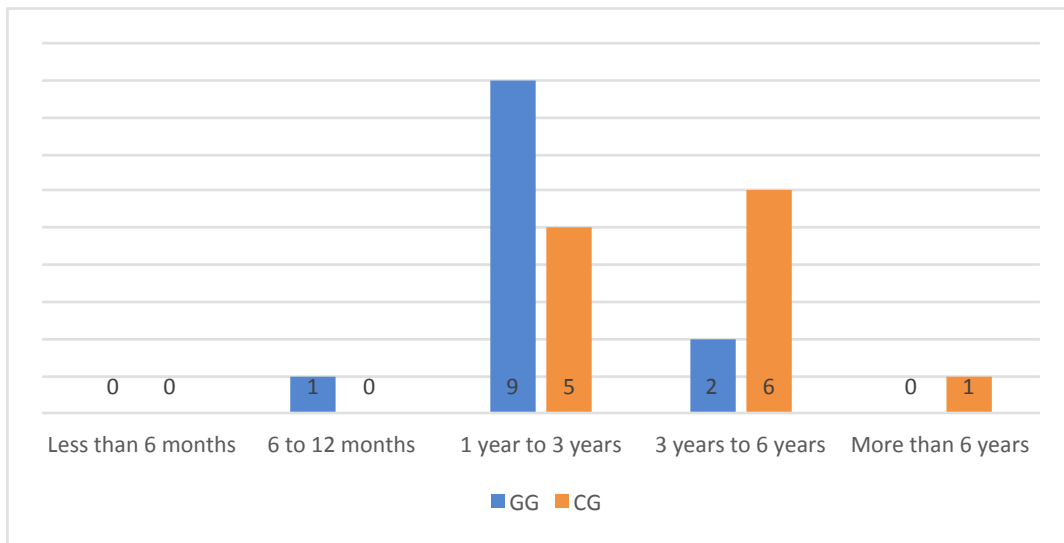


Figure 4.3: Q1: Software development experience

4.3.1 Background

Before the experiment started, each participant was asked to fill out the first section of the survey, which concerned their background and experience in software development. The questions asked were about their current studying/working situation (bachelor/master student, junior/senior developer etc.). They were also asked about their experience in software development, utilizing tools such as Eclipse and software traceability management tools.

In the gamification group there was a total of 12 participants. The group consisted of five bachelor students, three master students, two junior developers, one system designer and one mid-level developer. Two of the bachelor students worked at the same time as junior developers.

In the control group there was a total of 12 participants. The group consisted of one bachelor student, seven master students, three junior developers and one developer. One of the participants were a master student and worked at the same time as a junior developer.

Every participant were asked the following three questions:

- **Q1: What is your experience in software development?** See results in figure 4.3
- **Q2: What is your experience in using Eclipse for software development?** See results in figure 4.4
- **Q3: Do you have any experience with software traceability?** See results in figure 4.5

The questions **Q1**, **Q2** and **Q3** were asked in order to see if either group would in

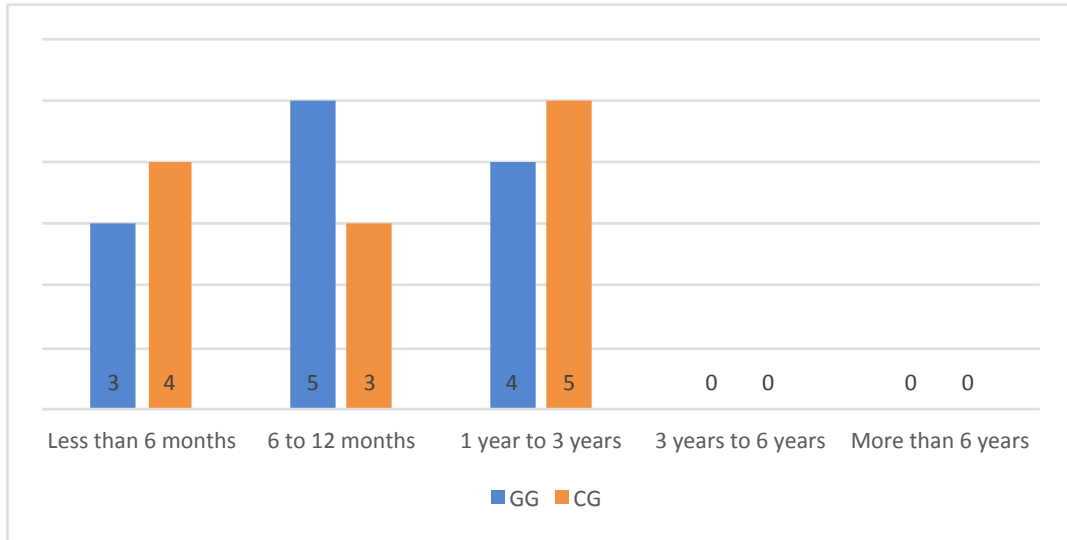


Figure 4.4: Q2: Experience in using Eclipse

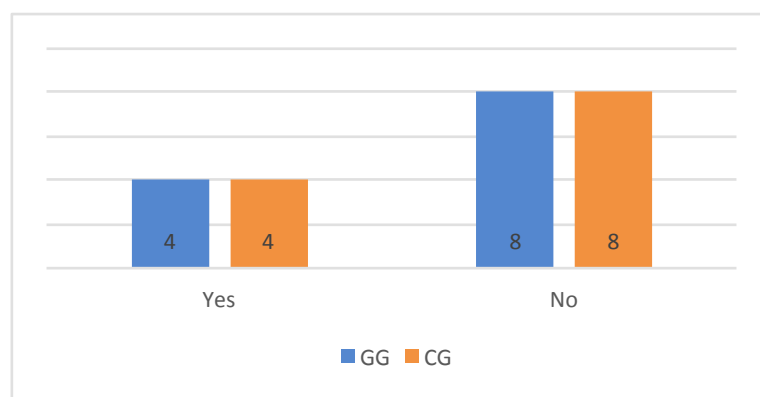


Figure 4.5: Q3: Experience with software traceability

some way have an unfair advantage. If this would have been the case the results would not have been a reflection of the system but rather the experience of the participants.

Looking at the graphs, there is a quite good and even spread of experience between the two groups, especially when looking at figure 4.4 and 4.5. The biggest difference to mention and keep in mind can be seen in figure 4.3, which shows that the control group does have more people with longer experience in software development. However, there was not a big difference in the spread of experience between the two groups and should not give the control group an unfair advantage other than it potentially resulting in a slightly better outcome of the vetting task for them.

4.3.2 Post-experiment questions

After the experiment was done, the participants were asked to answer a post-experiment survey.

4.3.2.1 General questions about the experiment

Some of the questions in the survey were the same regardless of the group the participants were in. The results from these questions are presented and compared in this section.

The questions **Q4** and **Q5** were asked for the same reason mentioned in the previous section, in order to see if there is a reason to believe that one of the groups would have an unfair advantage in any way. As mentioned before, there would only be a reason to believe this if there was a clear spread on the graphs. By looking at figure 4.6 there is only two of the participants who had prior experience in similar systems and in figure 4.7 there is almost an identical spread between the groups. The responses from these questions shows that there is an even spread of knowledge between the groups.

The first question was a *Yes* or *No* answer, while the second could be answered on a scale from 1-5, 1 being *I completely did not understand what the system does* and 5 being *I completely understood what the system does*.

- **Q4: Do you have experience with developing software similar to MedFleet (Software that involves the use of drones and route planning)?** See results figure 4.6
- **Q5: How confident were you with your understanding of the Med-Fleet system?** See results figure 4.7

The questions **Q6** and **Q7** assisted in answering **RQ1** and **RQ2** by making sure that adding the level and badge feature does not make the vetting process less enjoyable than it was before as well as not making people less motivated to work on the task by adding them. Both questions were answered on a scale of 1-5, 1 being *Strongly*

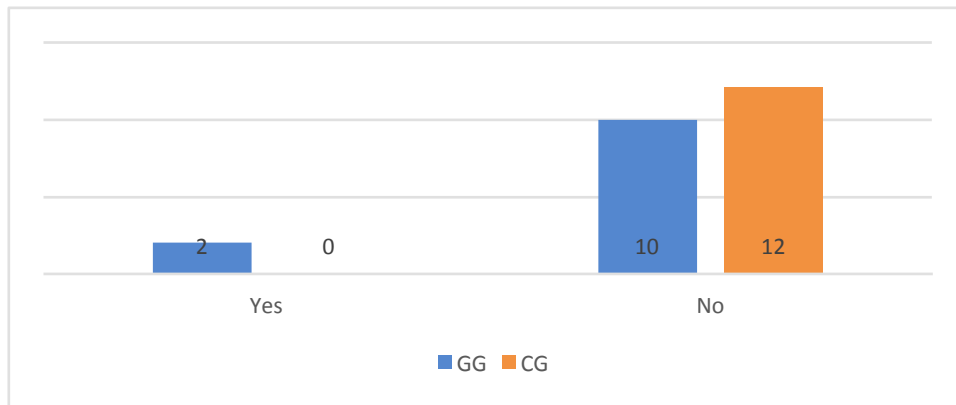


Figure 4.6: Q4: Experience with systems similar to MedFleet

disagree and 5 being *Strongly agree*.

Looking at the graph in 4.8, it shows that it at least did not make it less enjoyable for the participants in the gamification group to add the levels and badges. The results also indicates that the levels and badges did increase the enjoyment of the vetting process to a certain degree. When running a Mann-Whitney u-test on the responses from **Q6**, the result is significant ($p \approx 0.040$)

Looking at the graph in figure 4.9, it shows that the majority of the participants from both groups did have a sense of motivation to complete the task. Only two of the participants from the gamification group indicated that they had a lack of motivation compared to the control group which had five. No participant from the control group showed any indication of a middle ground, it was either a sense or lack of motivation. When running a Mann-Whitney u-test on the responses from **Q7**, the result is insignificant ($p \approx 0.4$)

- **Q6: The current process for vetting trace links (as used in the experiment) is enjoyable?** See figure 4.8
- **Q7: I felt motivated to complete the task** See figure 4.9

4.3.2.2 System usability scale

In order to compare the usability of the versions of Capra, we used the System Usability Scale (SUS). It contains a total of 10 statements, all of which is answered on a scale of 1-5, 1 being that you "Strongly disagree" with the statement and 5 being that you "Strongly agree" with the statement. The statements can be found in Appendix D and also in [34].

In table 4.1 you can see the average result for each statement (S1-S10) given by both groups and also the final score the systems attained. The final score can land between 0-100.

In the study by Bangor et. al [35], a way to interpret the SUS score is presented.

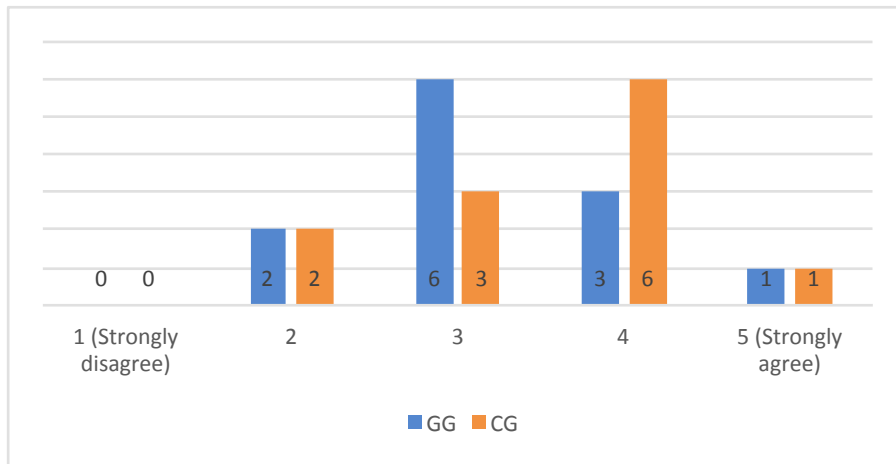


Figure 4.7: Q5: Understanding of the MedFleet system

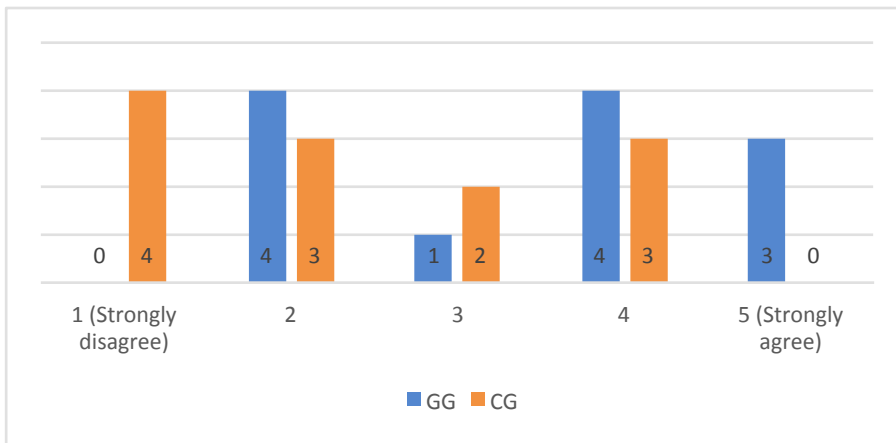


Figure 4.8: Q6: Enjoyment of the vetting process

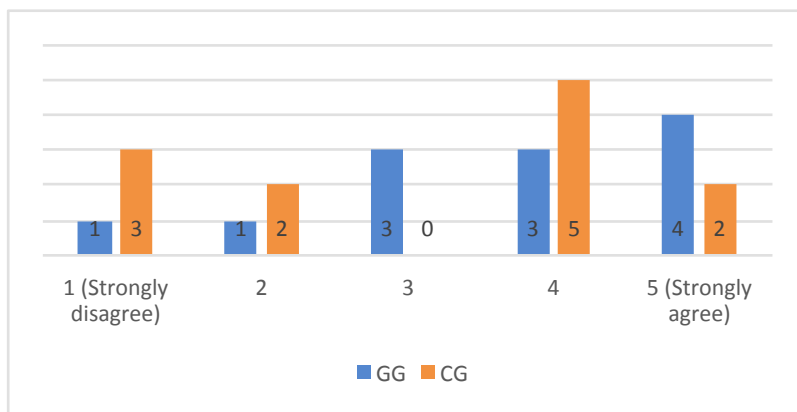


Figure 4.9: Q7: Motivation to complete the task

The design and usability with a score of 0-25 is considered the *worst imaginable*, 26-38 is considered to be *poor*, 38-52 is considered to be *OK*, 52-73 is considered to be *good*, 73-85 is considered to be *excellent* and finally 85-100 is considered the *best imaginable*.

Group	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Score
GG	2	2.92	3.2	2.75	2.75	3.08	3.08	2.83	2.5	2.75	69.6
CG	1.25	2.66	2.83	3.08	2.66	3.25	0.91	2.75	2	3	61.04

Table 4.1: SUS score

The gamification group gave it a final score of 69.6. Taking into account the scale presented in [35], Capra extended with the levels and badges feature is considered to be *good* and note that it is just on the brink to be considered excellent.

The control group who used the standard version of Capra, gave it a final score of 61.04. Taking into account the scale presented in [35], Capra without the levels and badge feature is also considered to be *good*, but at the same time it scored eight points lower than the extended version.

The results from a Mann-Whitney u-test on the average score from both groups showed to be insignificant ($p \approx 0.904$).

Despite the fact the the results were insignificant from the test, the SUS-score on its own should not be discarded. It is a positive result that even though Capra has been extended with the level and badge features, the usability stays intact.

4.3.2.3 Gamification group on levels and badges

Since the gamification group got to interact with the levels and badges feature, they are asked a few questions on how they perceived them and if they thought that it had a positive or negative impact on the vetting process or on their individual performance.

The questions **Q8-Q13** were asked to the participants in the gamification group about the levels and badges feature. Each question was answered on a scale of 1-5, where 1 being "Strongly disagree" and 5 being "Strongly agree". The answers presented on table 4.2 shows the average result for each question for the two features.

The answers from questions **Q8-Q9** and **Q12-Q13** will be considered the most with regards to **RQ1** and **RQ2**, as well as some of the comments given by the participants. The answers from **Q10-Q11** will be considered the most with regards to **RQ3**, also here including some of the comments given by the participants.

- **Q8: I had no issue understanding the levels/badges**
- **Q9: The levels/badges made the task of vetting trace links satisfying**

- **Q10:** Do you think that the levels/badges helped you decrease the time spent on each link?
- **Q11:** Did the levels/badges make you focus more on verifying as many links as possible instead of verifying each link correctly?
- **Q12:** I felt that the levels/badges contributed towards being motivated to complete the task
- **Q13:** Overall, the levels/badges feature was a good addition to the traceability tool

Looking at the results presented in table 4.2, there is not a lot of difference on the average result between the levels and badges feature. The majority of the participants did not have an issue to understand the features, but it was considered easier to understand the badges which is overall the biggest difference between the two features.

Feature	Q8	Q9	Q10	Q11	Q12	Q13
Levels	3.4	3.6	3.1	3	3.4	3.9
Badges	4.2	3.8	2.9	3	3.5	3.8

Table 4.2: GG responses regarding the *levels* and *badges* feature

4.3.2.4 Control group on levels and badges

The control group never got to interact with the levels and badges feature so they could obviously not be asked what they thought about it. Instead they got a glimpse on how the two features could look like and were asked on how they think the vetting process and their own individual performance would be affected by adding these features.

The questions **Q14-Q17** were asked to the participants in the control group about the levels and badges feature. Each question was answered on a scale of 1-5, where 1 being "Strongly disagree" and 5 being "Strongly agree". The answers presented on table 4.3 shows the average result for each question for the two features.

The answers from question **Q14** will be considered the most with regards to **RQ1** and **RQ2**, as well as some of the comments given by the participants. The answers from **Q15-Q17** will be considered the most with regards to **RQ3**, also here including some of the comments given by the participants.

- **Q14:** Do you believe that levels/badges would make the task of verifying traceability links more satisfying?
- **Q15:** Do you believe that levels/badges would help in decreasing the time spent on each link?
- **Q16:** Do you believe that levels/badges would make you focus less on verifying the links correctly and focus more on verifying as many as possible?

- **Q17: Do you believe that levels would make you skip the more advanced links and go for easier ones in order to “level up”/receive more badges faster?**

Much the same as the results from the participants interacting with the levels and badges feature, the participants from the control group did not think there would be that much of a difference between the two features. The difference worth mentioning is that the participants thought that the levels feature would make them focus less on correctness and more on verifying as many links as possible, at least compared to the badges feature.

Feature	Q14	Q15	Q16	Q17
Levels	3.4	2.75	3.4	3.2
Badges	3.6	2.6	2.8	3.1

Table 4.3: CG responses regarding the *levels* and *badges* feature

4.3.2.5 Answering the research questions

In order to help us answer **RQ1** and **RQ2**, the answers from questions **Q8-Q9** and **Q12-Q14** are considered the most, as well as some of the comments given by the participants. When asked to rank certain aspects of the levels and badges from these questions, the majority of the participants from both groups showed a positive opinion about them. This is an indication that both levels and badges are positive candidates to implement in the traceability vetting process.

Regarding the impact they would have on a users motivation for the vetting task, the results show that it would at least be a positive impact but it is not as clear on to what extent the impact would be. This would require more iterations from this study and also to test it in the industry.

4.3.3 Vetting task results

This section present the results from the vetting task that all the participants undertook. The results from the gamification group can be seen in table 4.4 and the results from the control group can be seen in table 4.5.

The tables acronyms for each column stands for:

- **P:** Participant
- **CA:** Correctly accepted
- **IA:** Incorrectly accepted
- **TA:** Total accepted
- **ACR:** Accepted correctness rate
- **CR:** Correctly rejected
- **IR:** Incorrectly rejected

- **TR:** Total rejected
- **RCR:** Rejected correctness rate

The gamification group vetted a total of 1564 links (accepted 836, rejected 728), which is 130 links on average per participant. The average correctness rate was 50.4%

P	CA	IA	TA	ACR	CR	IR	TR	RCR
1	12	33	45	26.7%	45	2	47	95.7%
2	6	55	61	9.8%	84	8	92	91.3%
3	12	50	62	19.4%	58	2	60	96.7%
4	11	116	127	8.7%	54	4	58	93.1%
5	13	58	71	18.3%	27	1	28	96.4%
6	13	111	124	10.5%	101	4	105	96.2%
7	9	76	85	10.6%	44	5	49	89.8%
8	8	28	36	22.2%	65	6	71	91.5%
9	11	53	64	17.2%	32	3	35	91.4%
10	6	74	80	7.5%	50	8	58	86.2%
11	9	54	63	14.3%	70	5	75	93.3%
12	6	12	18	33.3%	42	8	50	84%

Table 4.4: GG results from the vetting task

The control group vetted a total of 1911 links (accepted 911, rejected 1000), which is 160 links on average per participant. The average correctness rate was 54.4%

P	CA	IA	TA	ACR	CR	IR	TR	RCR
1	12	67	79	15.2%	36	1	37	97.3%
2	8	75	83	9.6%	84	6	90	93.3%
3	5	60	65	7.7%	54	9	63	85.7%
4	8	54	62	12.9%	80	6	86	93%
5	9	165	174	5.2%	134	9	143	93.7%
6	9	123	132	6.8%	221	9	230	96%
7	6	71	77	7.8%	119	11	130	91.5%
8	10	22	32	31.3%	65	4	69	94.2%
9	13	12	25	52%	23	1	24	95.8%
10	10	20	30	33%	16	3	19	84.2%
11	10	87	97	10.3%	34	4	38	89.5%
12	8	47	55	14.5%	65	6	71	91.5%

Table 4.5: CG results from the vetting task

As mentioned in the experiment design section in the previous chapter, there were two null hypotheses to be tested with a Mann-Whitney u-test:

Null Hypothesis 1: "There's no significant difference between the amount of vetted links between the two groups."

The results from the test on each participants total number of vetted links showed to be insignificant ($p \approx 0.542$). Therefore, we can not reject the null hypothesis.

Null Hypothesis 2: "There's no significant difference between the vetting correctness of the two groups."

Similarly, the results from this test on each participants average correctness rate showed to be insignificant ($p \approx 0.912$). Therefore, we can not reject the null hypothesis.

5

Discussion

In this section we discuss the implications of the result and how it relates to the goals of our study.

5.1 The effects of gamification on traceability management tools

The participants showed a mixed attitude towards the idea of gamification, with 66% of the CG answering that the system would be more satisfying to use if a leveling system was implemented, and the rest answering negatively. As for the badges features, the CG answered 58.3% positively, with 16.6% answering neutrally. Most favor the idea of gamification in terms of user satisfaction, but doubt it would have an effect on their performance. This is interesting since the GG was slower, as they vetted a total of 1564 links, whilst the CG vetted a total of 1911 links. As for enjoyment of the work process, $\approx 58.3\%$ of the GG answered positively, whilst in the CG only 25% answered positively, with none strongly agreeing that it was enjoyable.

Since the data is mostly insignificant we cannot with any certainty say if gamification makes the user less accurate or slower during the vetting process. However, given that in the GG the user's stated enjoyment of the system was significantly positive, future studies should investigate how gamification can affect a users performance over multiple sessions. Given that the system is more enjoyable to use, it might be more tolerable to use it over longer or more frequent sessions. While our study did not extend to multiple opportunities of using the system, we can safely assume that if a system is more enjoyable, people can be more open to using it again.

5.2 Cost of speed and competition

One free-form comment that was brought up was the factor that the gamification elements could encourage a user to vet more links rather than focus on vetting links correctly. This is a reasonable concern since only the speed at which one vets links can be realistically rewarded through gamification features.

Was our implementation unable to encourage both speed and quality of the vetting through a faulty design? In our case we would argue our design was not incorrect in this aspect. This is since one can reward the speed at which a user vets links, but

one cannot reward the quality. To do so it would require the ability to determine which vetting decision is correct to reward the user appropriately. As of now there exists no reliable means of determining which vetting decision is correct. If such a tool existed there would be little need for our study or human input to begin with.

That speed compromises quality of the vetting is however an interesting factor to consider, since at least one study has shown that gamification applied without any sort of quality feedback can harm the user’s ability to make correct decisions, with the benefit of increasing speed. But if quality feedback is provided the speed remains faster, whilst the correctness is not affected [30]. However, our results shows no conclusive or significant data on whether gamification affects speed or correctness.

Alternative means of rewarding correctness have been considered, such as rewarding for taking extra time to analyze a trace artifact in detail. Another alternative proposed by an experiment participant is to allow the participant to see the vetting decisions of other users upon making their own. The revealed other users decision would allow the first user to get an idea of what others thought was correct, but is by no means guaranteed to be correct. This concept however leans dangerously close to competition and thus some players might be discouraged by it as discussed in the theory chapter of this study. It could be interesting to analyze the effects of different correctness specific reward systems in a future study and allow optional competitive features, but this lies outside the scope of our study.

One other comment mentioned on at least two occasions, was that it was difficult to ascertain how well they did. This is assumed to be because the levels they have earned have no particular meaning until they are ascribed one. It’s difficult to gauge if reaching the arbitrary level 10 is “good” until they can compare that result with another.

This is a legitimate problem with gamification, for as soon as we let others compare, the less well performing can be discouraged since they cannot compete. Once you cater to the competitive minded audience it is more difficult to cater to those who dislike competition and vice versa as previously explained. This indicates that for optimal gamification in an environment, it needs to be properly aligned with the goals and personality of its users, which can be difficult if you have a diverse user base.

5.3 Inaccurate acceptations and accurate rejections skew

The results show a considerable lean towards inaccurate accepted vetting decisions and accurate rejecting decisions, with only $\approx 13,8\%$ of the GG’s accept decisions being correct, and $\approx 92,3\%$ of their rejections being correct. Similarly the CG has a ratio with $\approx 11,8\%$ correct accepts and $93,1\%$ correct rejections. One explanation for the low accept correct rate is the nature of how the trace links were generated.

Automatic generation often produces results that appear correct superficially, but are in fact not correct. Along with this the generation is far more likely to create links that should be rejected than accepted, yet we see an almost perfectly even distribution between accepted and rejected vetting decisions, amounting to 1747 total accepted links vs 1728 total rejected across both groups.

We have considered that perhaps the users were assuming that they should accept one link for roughly each rejected one. Some participants even commented after the experiment that they felt obliged to accept some links rather than reject them, they assumed that at least a few of them had to be correct. One individual participant even made a comment after the experiment that since both the Accept and Reject badge required the same value to obtain, the one participant assumed by default there should roughly be an even distribution between accepted and rejected links. We considered whether there might be some weight behind it since the groups had an accepted/rejected distribution that's close to 50% (the GG: $\approx 53\%$ accepted, $\approx 47\%$ rejected, the CG: $\approx 48\%$ accepted, $\approx 52\%$ rejected). But on an individual basis few participants have a close to 50% accept/reject ratio. Since we don't see an even accept/reject ratio commonly on an individual basis it's unlikely the users operated under the assumption that there should be one to begin with. The overall even distribution is likely therefore a coincidence, but to be certain the experiment would have needed a larger sample size or a second iteration with a survey incorporating these new questions.

The fact that the participants were new to the concept of vetting candidate trace links, and in combination with vetting a system they had no prior knowledge about is most likely one of the bigger reasons the acceptance accuracy were so low. A different result could probably have been recorded if the system to be vetted would have been known beforehand and if the participants would have had previous experience in a similar vetting task. This is why further research in an industrial setting would be interesting, to have the vetting task be done by the systems actual developers. This would be a better representation of a real life setting and so it could produce a better result which shows if the gamification features would have had an impact on the correctness of the vetted links.

If we assume that the low correctness rate for accepted trace links is caused by how they can be misleadingly generated, this would have the meaning that if one wants to effectively utilize a traceability management tool with generated links, the user should be informed not only of the task, but how the links are created and what one should roughly expect in terms of accuracy as this could influence the vetting process.

5.4 Are traceability management tools worth using?

Tools often exist to improve a process, or enable it in the first place. In the case of traceability management tools we are curious as to how useful they actually are since we see a considerably high incorrectness rate for accepted trace links. Roughly $\approx 13,8\%$ of the GG's accepted trace links were correct, $11,8\%$ for the CG. At this point one should consider whether its beneficial to use the system if the vast majority of accepted links are incorrect, especially given that inaccurate trace links can be more dangerous than none at all [3].

On the other hand, the rejected trace links are primarily accurate, the GG correctly rejecting $\approx 92.3\%$ of their links, whereas the CG correctly rejected $\approx 93.1\%$ of their links.

It is worth considering that if the low accept ratio might be due entirely to how the links are generated in this particular experiment. The links are generated by searching for keywords within the files and trying to map that to a probability of that particular trace artifact belonging to a given trace source. It is therefore fully possible for the generator to generate links that seem like they would be correct as they include all the necessary keywords, but in fact are not. In a system where all links are generated by manual input it is possible that these misleading links are less prevalent, but that the links are more prevalent is of course also possible. It is also possible that the low correctness ratio is only present in Capra and would not be found in other tools.

Determining exactly what is causing this noticeably low correctness would require further studies, but if the low correctness is universal rather than limited to our tool the viability of Traceability Management Tools could certainly be questioned. One other angle to be investigated is also whether humans should exclusively reject links rather both accept and reject, as they appear far better at rejecting than accepting.

5.5 SUS result

The high SUS score is an indication that the validity threat of poor UI design is less of a factor since in fact it is indicated the UI of the gamified version of Capra received a higher score than the standard version, while both of them received a positive score. A negative SUS score could have the implication that the gamification elements were of poor UI design and that the results obtained from them represents a system that is difficult to use. On the other hand if the SUS score is high, the data obtained from the study represents a system that is more in-line with industry and academic standards of good usability.

5.6 Insignificant result

The presented results were exclusively insignificant, except for for the responses given in **Q6**. Here we intend to discuss as to why that could be and what it means

for our study.

We did not expect factors such as correctness and speed to differ by a significant amount as our gamification features are rather lightweight and do not really have a huge impact on the vetting process itself. The intention of the gamification features was to have a positive impact on the user satisfaction and motivation, so that is where a significant difference was expected but unfortunately this was not the true for all cases.

There can be several reasons for the results to be insignificant, and the first one to come to mind would be the sample size. It is generally difficult to get a significant result from a sample size of 12 for each group and considering that the gamification features were rather lightweight it becomes even more difficult since the smaller the difference is between the systems, it is more likely to have a smaller impact on the outcome. This is something to consider for future iterations of similar studies, that if one does not have time or access to a large sample of participants, it could be beneficial to spend some extra time to develop and examine a traceability management tool which had the gamification features in mind from the start. Such a system might feel better integrated so it could show to have a bigger impact on both user satisfaction and motivation.

Another reason for the insignificant result could be that it is difficult to examine user satisfaction and motivation for a task when participants only test the system for 45 min. Ideally this would be done in a longitudinal study within a software company which would be willing to participate. In such a study you could be able to see how both user satisfaction and motivation got affected by the gamification features over time.

Even though the results were insignificant, it still shows that the majority of the participants thought that both the level and badges features had a positive impact on the motivation to complete the task (**Q12**) and that they were a good addition to the tool (**Q13**). The uncertainty lies on to what extent it would have an impact on motivation, which is not clear from the results of the experiment but could possibly be identified with a longitudinal study.

6

Conclusion

The objective of this study has been to investigate how traceability management tools can be gamified, to what extent it could have an effect on user motivation and what negative aspects it could bring. This goal was pursued by utilizing the design science research approach. The first step was identifying which types of gamification elements to implement into the traceability management tool Capra. To select viable gamification features we analyzed previous studies including implemented gamification elements and by sending out a pilot survey to previous participants of an experiment which included the use of Capra. In the end, the level and badge feature were decided to be implemented and tested for this study. The testing of the selected gamification features was done through an experiment in which 24 individuals participated.

The data gathered from the experiments and the responses from the surveys gave insight into how the gamification features were perceived, it also revealed a number of issues and angles that should be pursued in future work.

The results showed that the levels and badge features had little to no effect on the speed and correctness of the vetting links. This was an expected result as the features did not change the the fundamental part of the vetting process, instead they were simple features which would give the participants some sort of progress and accomplishment.

The gamification elements we have chosen and implemented have had no significant impact on the speed or correctness of its system's users. However, the participants found more enjoyment in the gamified system.

If the results for speed and correctness had been significant, this would imply that gamifying our gamification elements are detrimental to those two qualities. However, our gamification elements are applicable if one seeks to create a more enjoyable system. As to directly answer **RQ1**, both badges and levels can be used to gamify a traceability management tool, if one seeks to increase enjoyment while using the system.

With regards to **RQ2** the participants were asked about the enjoyment of the process, the motivation to complete the task in general and whether the gamification elements assisted in motivating them. The results showed a clear indication that the gamification had a positive effect on motivation and it would become clearer with further improvements on the tool and the gamification features. When comparing

the enjoyment of the vetting task as perceived by the participants, the gamification features showed to make a significant difference. But when comparing the responses from the questions asking directly about motivation, the results showed to be insignificant. So further research is still required in order to confirm if these indications holds true.

As for **RQ3**, it is unclear whether there are disadvantages to adding gamification to traceability management tools, this would require further study as our results involving speed and correctness and usability were non-significant.

There are some inherent and very obvious disadvantages to gamification however, such as development costs and increased system complexity. But these are inherent as soon as you extend any system with any sort of feature.

The results also showed that different users prefer different gamification features. Some consider the competitive element important and want a clear sense of achievement, they want their results compared to that of others in order to know if their own result can be considered acceptable. Other participants are satisfied with the non-competitive elements currently present and express no desire in introducing such a feature. It could be of value to the research community to examine differently designed gamification systems applied to traceability management tools, specifically one that focuses on the competitive aspect.

Another interesting angle to be investigated is rather than simply adding gamification elements to a traceability tool, a complete gamification overhaul of the tool could potentially be done, the complete opposite of a simple gamification extension. With that said, too much focus put on the game elements might put less focus on the task at hand which could possibly have a negative effects on speed and correctness. Future studies needs to continuously consider and measure such factors.

This study has revealed the need for further study on the interaction between gamification and traceability management tools. Many questions are still left unanswered or remain in need of a more extensive study to verify some of the indications presented in the study.

Bibliography

- [1] Cleland-Huang, J., Gotel, O. C., Huffman Hayes, J., Mäder, P., Zisman, A. (2014, May). Software traceability: trends and future directions. In Proceedings of the on Future of Software Engineering (pp. 55-69). ACM.
- [2] Mader, P., Gotel, O., Philippow, I. (2009, August). Motivation matters in the traceability trenches. In Requirements Engineering Conference, 2009. RE'09. 17th IEEE International (pp. 143-148). IEEE.
- [3] Arkley, P., Mason, P., Riddle, S. (2002, September). Position paper: Enabling traceability. In Proceedings of the 1st International Workshop on Traceability in Emerging Forms of Software Engineering, Edinburgh, Scotland (September 2002) (pp. 61-65).
- [4] Hayes, J. H., Dekhtyar, A., Sundaram, S. K., Holbrook, E. A., Vadlamudi, S., April, A. (2007). REquirements TRacing On target (RETRO): improving software maintenance through traceability recovery. *Innovations in Systems and Software Engineering*, 3(3), 193-202.
- [5] Kong, W. K., Hayes, J. H., Dekhtyar, A., Dekhtyar, O. (2012, September). Process improvement for traceability: A study of human fallibility. In Requirements Engineering Conference (RE), 2012 20th IEEE International (pp. 31-40). IEEE.
- [6] Cuddeback, D., Dekhtyar, A., Hayes, J. H., Holden, J., Kong, W. K. (2011, May). Towards overcoming human analyst fallibility in the requirements tracing process (NIER Track). In Proceedings of the 33rd International Conference on Software Engineering (pp. 860-863). ACM.
- [7] Cleland-Huang, J., Gotel, O., Zisman, A. (2012). *Software and systems traceability* (Vol. 2, No. 3, pp. 7-8). Heidelberg: Springer.
- [8] Cleland-Huang, J., Berenbach, B., Clark, S., Settini, R., Romanova, E. (2007). Best practices for automated traceability. *Computer*, 40(6).
- [9] Dubois, D. J., Tamburrelli, G. (2013, August). Understanding gamification mechanisms for software development. In Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering (pp. 659-662). ACM.
- [10] Hamari, J., Koivisto, J., Sarsa, H. (2014, January). Does gamification work?—a literature review of empirical studies on gamification. In System Sciences (HICSS), 2014 47th Hawaii International Conference on (pp. 3025-3034). IEEE.
- [11] Deterding, S., Dixon, D., Khaled, R., Nacke, L. (2011, September). From game design elements to gamefulness: defining gamification. In Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments (pp. 9-15). ACM.

- [12] Pedreira, O., García, F., Brisaboa, N., Piattini, M. (2015). Gamification in software engineering—A systematic mapping. *Information and Software Technology*, 57, 157-168.
- [13] Kappen, D. L., Nacke, L. E. (2013, October). The kaleidoscope of effective gamification: deconstructing gamification in business applications. In *Proceedings of the First International Conference on Gameful Design, Research, and Applications* (pp. 119-122). ACM.
- [14] Parizi, R. M., Kasem, A., Abdullah, A. (2015, July). Towards gamification in software traceability: Between test and code artifacts. In *Software Technologies (ICSOFT), 2015 10th International Joint Conference on* (Vol. 1, pp. 1-8). IEEE.
- [15] Hanus, M. D., Fox, J. (2015). Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. *Computers Education*, 80, 152-161.
- [16] Clark, M. C., Rossiter, M. (2008). Narrative learning in adulthood. *New directions for adult and continuing education*, 2008(119), 61-70.
- [17] Kapp, K. M. (2012). *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley Sons.
- [18] Camilleri, V., Busuttil, L., Montebello, M. (2011). Social interactive learning in multiplayer games. In *Serious games and edutainment applications* (pp. 481-501). Springer, London.
- [19] Charles, D., Charles, T., McNeill, M., Bustard, D., Black, M. (2011). Game-based feedback for educational multi-user virtual environments. *British Journal of Educational Technology*, 42(4), 638-654.
- [20] DomíNquez, A., Saenz-De-Navarrete, J., De-Marcos, L., Fernández-Sanz, L., Pagés, C., MartíNez-HerráIz, J. J. (2013). Gamifying learning experiences: Practical implications and outcomes. *Computers Education*, 63, 380-392.
- [21] Mekler, E. D., Brühlmann, F., Opwis, K., Tuch, A. N. (2013, October). Do points, levels and leaderboards harm intrinsic motivation?: an empirical analysis of common gamification elements. In *Proceedings of the First International Conference on gameful design, research, and applications* (pp. 66-73). ACM.
- [22] Eickhoff, C., Harris, C. G., de Vries, A. P., Srinivasan, P. (2012, August). Quality through flow and immersion: gamifying crowdsourced relevance assessments. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval* (pp. 871-880). ACM.
- [23] Farzan, R., DiMicco, J. M., Millen, D. R., Brownholtz, B., Geyer, W., Dugan, C. (2008, April). When the experiment is over: Deploying an incentive system to all the users. In *symposium on persuasive technology*.
- [24] Farzan, R., DiMicco, J. M., Millen, D. R., Dugan, C., Geyer, W., Brownholtz, E. A. (2008, April). Results from deploying a participation incentive mechanism within the enterprise. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 563-572). ACM.
- [25] Grant, S., Betts, B. (2013, May). Encouraging user behaviour with achievements: an empirical study. In *Proceedings of the 10th Working Conference on Mining Software Repositories* (pp. 65-68). IEEE Press.

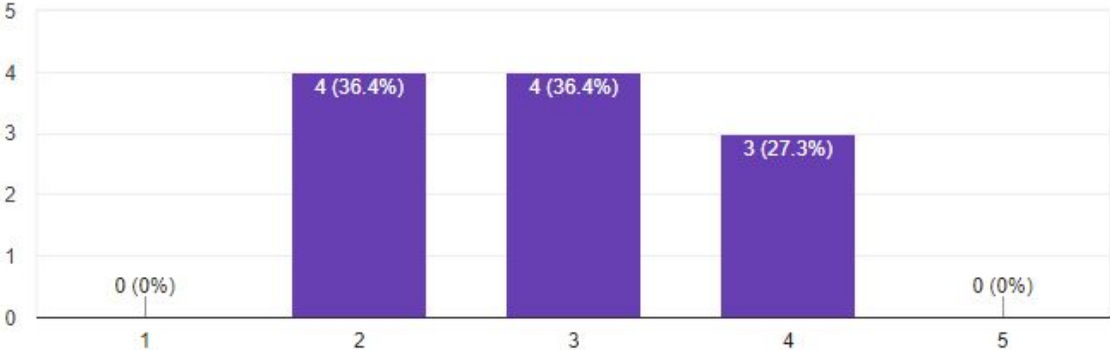
-
- [26] Passos, E. B., Medeiros, D. B., Neto, P. A., Clua, E. W. (2011, November). Turning real-world software development into a game. In *Games and Digital Entertainment (SBGAMES), 2011 Brazilian Symposium on* (pp. 260-269). IEEE.
- [27] Flatla, D. R., Gutwin, C., Nacke, L. E., Bateman, S., Mandryk, R. L. (2011, October). Calibration games: making calibration tasks enjoyable by adding motivating game elements. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (pp. 403-412). ACM.
- [28] Hamari, J., Koivisto, J. (2013). Social motivations to use gamification: an empirical study of gamifying exercise.
- [29] Farzan, R., Brusilovsky, P. (2011). Encouraging user participation in a course recommender system: An impact on user behavior. *Computers in Human Behavior*, 27(1), 276-284.
- [30] Korn, O., Funk, M., Schmidt, A. (2015, June). Towards a gamification of industrial production: a comparative study in sheltered work environments. In *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (pp. 84-93). ACM.
- [31] Von Alan, R. H., March, S. T., Park, J., Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 28(1), 75-105.
- [32] Vaishnavi, V., Kuechler, W. (2004). Design research in information systems.
- [33] Peffers, K., Tuunanen, T., Rothenberger, M. A., Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3), 45-77.
- [34] Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 4-7.
- [35] Bangor, A., Kortum, P., Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3), 114-123.
- [36] Deci, E. L., Koestner, R., Ryan, R. M. (1999). A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. *Psychological bulletin*, 125(6), 627.
- [37] Kloos, C. D., Muñoz-Merino, P. J., Alario-Hoyos, C., Ayres, I. E., Fernández-Panadero, C. (2015, March). Mixing and blending MOOC Technologies with face-to-face pedagogies. In *Global Engineering Education Conference (EDUCON), 2015 IEEE* (pp. 967-971). IEEE.

A

Appendix 1 - pilot survey

Do you think that the current process for vetting trace links (as used in the experiment) is enjoyable?

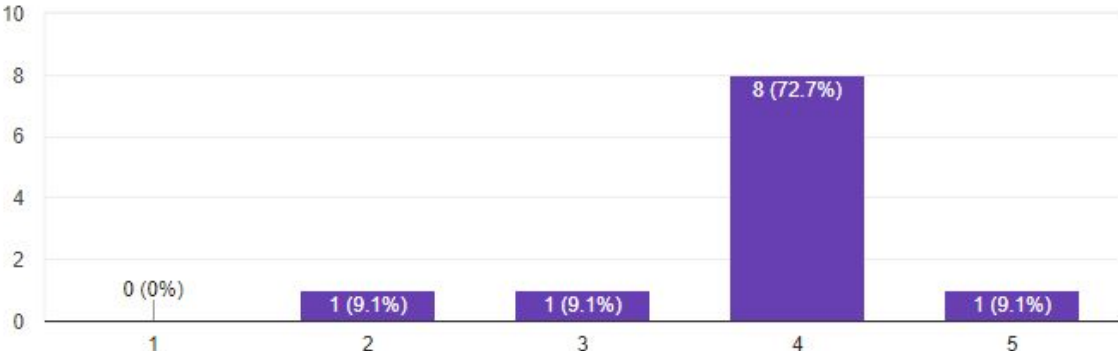
11 responses



Progress Bar

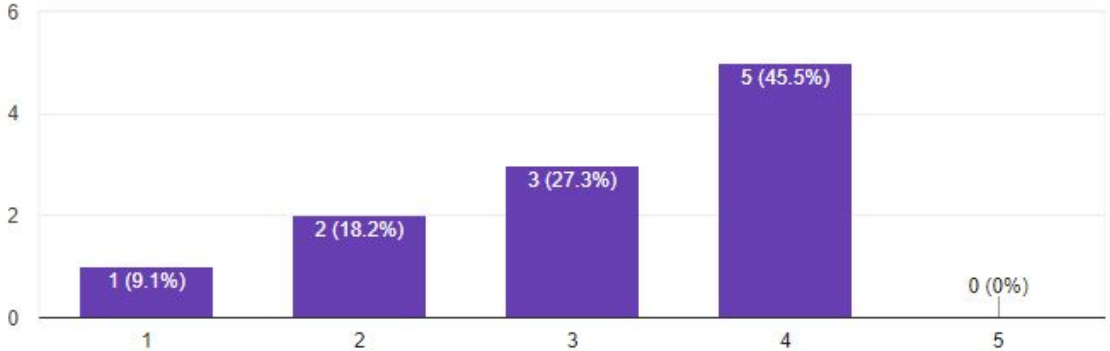
Do you believe this functionality would make the task of verifying traceability links more satisfying?

11 responses



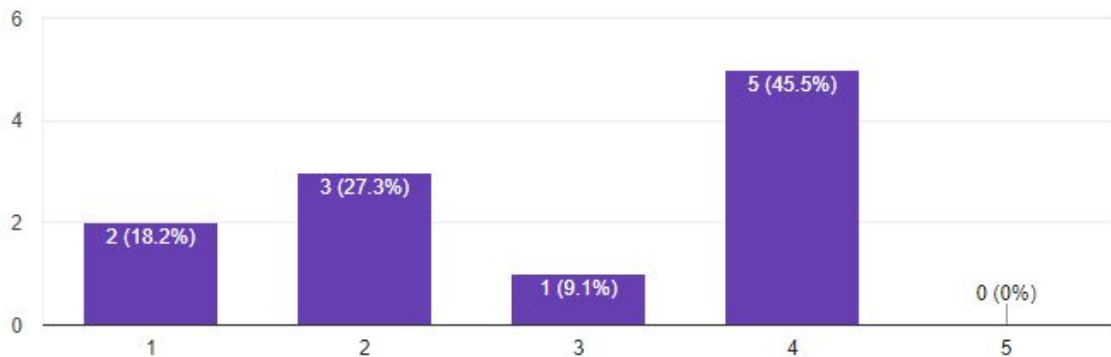
Do you believe this functionality would help in decreasing the time spent on each link?

11 responses



Do you believe this functionality would make you focus less on verifying the links correctly and focus more on verifying as many as possible?

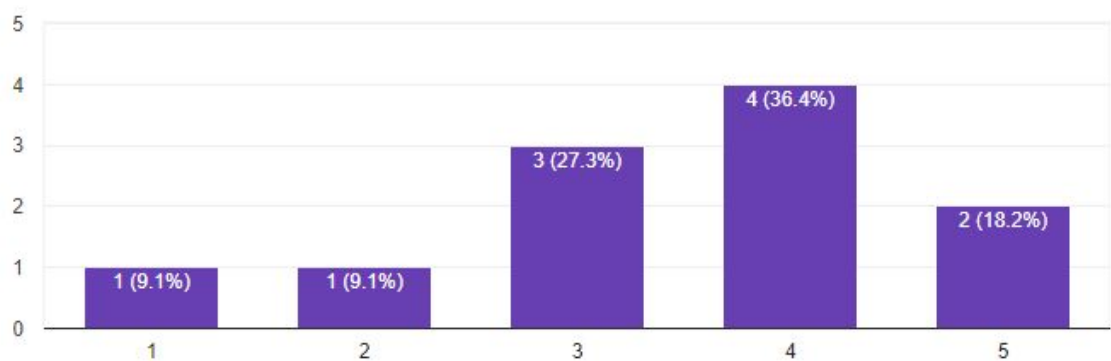
11 responses



Given this functionality, do you believe you would skip the more advanced links and go for easier ones that could potentially fill up the bar quicker?



11 responses



If you want to clarify any of your answers given above, feel free to do so here.

3 responses

It's usually nice to see progress being made as you work.

If its a lot of links, the progress-bar will be discouraging, hence will make me find a way to fill it up quickly

It would be nice to have an indicator the correctness of the verified links at the end of the process.

Do you think a progress bar could have any negative effects? If yes, which ones?

5 responses

On the other hand, if the number of links to verify is very large, it may be disheartening to see little to no effect on the progressbar as you verify links

I would be less motivated to finish if I don't see movement response as I finish links. So it should be mapped to something that moves quicker

can be discouraging and make one loose focus when there are a lot of links to connect.

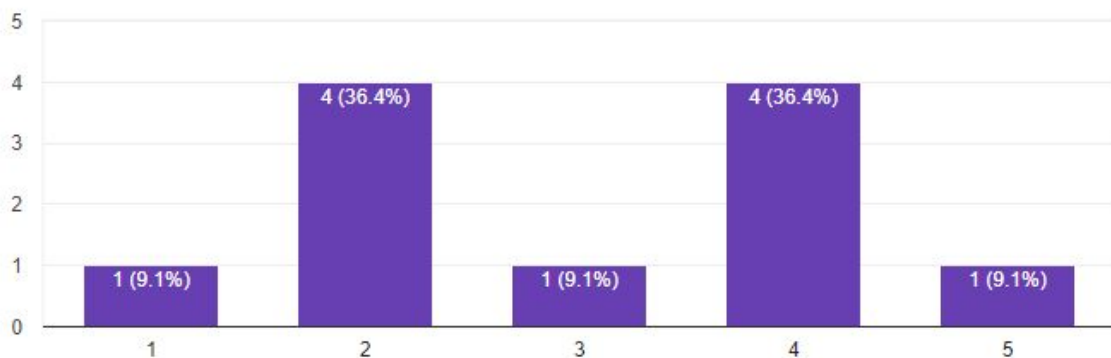
I think that the progress bar would not affect the verification process.

Motivating one to fill up the bar. Personally I wouldn't do that because it really depends on the nature of my project and how important I view the links. The progress bar is just a good way to help know how far I am from accomplishing my task

Leader board

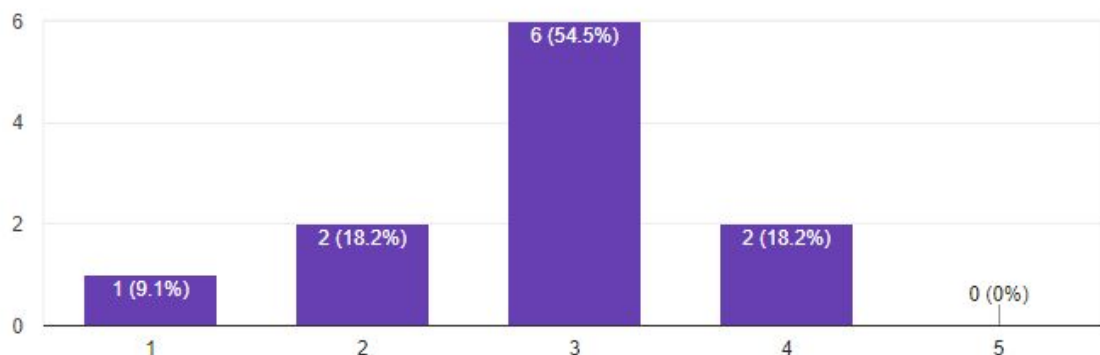
Do you believe this functionality would make the task of verifying traceability links more satisfying?

11 responses



Do you believe this functionality would help in decreasing the time spent on each link?

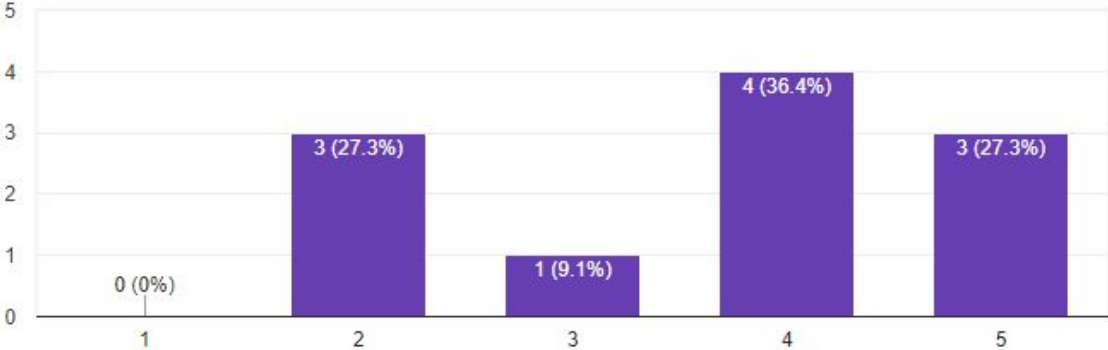
11 responses



Do you believe this functionality would make you focus less on verifying the links correctly and focus more on verifying as many as possible?

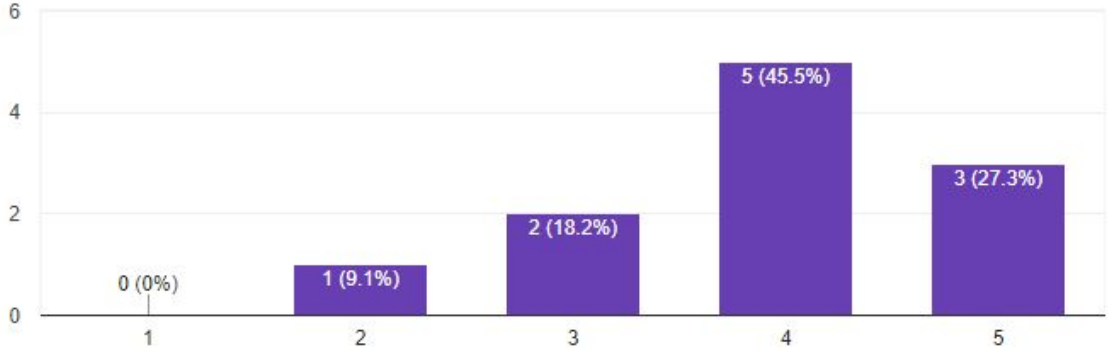


11 responses



Given this functionality, do you believe you would skip the more advanced links and go for easier ones in order to quickly rise on the leaderboard?

11 responses



Do you think a leader board could have any negative effects? If yes, which ones?

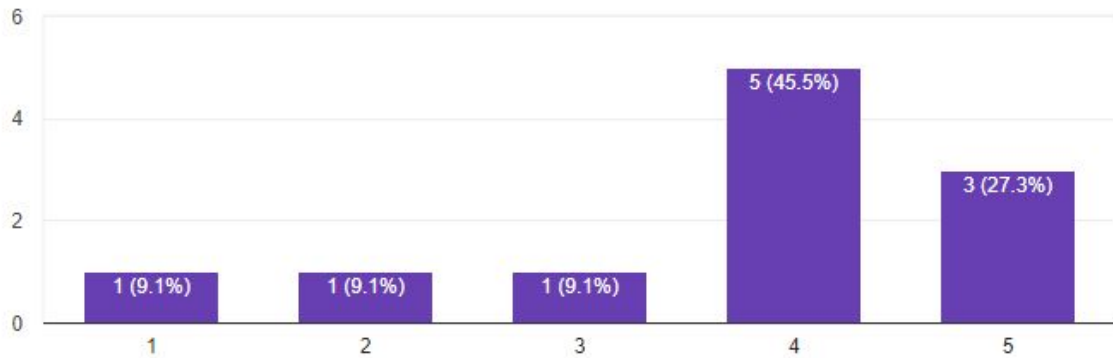
5 responses

- Personally, I think, I could be demotivated by such a leader board. The competition does not motivate me so much. Might work well on other people.
- discourages if you are at the bottom of the board.
- It would be fun to have the leader board
- Fear of embarrassment might lead to quick but poor quality work. But it may also be positive in that people may verify links
- If there is a runaway leader, in an unbeatable lead, that could discourage others.

Levels

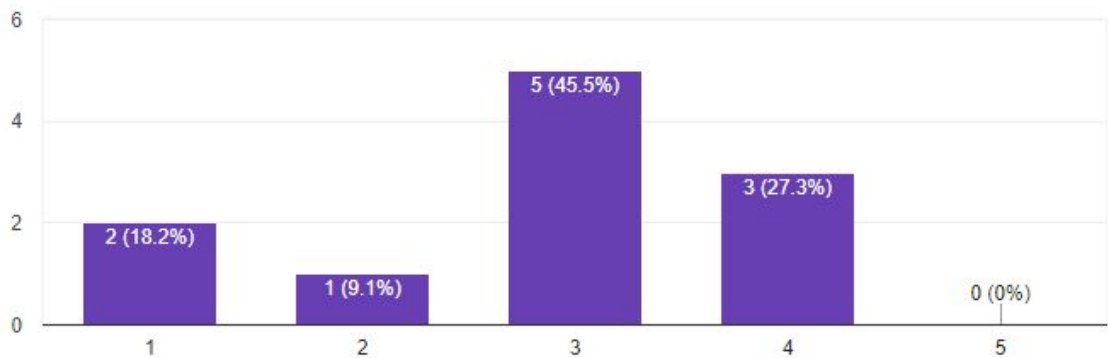
Do you believe this functionality would make the task of verifying traceability links more satisfying?

11 responses



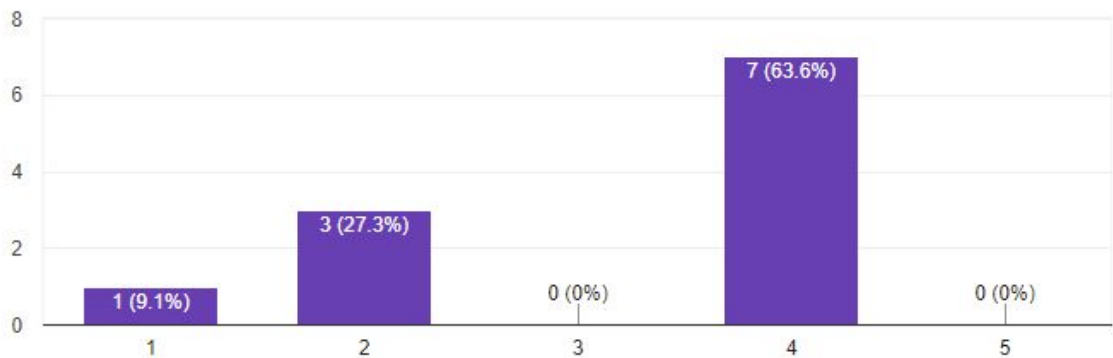
Do you believe this functionality would help in decreasing the time spent on each link?

11 responses



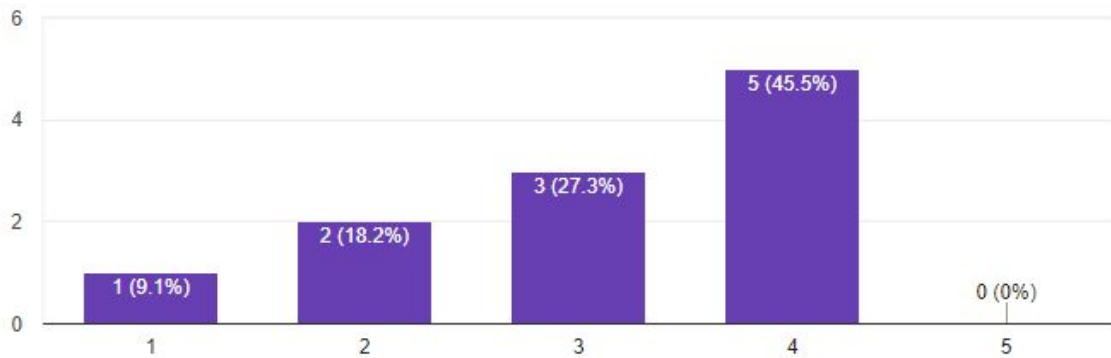
Do you believe this functionality would make you focus less on verifying the links correctly and focus more on verifying as many as possible?

11 responses



Given this functionality, do you believe you would skip the more advanced links and go for easier ones that could potentially make you level up faster?

11 responses



If you want to clarify any of your answers given above, feel free to do so here.

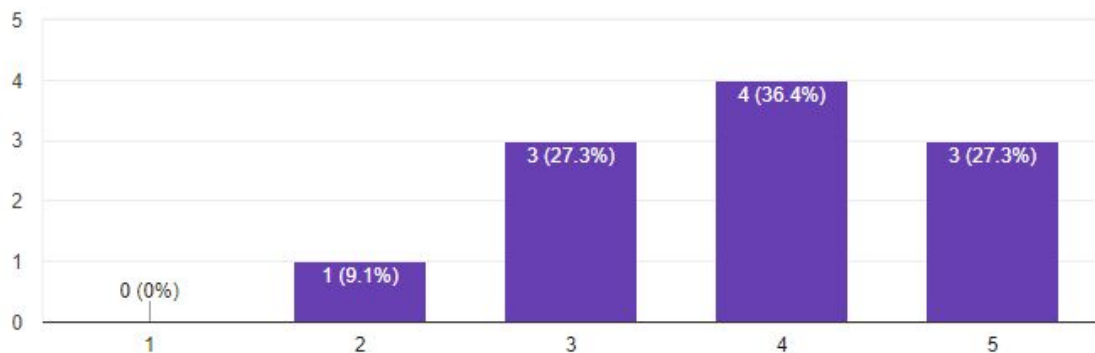
1 response


You can avoid skipping difficult ones by rewarding those with more "levels" earned.

Badges

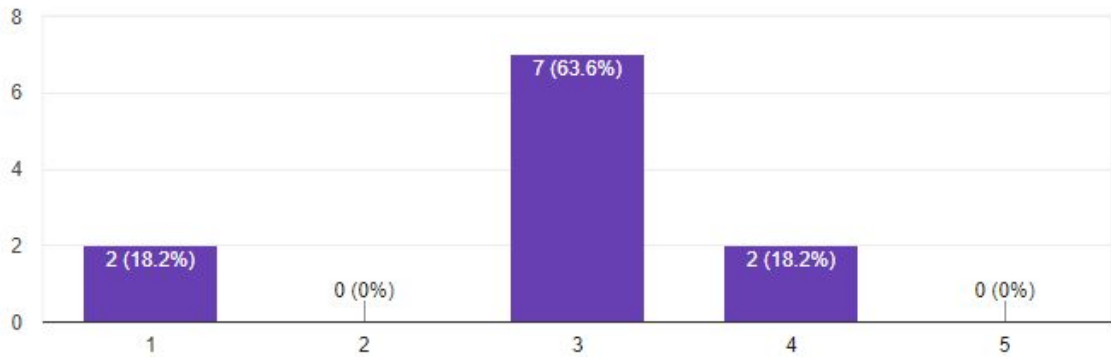
Do you believe this functionality would make the task of verifying traceability links more satisfying?


11 responses



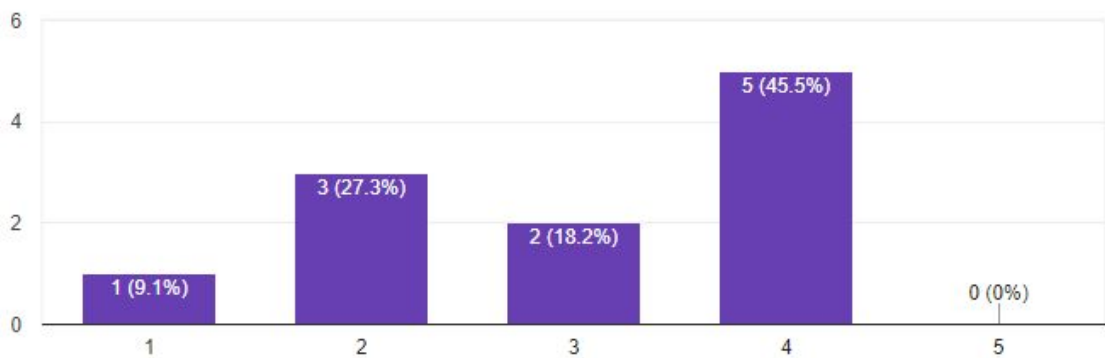
Do you believe this functionality would help in decreasing the time spent on each link? 


11 responses



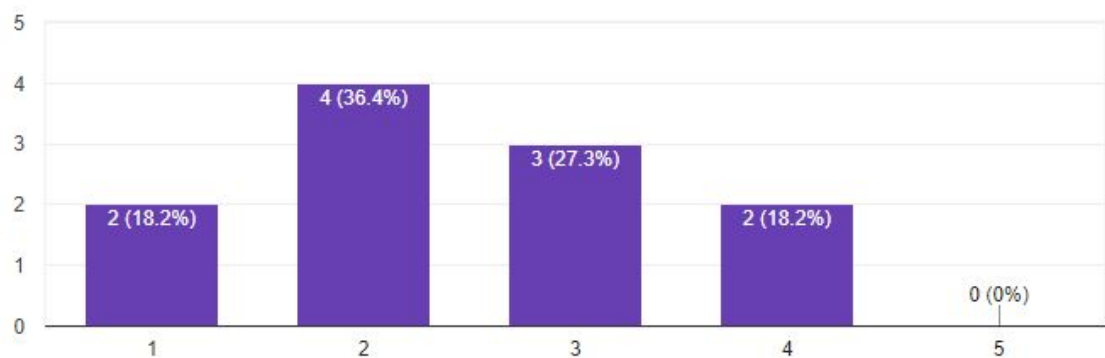
Do you believe this functionality would make you focus less on verifying the links correctly and focus more on verifying as many as possible? 

11 responses



Given this functionality, do you believe you would skip the more advanced links and go for easier ones that could potentially give you more badges? 

11 responses



Extra feedback

Extra feedback?

1 response

If the goal is to have correct tracelinks, I think all of the mentioned features have the same problem, of focusing not on correctness itself, but on the number of processed links, built-in.

B

Appendix 2 - GG survey

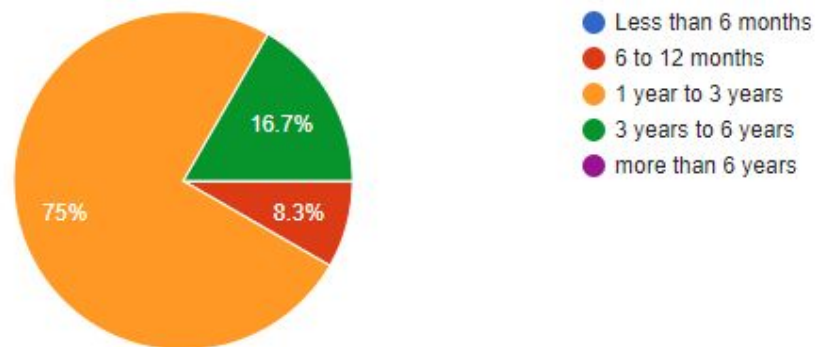
Reading the graphs: All questions/graphs which has a scale on 1-5 in this appendix should be read as 1 standing for “Strongly disagree” and 5 standing for “Strongly agree”. There’s only one exception, but the scale is explained in that particular instance.

Pre-experiment questions

What is your current work situation?
Junior Developer
Bachelor student
Bachelor Student SEM
Second line support
Bachelor Student
Mid-level Developer
Bachelor Student, Junior Developer
System Designer
Master Student
Master Student
Junior developer
Master

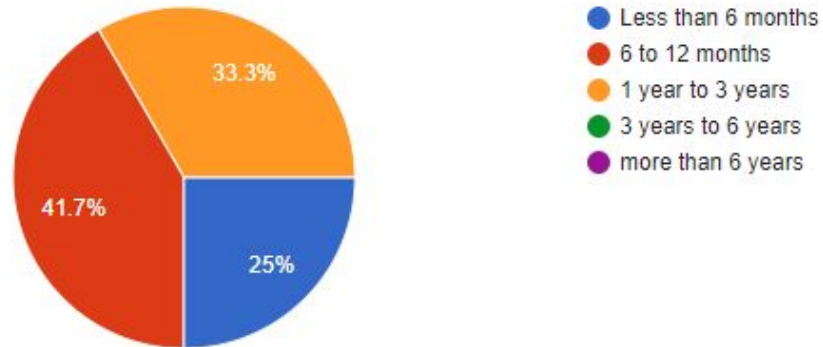
What is your experience in software development? (If you are a student, include the number or months developing software as a student)

12 responses



What is your experience in using Eclipse for software development? (If you are a student, include the number or months using Eclipse as a student)

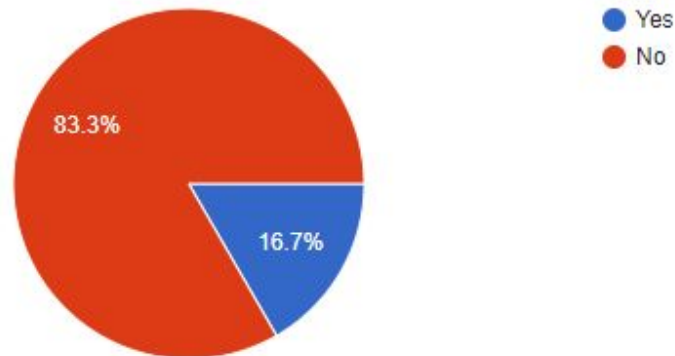
12 responses



Post-experiment questions

Do you have experience with developing software similar to MedFleet (Software that involves the use of drones and route planning)?

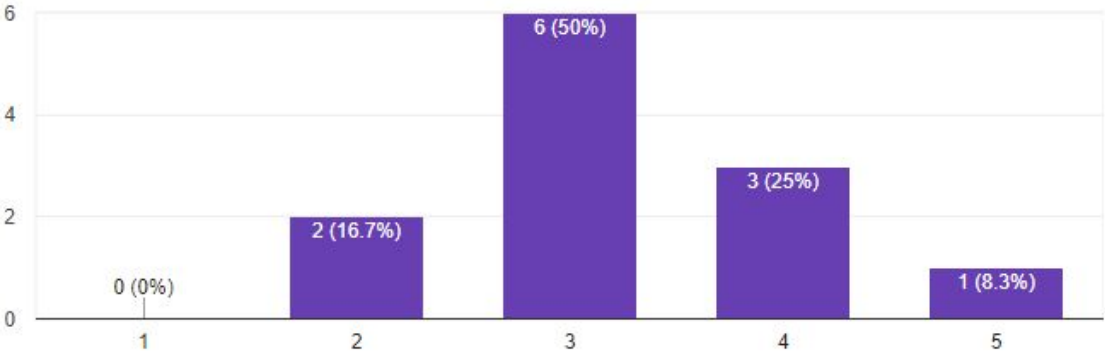
12 responses



On a scale of 1 – 5 (where 1 is “I completely did not understand what the system does” and 5 “I completely understood what the system does”), how confident were you with your understanding of the MedFleet system?

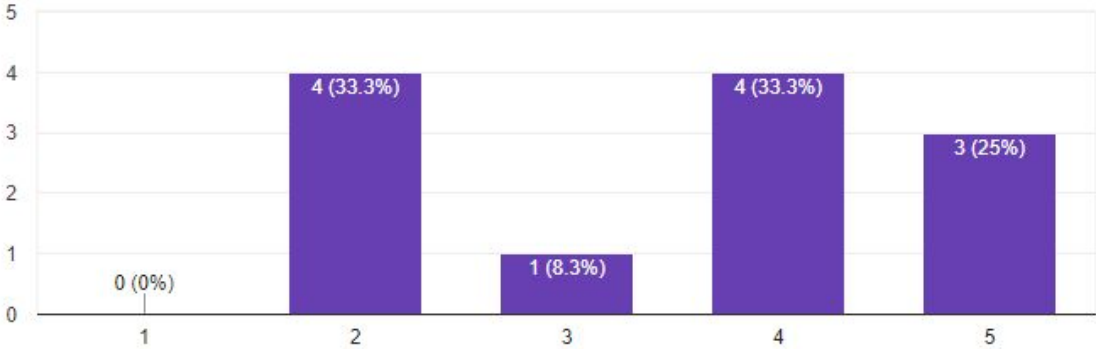


12 responses



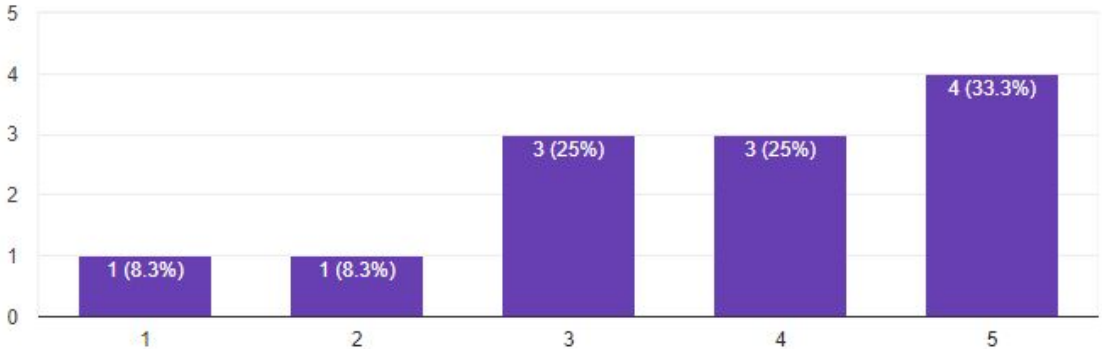
The current process for vetting trace links (as used in the experiment) is enjoyable?

12 responses



I felt motivated to complete the task

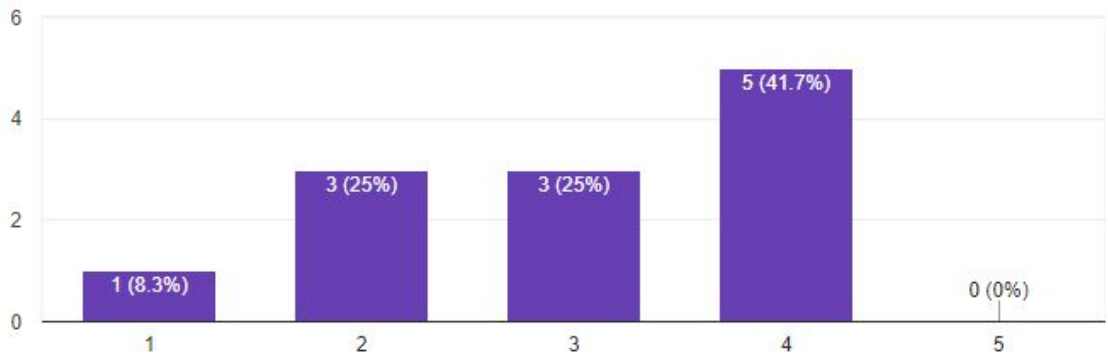
12 responses



System Usability Scale

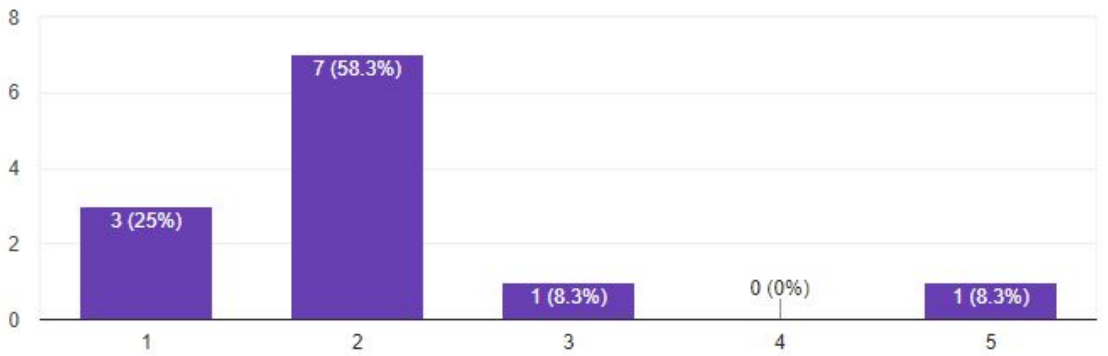
I think that I would like to use this system frequently

12 responses



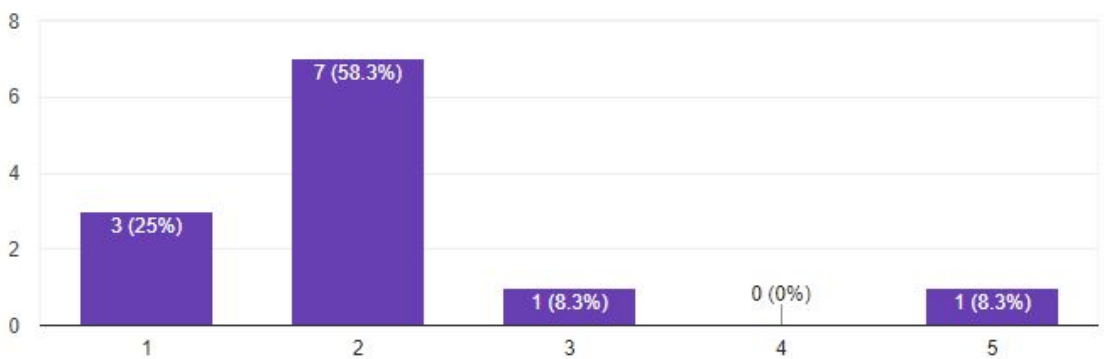
I found the system unnecessarily complex

12 responses



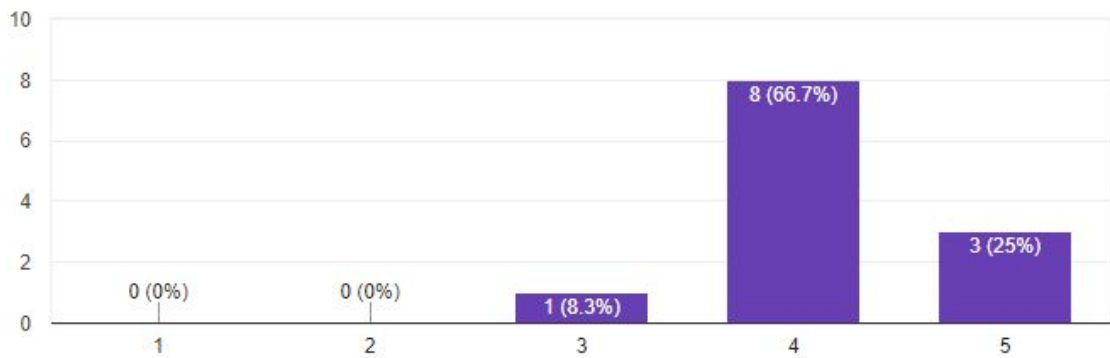
I found the system unnecessarily complex

12 responses



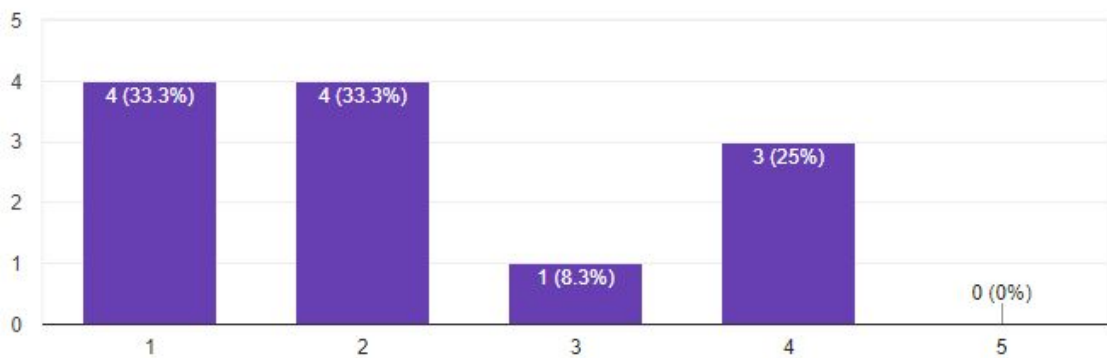
I thought the system was easy to use

12 responses



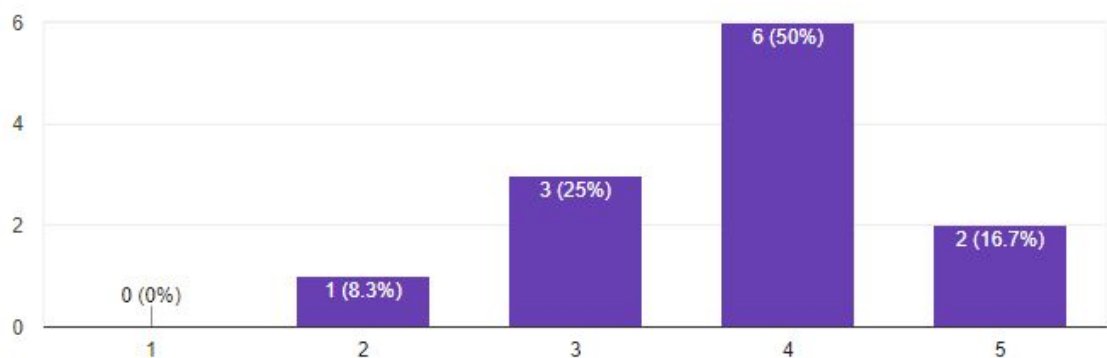
I think that I would need the support of a technical person to be able to use this system

12 responses



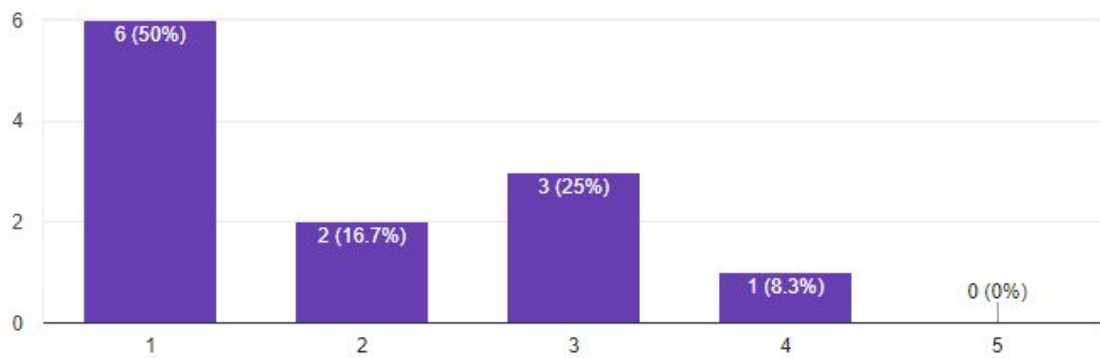
I found the various functions in this system were well integrated

12 responses



I thought there was too much inconsistency in this system

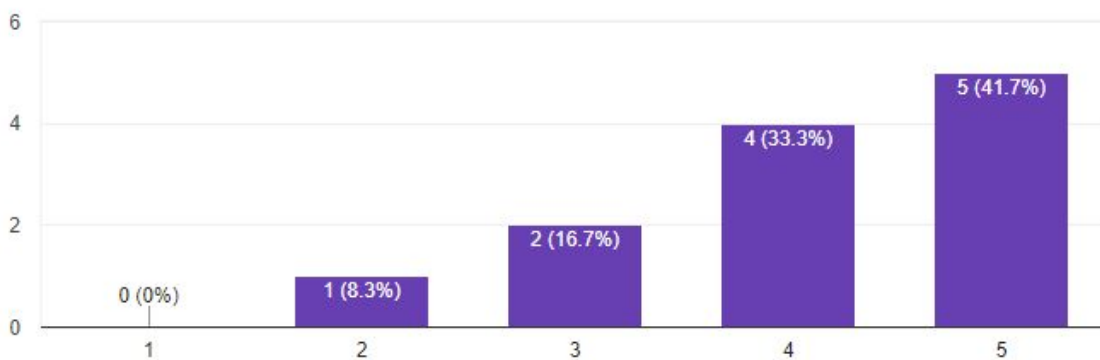
12 responses



I would imagine that most people would learn to use this system very quickly

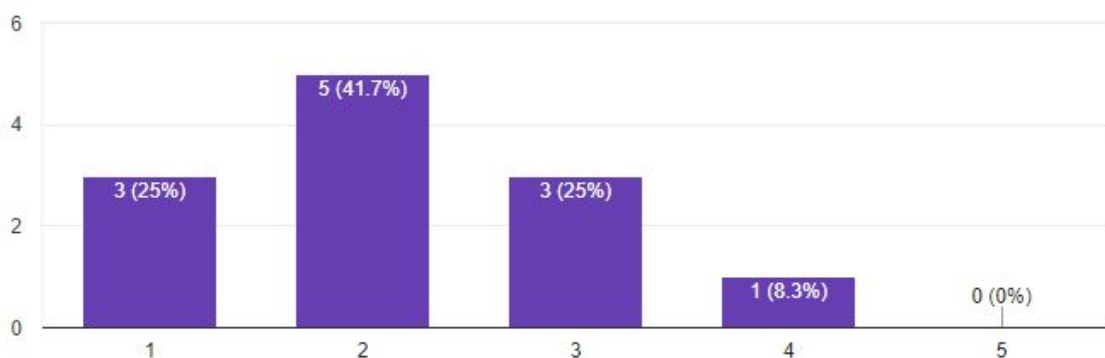


12 responses



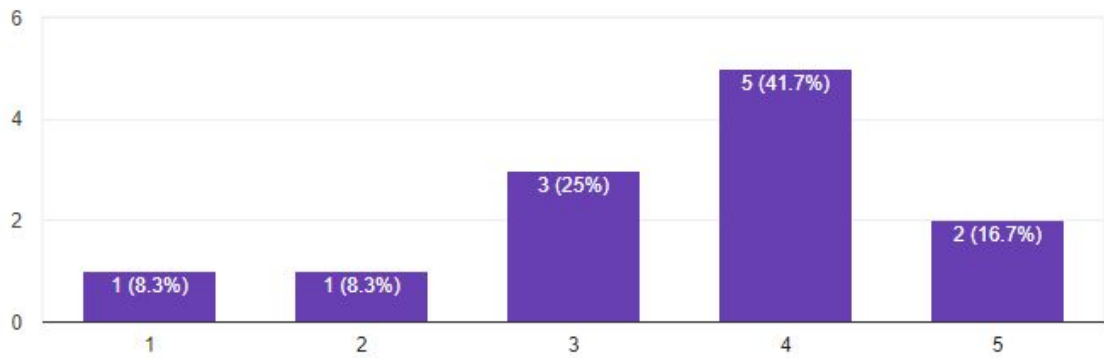
I found the system very cumbersome to use

12 responses



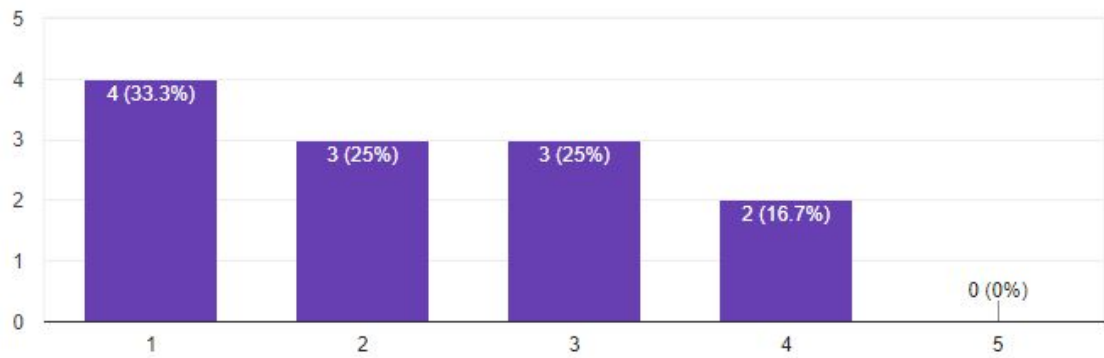
I felt very confident using the system

12 responses



I needed to learn a lot of things before I could get going with this system

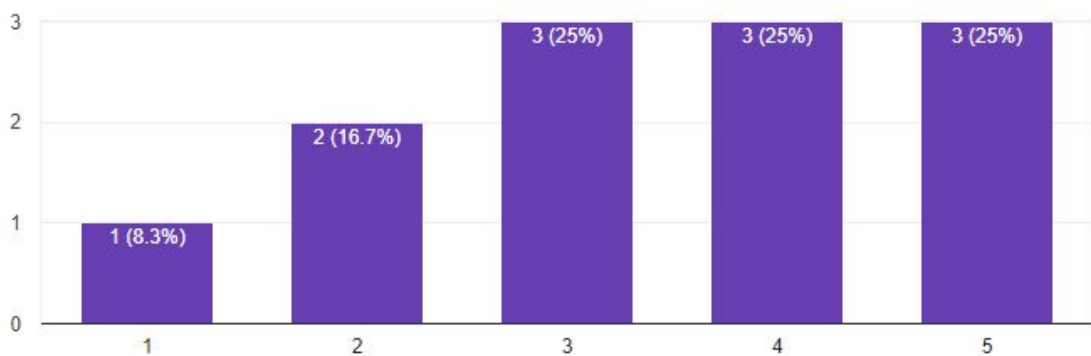
12 responses



Levels

I had no issue understanding the levels.

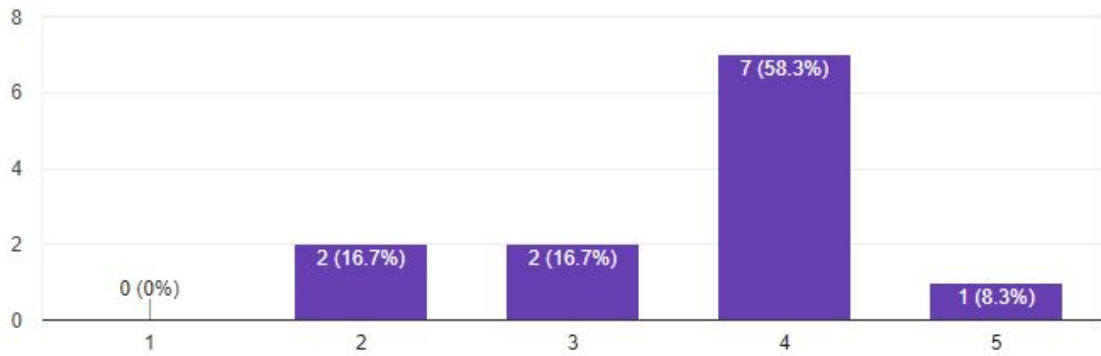
12 responses



The levels made the task of vetting trace links satisfying.



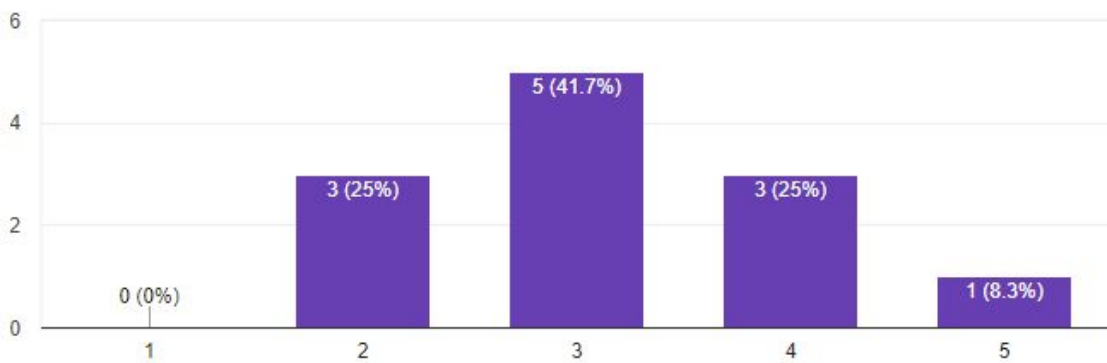
12 responses



Do you think that the levels helped you decrease the time spent on each link?



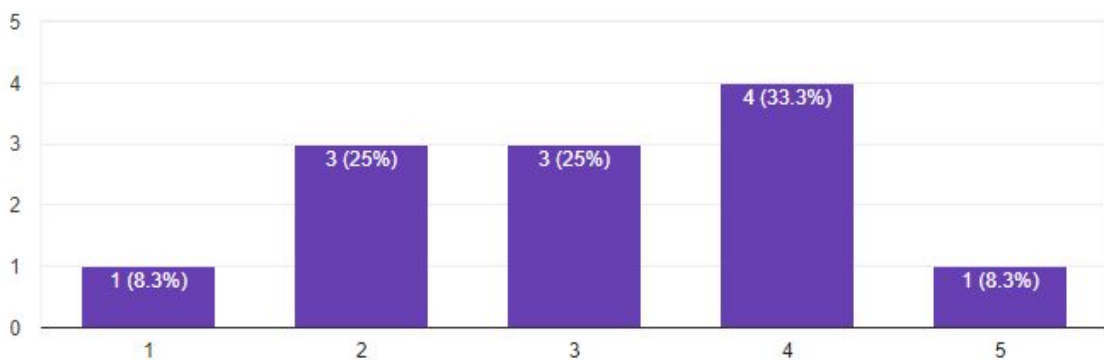
12 responses



Did the levels make you focus more on verifying as many links as possible instead of verifying each link correctly?



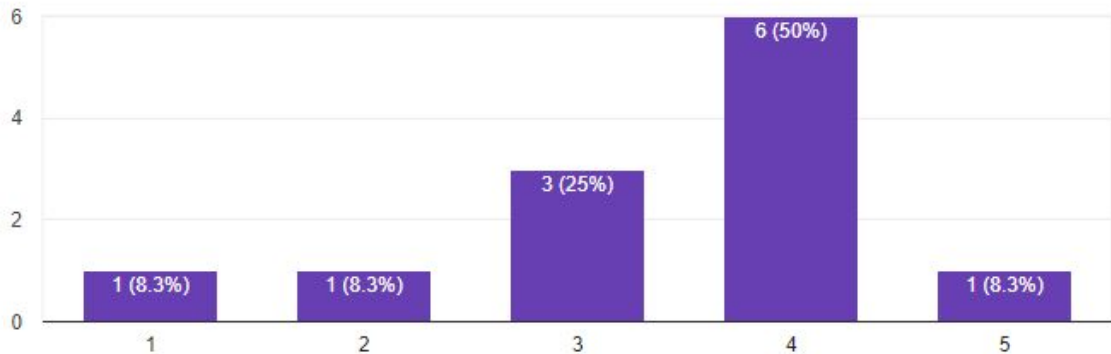
12 responses



I felt that the levels contributed towards being motivated to complete the task.



12 responses



If you didn't feel that the levels made you more motivated, can you identify a reason as to why?

3 responses

takes too long

It was difficult to tie them to any sense of progress. Partly I would say this is because there is no sense of achievement or comparison. Is level 20 good? I'm not sure.

No not really, it motivated me a little but due to the UI (Divided attention) i did not focus as much on lvls

Can you identify any negative effects of the level system that affected your ability to work on your task?

7 responses

Some people only wants to level up and doesn't care if they are wrong

It might have decreased the correctness of my verified links

There was a max cap

hungry and tired

It was not something that I really paid attention to. In this sense I was more motivated by trying to do the task as best as possible.

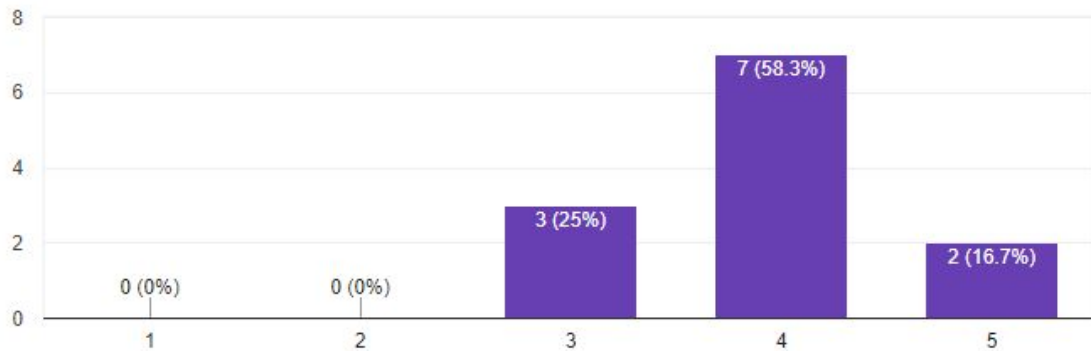
In this particluar test I knew that my performance wasn't measured to the real variables. However I do also understand that in real life, this is not an option.

No

Overall, the levels feature was a good addition to the traceability tool.



12 responses



Feel free to add any additional comments on the levels feature.

4 responses

I think it made it more fun since the verification might have felt a bit more like a chore without it

It's a start. This is a somewhat tedious task so I see the value in making it more interesting but I definitely feel as if I need some more investment in the levels. Either competition or some tangible reward (even an upgrade in cosmetics, such as extra chrome or a visualisation of my progress so far)

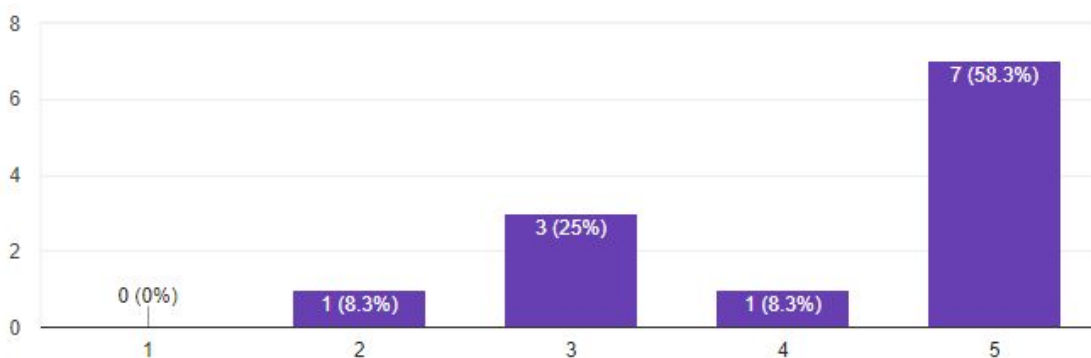
I didnt understand the point system, but since my level increased steadily I didnt care.

see next page

Badges

I had no issue understanding the badges.

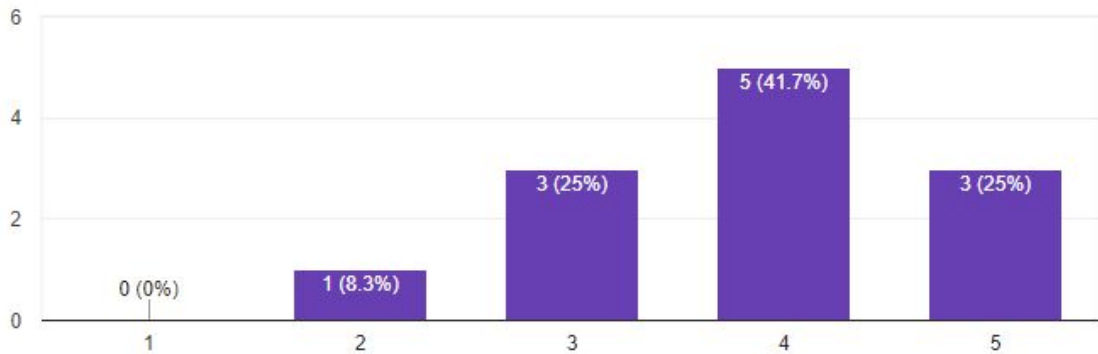
12 responses



The badges made the task of vetting trace links satisfying.

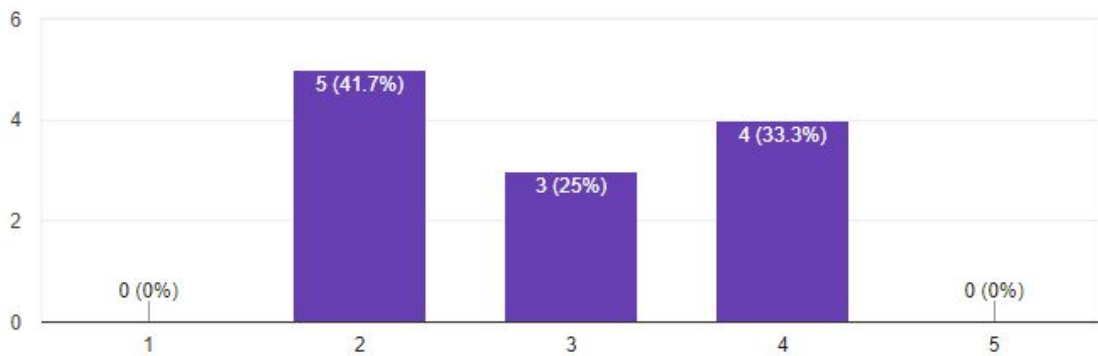


12 responses



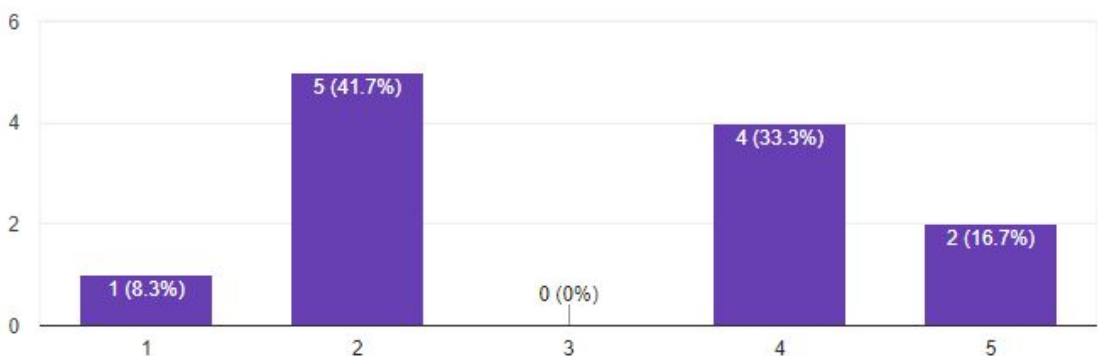
Do you think that the badges helped you decrease the time spent on each link?

12 responses



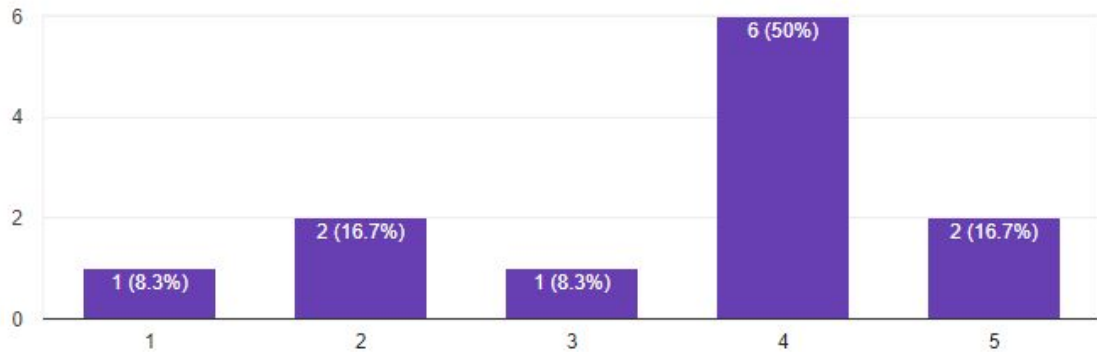
Did the badges make you focus more on verifying as many links as possible instead of verifying each link correctly?

12 responses



I felt that the badges contributed towards being motivated to complete the task.

12 responses



If you didn't feel that the badges made you more motivated, can you identify a reason as to why?

4 responses

not badge upgrades

They topped out very early and were not super visually interesting

I didnt realize there were badges.

Like I wrote before with the UI (divided attention)

Can you identify any negative effects of the badge system that affected your ability to work on your task?

6 responses

Some people only wants to get all the badges and will do so they gain them as fast as possible and doesn't care if they select the correct links

Their effect disappeared when all badges where achieved

Eagerness to answer

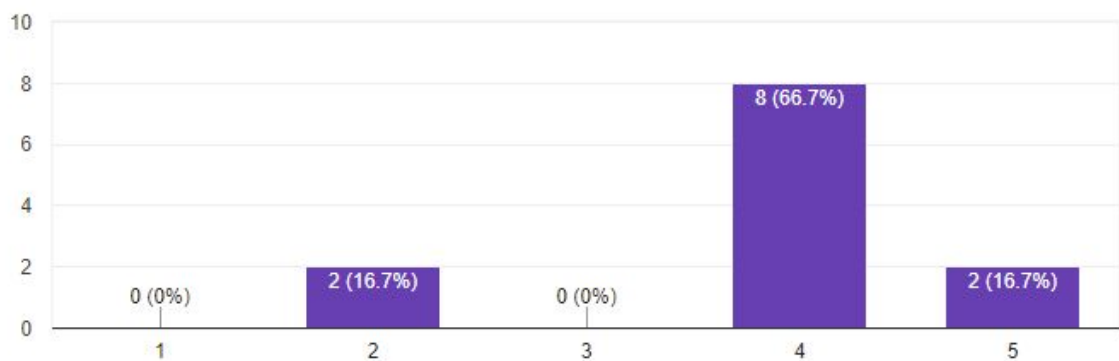
max cap on badged

No

no it was down in the corner so i did not focus on it

Overall, the badges feature was a good addition to the traceability tool.

12 responses



Feel free to add any additional comments on the badges feature.

3 responses

Nicely designed

Similarly to the levels, there needs to be some sort of comparison and more visual interest for the badges to be effective

levels + badge feature, make these a little bit more interactive, its hard to notice them in the corner

Can you think of a feature that is not in the tool, but that you think would be useful when vetting traceability links?

4 responses

nope

Visualisation of both my progress (levels are a start but if you are a completionist you also need a target) and maybe some visualisation of the links made so far

Not having the AI's prediction ratio would be nice, to give the "linker" a bigger inclination of not sneering at them and taking them into consideration when making their decision.

no

C

Appendix 3 - CG survey

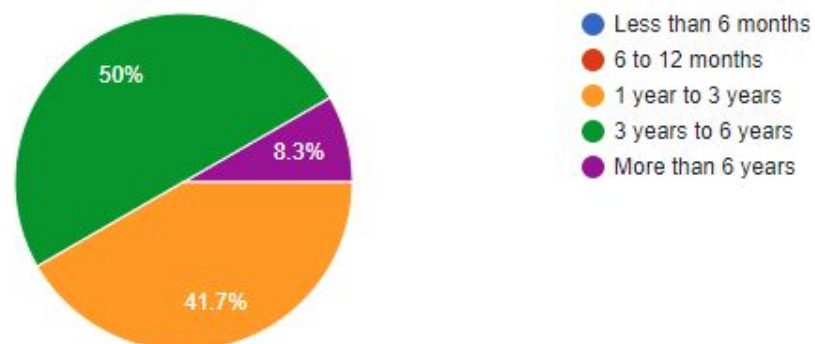
Reading the graphs: All questions/graphs which has a scale on 1-5 in this appendix should be read as 1 standing for “Strongly disagree” and 5 standing for “Strongly agree”. There’s only one exception, but the scale is explained in that particular instance.

Pre-experiment questions

What is your current work situation?
Bachelor degree
Master student
Software Developer + MSc Student
Junior Developer
Master Student
Master Student
Master Student
Developer
Junior developer
Master Student
Master student
Junior Developer

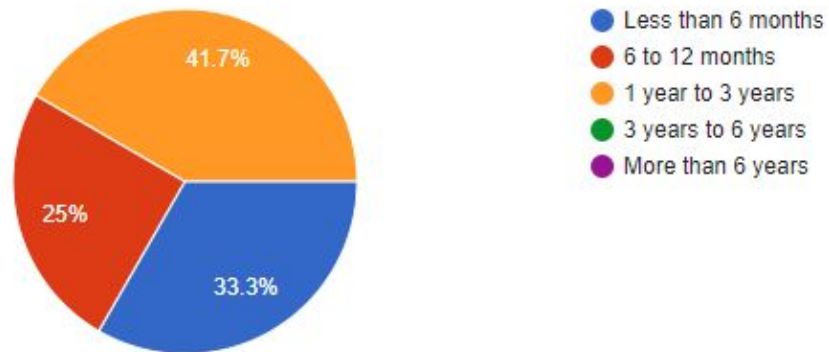
What is your experience in software development? (If you are a student, include the number or months developing software as a student)

12 responses



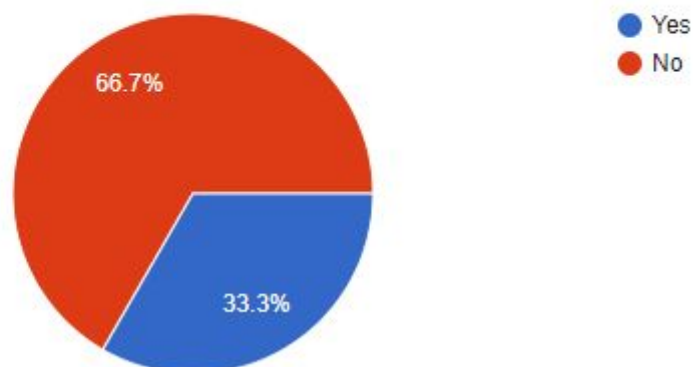
What is your experience in using Eclipse for software development? (If you are a student, include the number or months using Eclipse as a student)

12 responses



Do you have any experience with software traceability?

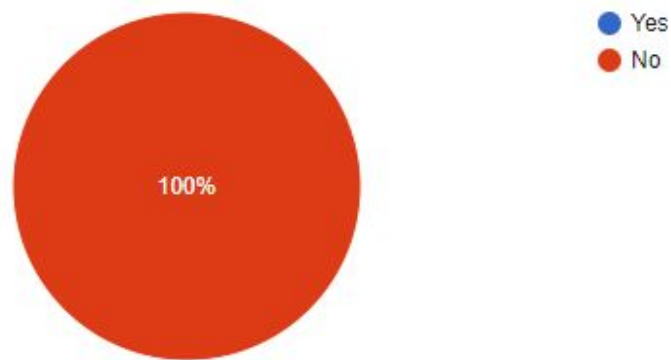
12 responses



Post-experiment questions

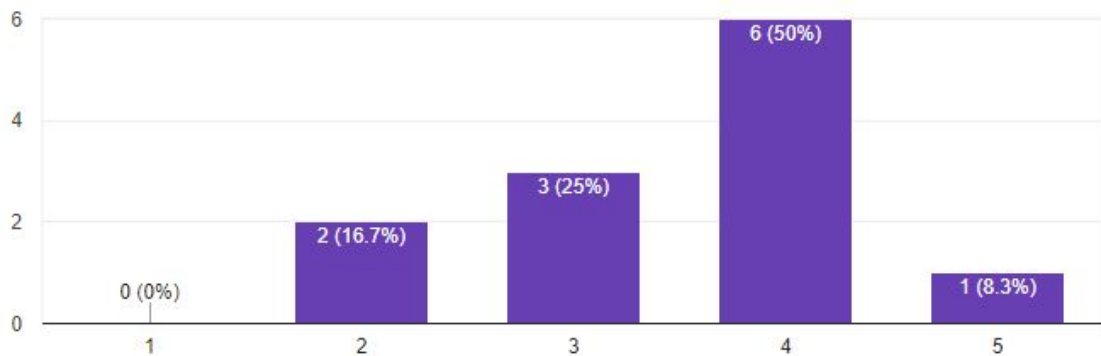
Do you have experience with developing software similar to MedFleet (Software that involves the use of drones and route planning)?

12 responses



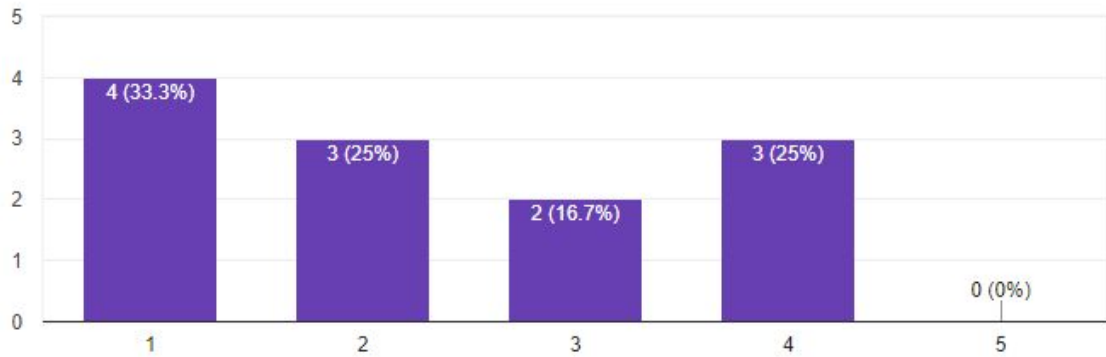
On a scale of 1 – 5 (where 1 is "I completely did not understand what the system does" and 5 "I completely understood what the system does"), how confident were you with your understanding of the MedFleet system?

12 responses



The current process for vetting trace links (as used in the experiment) is enjoyable?

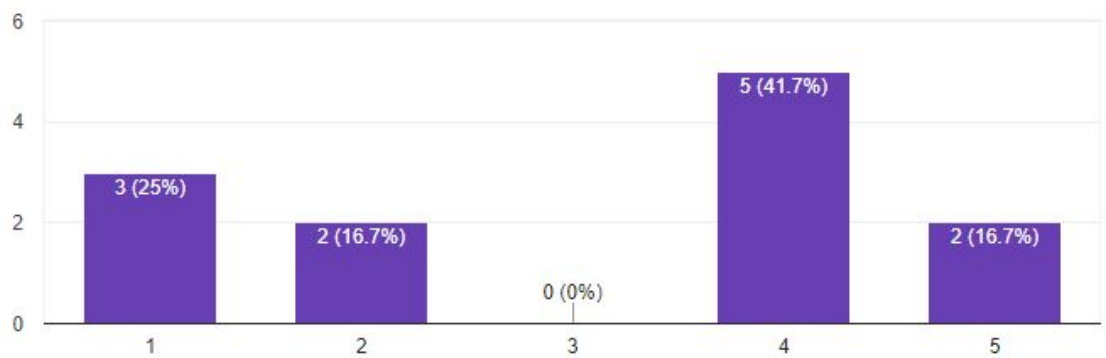
12 responses



I felt motivated to complete the task



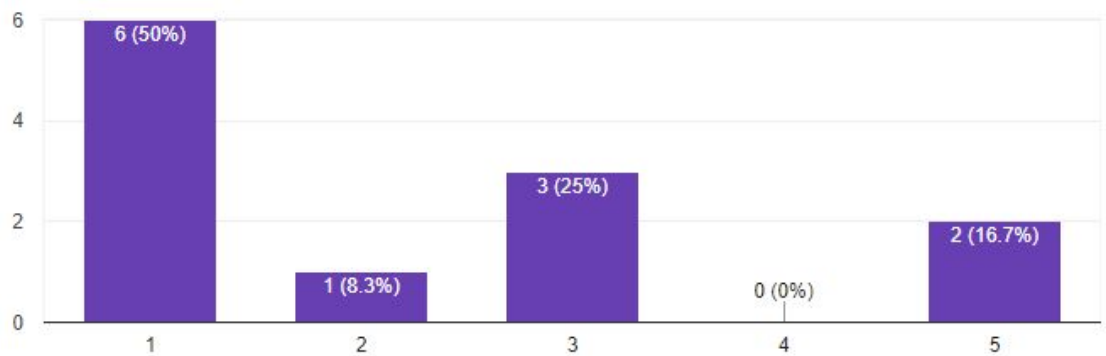
12 responses



System Usability Scale

I think that I would like to use this system frequently

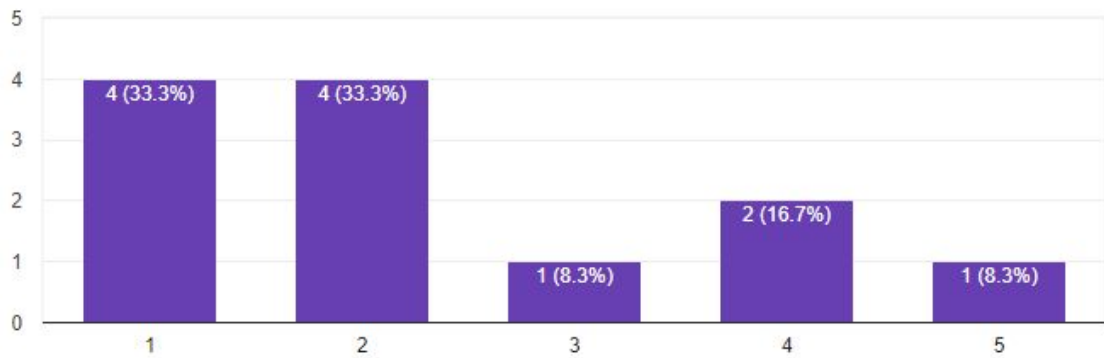
12 responses



I found the system unnecessarily complex

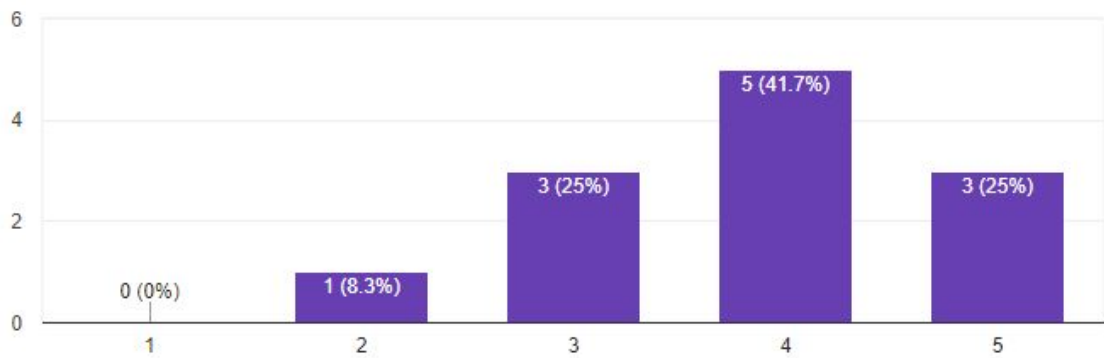


12 responses



I thought the system was easy to use

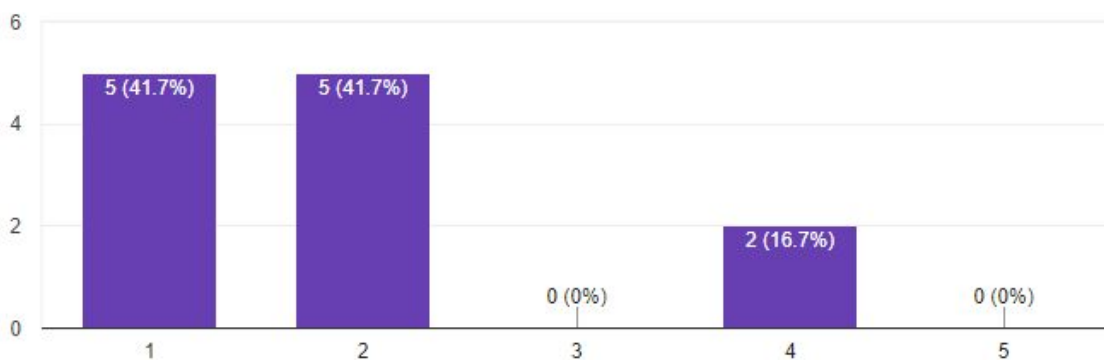
12 responses



I think that I would need the support of a technical person to be able to use this system



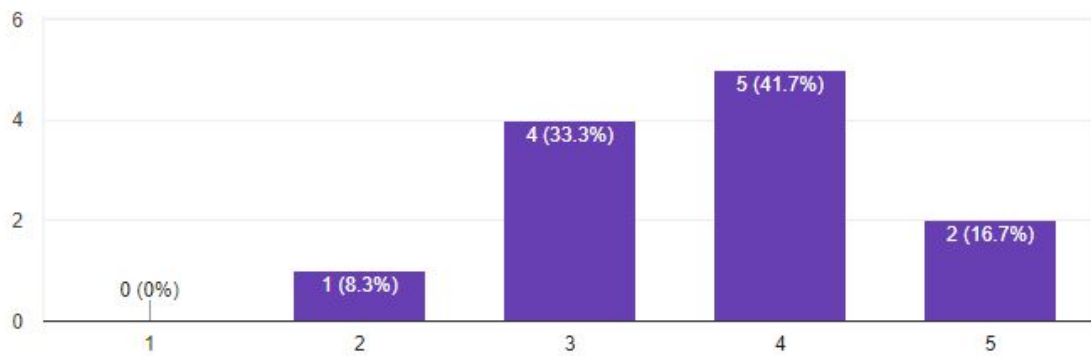
12 responses



I found the various functions in this system were well integrated



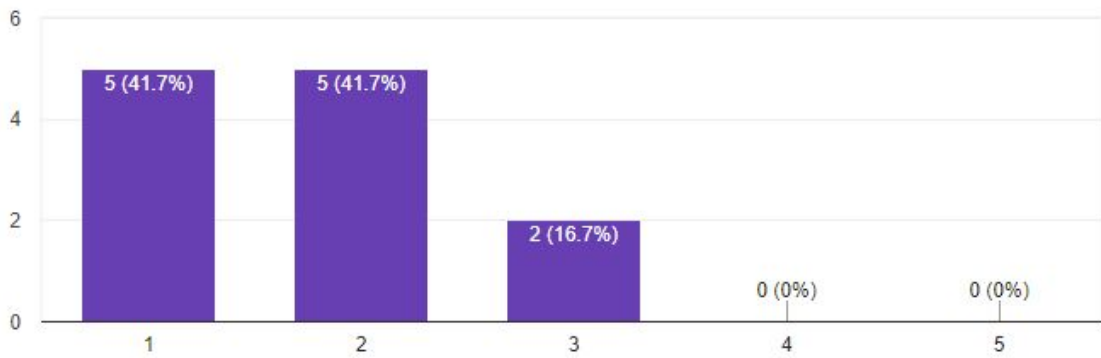
12 responses



I thought there was too much inconsistency in this system

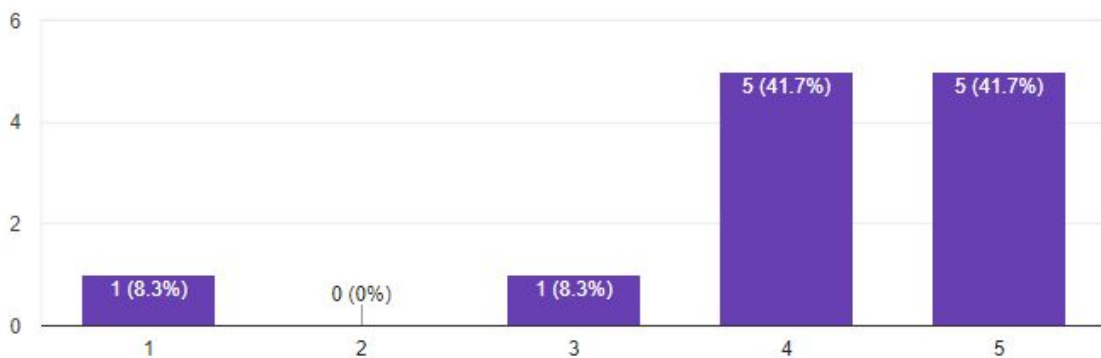


12 responses



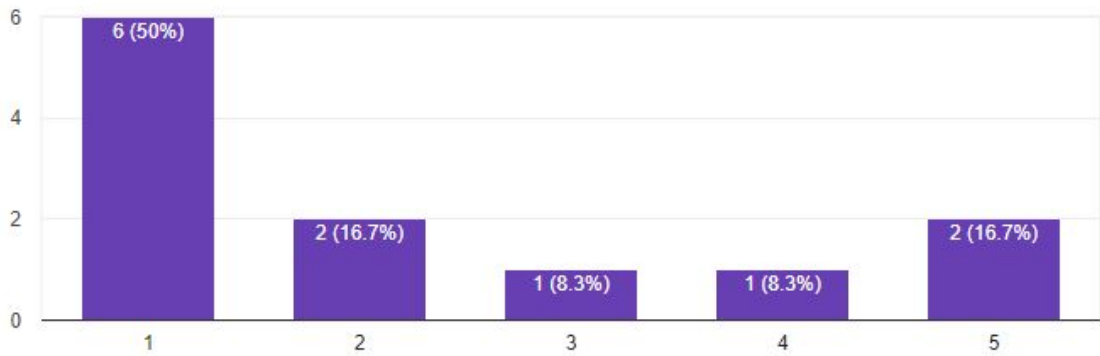
I would imagine that most people would learn to use this system very quickly

12 responses



I found the system very cumbersome to use

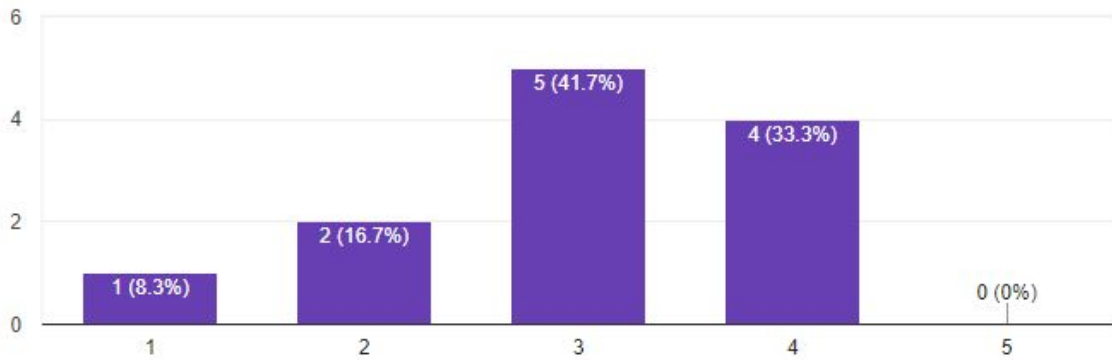
12 responses



I felt very confident using the system

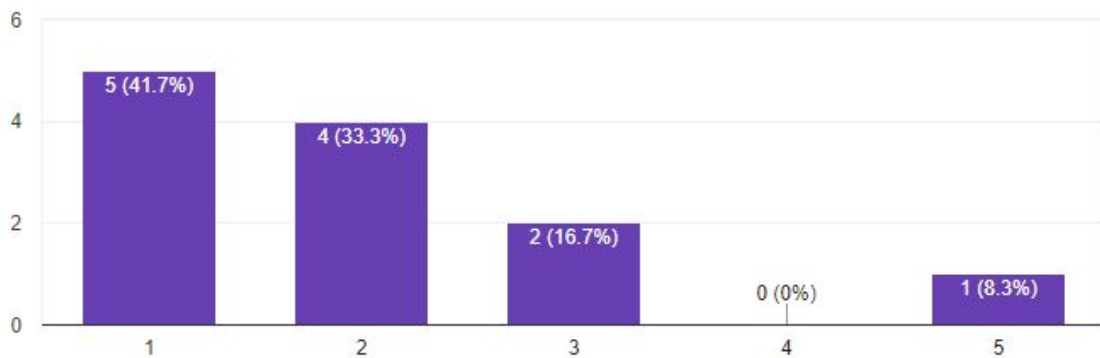


12 responses



I needed to learn a lot of things before I could get going with this system

12 responses

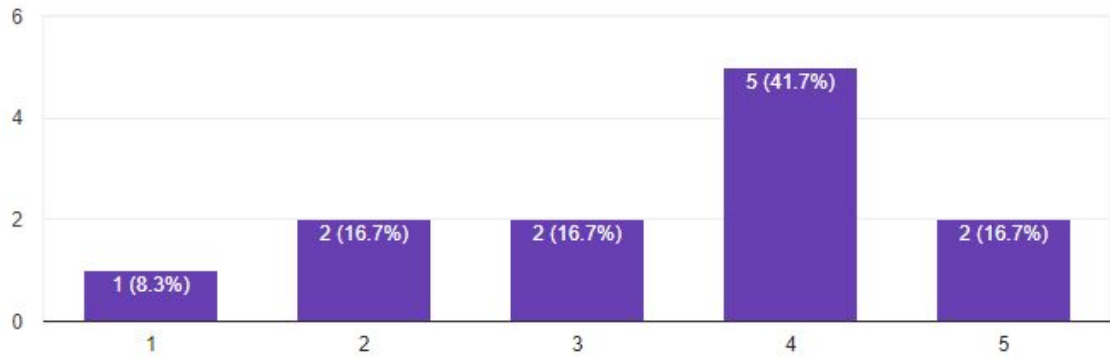


Levels

Do you believe that levels would make the task of verifying traceability links more satisfying?

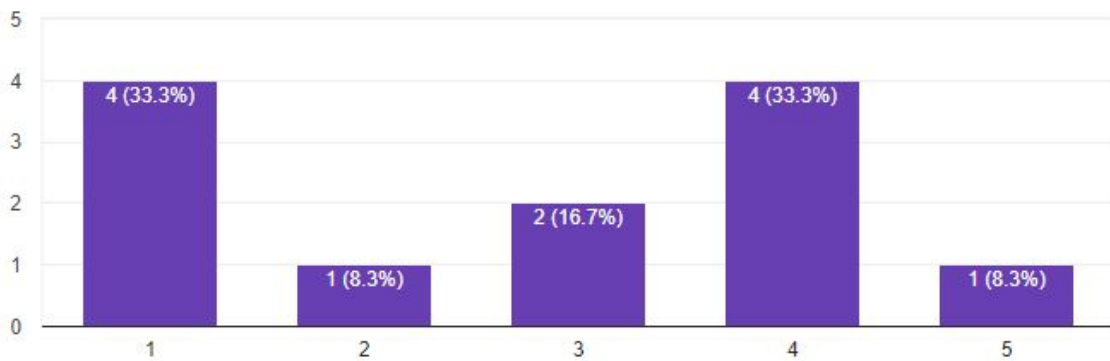


12 responses



Do you believe that levels would help in decreasing the time spent on each link?

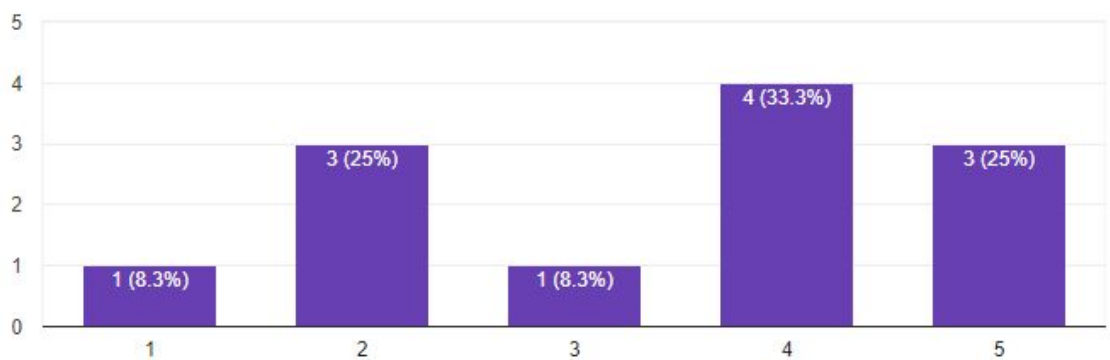
12 responses



Do you believe that levels would make you focus less on verifying the links correctly and focus more on verifying as many as possible?

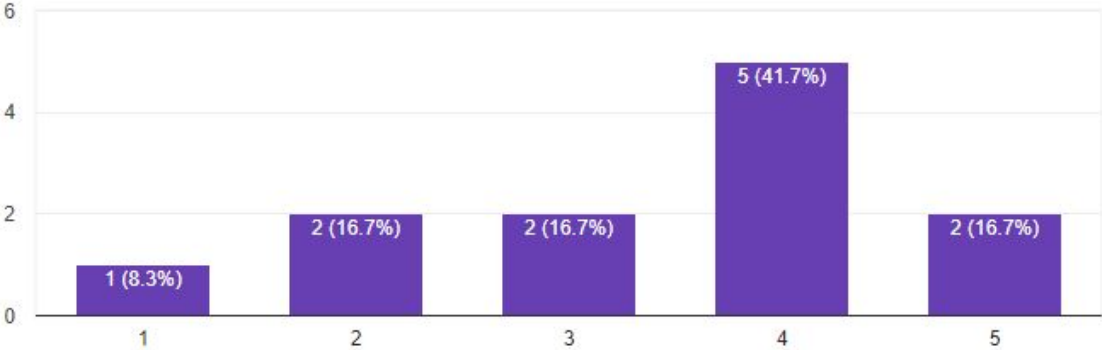


12 responses



Do you believe that levels would make you skip the more advanced links and go for easier ones in order to "level up" faster?

12 responses



If you want to clarify any of your answers given above, feel free to do so here.

5 responses

- I believe in general the levels will not be a great motivating factor, unless they are tied to a real world consequence (i.e. the performance review at your company evaluates you based on your level, and that has an impact on the achieved raise/bonus). My answers were with the assumption that the levels would have no real world impact.
- If levels are based on getting correct scores I believe they would be useful
- It is unclear what the benefit of the level is for me.
- no
- It is difficult to say whether I would attempt to complete more requirements or if I would attempt to get more traceability links correct when I'm unsure which would grant me more "points", so to say. I believe that if completing 'Harder links' rewarded more points, this would create more of an incentive to complete them whilst 'Easier links' awarded less points.

Do you think that a levels feature could have any negative effects? If yes, which ones?

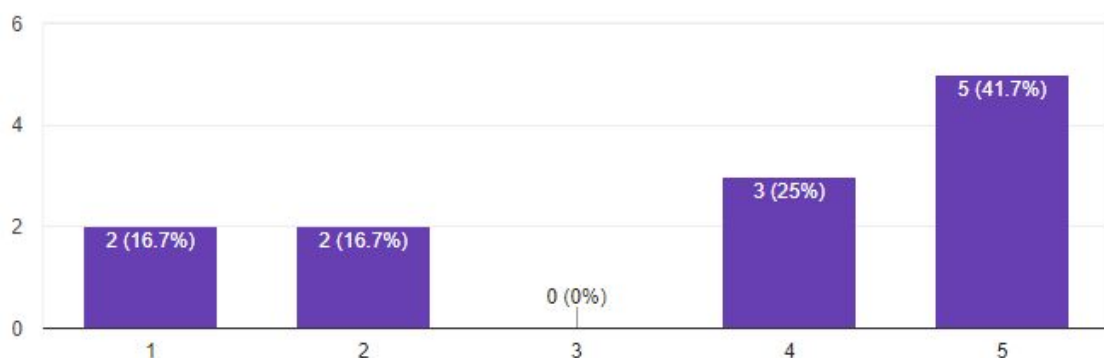
9 responses

not if its used as a motivation, yes if its used as a "ranking" within employees(
All depends on how you level up.
People might start caring about the levels more than the actual work.
If one would start doing the easiest verification first, critical ones may be left untouched. Unless they are weighted it would be hard to guarantee any kind of impact.
It would make completing links more important than verifying them
Not enough time spent on each link
Depends on the person i suppose, some might take it more seriously and could be positive, but personally for me i wouldn't so i would say negative for me.
no
The gamefication of traceability could perhaps lead to inaccurate descriptions of a persons effectivity and productivity if the system could be abused by gaining a lot of points for tasks which do not require much effort. If used as a key performance indicator by a more senior member of staff for evaluating the individuals using the system, If they are not aware of how the system works and scores its users, the 'Level' could be misinterpreted as an indicator of knowlegde of profficiency. It may also, like mentioned in one of the questions above, lead to the prioritization of less critical tasks in order to score more 'points' rather than actually tackling the most important and those with the highest priority.

Badges

Do you believe that badges would make the task of verifying traceability links more satisfying?

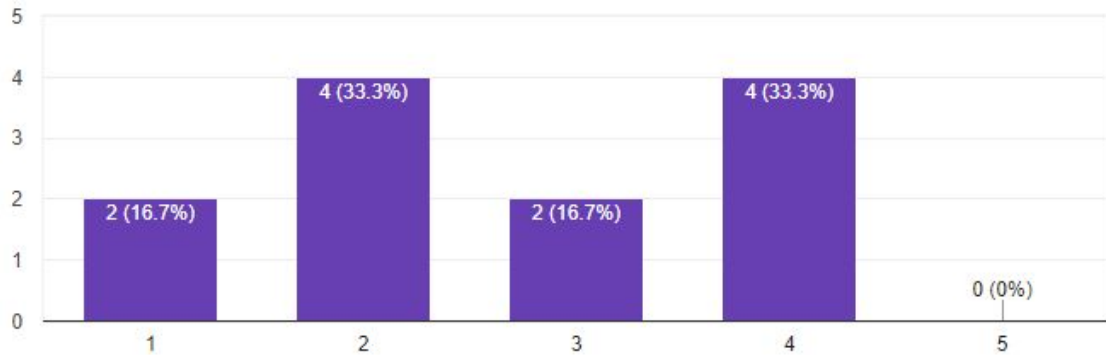
12 responses



Do you believe that badges would help in decreasing the time spent on each link?

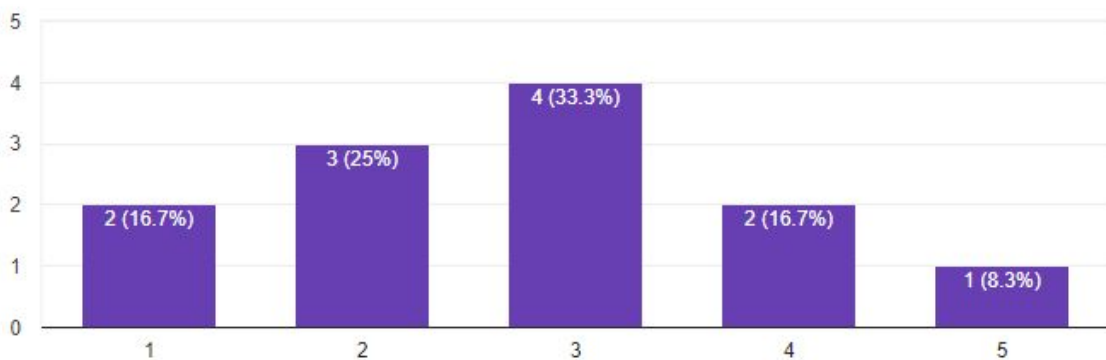


12 responses



Do you believe that badges would make you focus less on verifying the links correctly and focus more on verifying as many as possible?

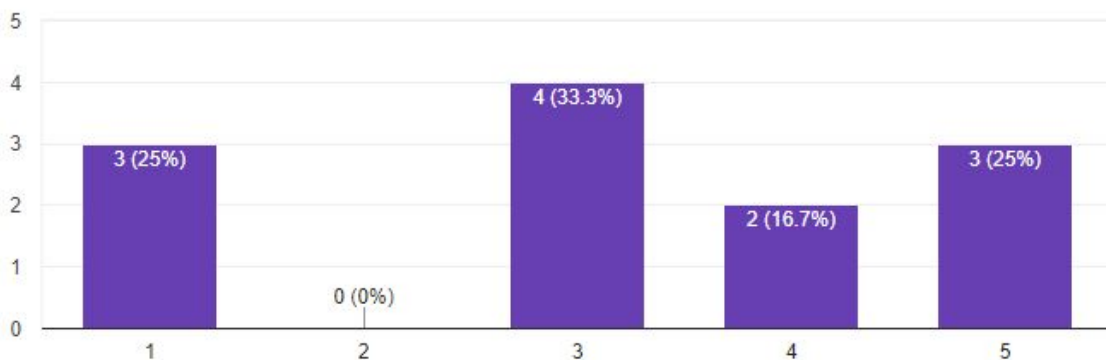
12 responses



Do you believe that badges would make you skip the more advanced links and go for easier ones in order to receive more badges?



12 responses



If you want to clarify any of your answers given above, feel free to do so here.

5 responses

Similar thoughts here as with the levels.

The badges track how diligently I am doing the task, so I would spend more time on each link

I wouldnt be more motivated to perform this kind of task by badges

no

I think that the badge idea is good. It motivates the user to perform their task more thoroughly and hence lead to a more in-depth analysis of the traceability task. Ensuring that the 'Opening of a source file' is not just done for the badge achievement and that the user actually looks through the file is something that could perhaps be handled by analysing how long the user looks at the code, if any attempt is made to scroll through it, rather than just opening it.

Do you think that a badge feature could have any negative effects? If yes, which ones?

7 responses

Less time spent on the more cumbersome links.

Similar thoughts here as with the levels.a

None really, it would be a good indicator of progress

I would feel monitored

Well some users (like me) probably wont care at all about them

I reason the same way as with levels.

no

Can you think of a feature that is not in the tool, but that you think would be useful when vetting traceability links?

9 responses

notification/list on when you skip links, easy to get back to them (if you can do so)

Difficulty levels presented on each link. E.g. links estimated to be more complex would get a higher difficulty level. Probably incorporate some machine learning algorithm which learns how long/complex the various links are.

Seeing relevant system documentation alongside the artifacts.

Give more points for harder ones and you'll get around the problem with people only focusing on the easier ones

Highlighting relevant code/assumptions/errors

Undo, Showing the vetted links

Seeing texts from excel directly in the tool

no

The long list of tasks is very mundane and can be intimidating. Something as simple as coloring each line slightly differently could increase a persons ability to sit and look at the tool for a longer period of time. Instead of removing the task when accepted/rejected, maybe highlight it in a green and red color and when the final task is completed, then remove the requirement. This would allow the user to reconsider their final answer before completing the requirement completely.

D

Appendix 4 - SUS statements

System Usability Scale

© Digital Equipment Corporation, 1986.

	Strongly disagree						Strongly agree
1. I think that I would like to use this system frequently	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	1	2	3	4	5		
2. I found the system unnecessarily complex	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	1	2	3	4	5		
3. I thought the system was easy to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	1	2	3	4	5		
4. I think that I would need the support of a technical person to be able to use this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	1	2	3	4	5		
5. I found the various functions in this system were well integrated	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	1	2	3	4	5		
6. I thought there was too much inconsistency in this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	1	2	3	4	5		
7. I would imagine that most people would learn to use this system very quickly	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	1	2	3	4	5		
8. I found the system very cumbersome to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	1	2	3	4	5		
9. I felt very confident using the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	1	2	3	4	5		
10. I needed to learn a lot of things before I could get going with this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	1	2	3	4	5		

E

Appendix 5 - Experiment Instructions

Vetting automatically generated traceability links.

Experiment instructions

The MedFleet system

In this experiment, you will interact with a system called MedFleet. MedFleet uses a fleet of drones to deliver medical kits to users that request assistance. The requests originate from a mobile application that uses GPS to identify the current location of the user that needs help. The incoming requests are then prioritized, scheduled, and assigned to one of the drones in the fleet. The drone is then dispatched to deliver the medical kit to the GPS coordinates. Example applications include:

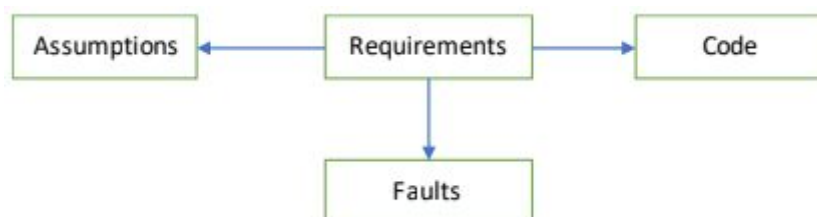
Natural Disaster - Victims of an earthquake can request assistance when emergency services in the area are fully utilized. State Park or Ski Mountain - Hikers, Campers, or Skiers that have a serious injury in a remote location can request medical assistance.

The workspace available to you contains the following artefacts:

1. Requirements of the system
2. Source code of the system
3. The assumptions that were made during the development of the system
4. A description of faults that could occur, their effects and criticality

Your Task

The different artefacts provided are related to each other (i.e. have traceability links between them). **In this project, traceability links are created so that they can be used for maintenance tasks.** Currently the links have been created using an information retrieval technique called VSM-IDF (Vector Space Model – Inverse Document frequency). These links are not 100% correct i.e., they contain both true links and false links. **Your task** is to analyse the links and accept the links you think are true and reject the links you think are false. The allowed relationships between the different artefact types is given in the figure below. The arrows indicate a source to target relationship.



The links are given to you in form of a list of sources (i.e. requirements) and targets (Java code, assumptions or faults). Please vet the links from top to bottom, do not skip any links in the process. If you cannot make a decision on if the target associated with a source is correct or not, you can leave that target in the list. If you have any questions, you are allowed to ask them during the task, but if you have any recommendations on how to make the tool better or the task easier, please keep these until the end of the Experiment.

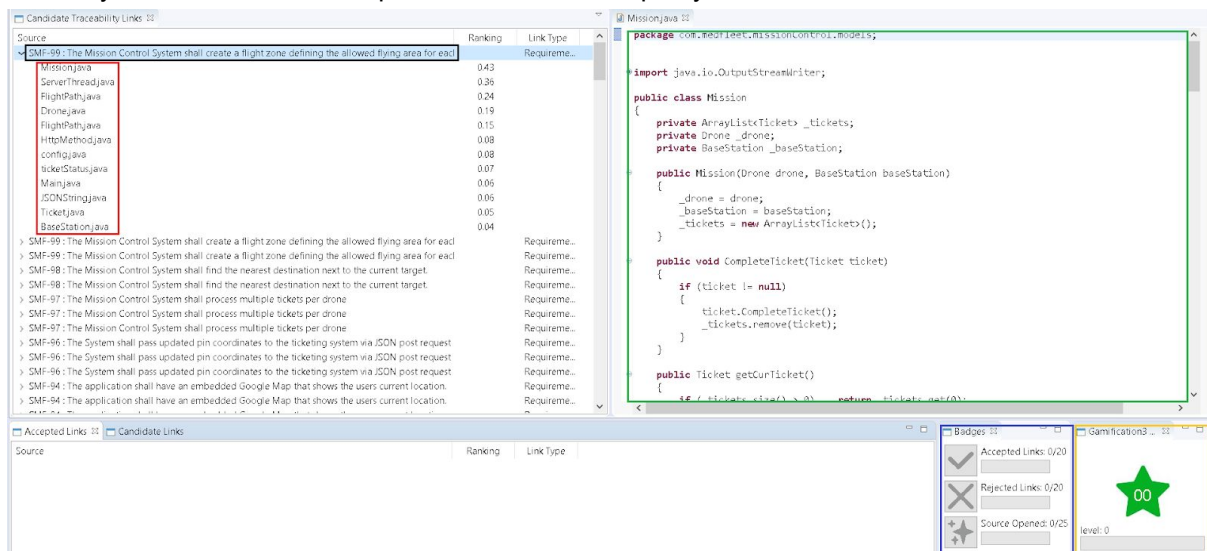
Aim of the experiment

The aim of this experiment is to compare the results of participants using the current version of Capra with the version we have extended with a few gamification features. We will collect the following measurements from the experiment:

1. Total time spent
2. The total number of links investigated
3. The correctness of the links accepted and rejected.

Using the tool

1. When the tool (Capra) is opened, it should look like the image below.
2. The candidate trace links are the links that you have to vet i.e. Accept or reject the link. The list of links shows you the sources (which are the requirements marked black), **you should expand the tree view** to see the different targets which can be code, assumptions or faults (marked red). You can accept or reject a link by selecting a target, right clicking and selecting accept or reject. If you are not sure about whether a link is correct or not, you can leave that link in the list and move on to the next one.
3. To get more information about the artifacts (i.e., requirements, assumptions or faults), right click on the artifact and click open (marked green).
4. Vet as many links as you can for a period of 45 minutes. The only restriction is that you should start from top to bottom, do not skip any links on the list.



Gamification features

Levels (marked yellow)

As you accept or reject any of the candidate links, you will get experience which in turn levels you up. Your current level and how much more experience that is required in order to level up, can be seen in the level window at the bottom right corner of the tool.

Badges (marked blue)

To the left of the level window you can see the badge window, which shows you 3 different kinds of badges which you can earn. One of the badges can be earned after you accept a certain amount of candidate links while one other badge can be earned after you reject a certain amount of candidate links. The last badge can be earned after you have opened and inspected a certain amount of target files (Java code, assumptions or faults).

Vetting automatically generated traceability links.

Experiment instructions

The MedFleet system

In this experiment, you will interact with a system called MedFleet. MedFleet uses a fleet of drones to deliver medical kits to users that request assistance. The requests originate from a mobile application that uses GPS to identify the current location of the user that needs help. The incoming requests are then prioritized, scheduled, and assigned to one of the drones in the fleet. The drone is then dispatched to deliver the medical kit to the GPS coordinates. Example applications include:

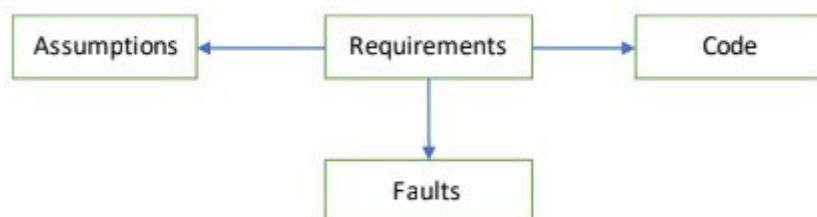
Natural Disaster - Victims of an earthquake can request assistance when emergency services in the area are fully utilized. State Park or Ski Mountain - Hikers, Campers, or Skiers that have a serious injury in a remote location can request medical assistance.

The workspace available to you contains the following artefacts:

1. Requirements of the system
2. Source code of the system
3. The assumptions that were made during the development of the system
4. A description of faults that could occur, their effects and criticality

Your Task

The different artefacts provided are related to each other (i.e. have traceability links between them). **In this project, traceability links are created so that they can be used for maintenance tasks.** Currently the links have been created using an information retrieval technique called VSM-IDF (Vector Space Model – Inverse Document frequency). These links are not 100% correct i.e., they contain both true links and false links. **Your task** is to analyse the links and accept the links you think are true and reject the links you think are false. The allowed relationships between the different artefact types is given in the figure below. The arrows indicate a source to target relationship.



The links are given to you in form of a list of sources (i.e. requirements) and targets (Java code, assumptions or faults). Please vet the links from top to bottom, do not skip any links in the process. If you cannot make a decision on if the target associated with a source is correct or not, you can leave that target in the list. If you have any questions, you are allowed to ask them during the task, but if you have any recommendations on how to make the tool better or the task easier, please keep these until the end of the Experiment.

Aim of the experiment

The aim of this experiment is to compare the results of participants using the current version of Capra with the version we have extended with a few gamification features. We will collect the following measurements from the experiment:

1. Total time spent
2. The total number of links investigated
3. The correctness of the links accepted and rejected.

Using the tool

1. When the tool (Capra) is opened, it should look like the image below.
2. The candidate trace links are the links that you have to vet i.e. Accept or reject the link. The list of links shows you the sources (which are the requirements marked black), **you should expand the tree view** to see the different targets which can be code, assumptions or faults (marked red). You can accept or reject a link by selecting a target, right clicking and selecting accept or reject. If you are not sure about whether a link is correct or not, you can leave that link in the list and move on to the next one.
3. To get more information about the artifacts (i.e., requirements, assumptions or faults), right click on the artifact and click open (marked green).
4. Vet as many links as you can for a period of 45 minutes. The only restriction is that you should start from top to bottom, do not skip any links on the list.

The screenshot displays the Capra tool interface. On the left, a table titled 'Candidate Traceability Links' lists various sources and their associated requirements. The 'Source' column includes files like Mission.java, ServerThread.java, FlightPath.java, Drone.java, FlightPath.java, HttpMethod.java, config.java, ticketStatus.java, Main.java, JSONString.java, Ticket.java, and BaseStation.java. The 'Ranking' column shows values ranging from 0.42 to 0.04. The 'Link Type' column is labeled 'Requirement...'. Below the table, a tree view shows expanded requirements, such as 'SMF-99: The Mission Control System shall create a flight zone defining the allowed flying area for each...'. On the right, a code editor shows the Java code for the 'Mission' class, including imports, class definition, and methods like 'CompleteTicket' and 'getFlightPaths'.

Source	Ranking	Link Type
Mission.java	0.42	Requirement...
ServerThread.java	0.38	Requirement...
FlightPath.java	0.24	Requirement...
Drone.java	0.19	Requirement...
FlightPath.java	0.15	Requirement...
HttpMethod.java	0.08	Requirement...
config.java	0.08	Requirement...
ticketStatus.java	0.07	Requirement...
Main.java	0.06	Requirement...
JSONString.java	0.06	Requirement...
Ticket.java	0.05	Requirement...
BaseStation.java	0.04	Requirement...

```
package com.medfleet.missionControl.models;

import java.io.OutputStreamWriter;

public class Mission
{
    private ArrayList<Ticket> _tickets;
    private Drone _drone;
    private BaseStation _baseStation;

    public Mission(Drone drone, BaseStation baseStation)
    {
        _drone = drone;
        _baseStation = baseStation;
        _tickets = new ArrayList<Ticket>();
    }

    public void CompleteTicket(Ticket ticket)
    {
        if (ticket != null)
        {
            ticket.CompleteTicket();
            _tickets.remove(ticket);
        }
    }

    public Ticket getCurTicket()
    {
        if (_tickets.size() > 0) return _tickets.get(0);
        return null;
    }

    public ArrayList<FlightPath> getFlightPaths()
    {
        ArrayList<FlightPath> _flightPaths = new ArrayList<FlightPath>();
        for (int i = 0; i < _tickets.size(); i++)
        {
            FlightPath fp = _tickets.get(i).getFlightPath();
            if (fp != null)
            {
                _flightPaths.add(fp);
            }
        }
        return _flightPaths;
    }
}
```