



Expertise  
and insight  
for the future

Tuomas Kuutti

# Comparing Requirements Management Tools – IBM Rational DOORS & HP ALM

Metropolia University of Applied Sciences  
Information and Communications Technologies  
Health Technology  
Bachelor's Thesis  
20.11.2019

Author Title Number of Pages Date	Tuomas Kuutti Comparing Requirements Management Tools – IBM Rational DOORS & HP ALM 37 pages + 2 appendices 20 November 2019
Degree	Bachelor of Engineering
Degree Programme	Information and Communications Technology
Professional Major	Health Technology
Instructors	Jaana Eloranta, Staff Software Engineer Erja Nikunen, Principal Lecturer
<p>This thesis was done for a company in the patient monitor business. Monitors are medical devices so their regulation is very strict. There needs to be evidence that all the requirements have been tested and that the tests have passed results. To make this easier and more robust, requirement management applications are used. The purpose of this thesis was to find out how the users feel about using IBM Rational DOORS, a requirement management application, and compare it to another application, HP ALM. Both are used in the company but on different projects.</p> <p>The theory section contains information about patient monitors, patient safety, Food and Drug Administration's role in regulation and their guidance to help fulfill the regulation. FDA is focused on as they regulate medical devices in the USA which is one of the largest markets in the world. Traceability is introduced, its role in software engineering and what are the benefits it produces.</p> <p>Requirement management applications DOORS, HP ALM and DOORS Next Generation are introduced.</p> <p>A survey was conducted with the DOORS users. The results (n=21) showed that the users are not overly happy with the application. The users who work with the application daily are more content with it. Also, the disliked and liked features of DOORS were collected in the survey. The most liked features were traceability and customizability. The most disliked features were the connection issues, slowness, and DOORS tables. The interviews (n=2) were done with people who were using HP ALM and had used DOORS. The results indicated that their preferred application would have been DOORS.</p> <p>The recommendations for improving the usability range from smaller fixes found from DOORS documentation, which could provide some boost for performance, to more extreme ones such as relocating the server to the local site. This would reduce the connection issues.</p>	
Keywords	traceability, requirements management, IBM Rational DOORS, HP ALM

Tekijä Otsikko	Tuomas Kuutti Comparing Requirements Management Tools – IBM Rational DOORS & HP ALM
Sivumäärä Aika	37 sivua + 2 liitettä 20.11.2019
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintäteknikka
Ammatillinen pääaine	Hyvinvointi- ja terveysteknologia
Ohjaajat	Jaana Eloranta, Staff Software Engineer Erja Nikunen, Yliopettaja, Metropolia AMK
<p>Tämä opinnäytetyö tehtiin yritykselle, joka tekee potilasmonitoreita. Monitorit ovat lääkinnällisiä laitteita, joten niihin liittyvät rajoitukset ovat erittäin tiukkoja. Valmistajalla pitää olla todisteet siitä, että kaikille vaatimuksille on testit ja ne on läpäisty. Tätä työtä voidaan helpottaa ja samalla tehdä siitä järjestelmällisempää vaatimustenhallintaohjelmilla. Tämän työn tavoitteena oli ottaa selvää mitä mieltä käyttäjät ovat IBM Rational DOORS -vaatimustenhallintaohjelmasta ja verrata sitä toiseen ohjelmaan nimeltä HP ALM. Molemmat ovat käytössä yrityksessä, mutta eri projekteissa.</p> <p>Teoriaosio sisältää tietoa potilasmonitoreista, potilasturvallisuudesta, FDA:n (Food and Drug Administration) roolista säännöstelyssä sekä heidän ohjeistuksistaan sen täyttämiseksi. FDA:han keskitytään, koska he pitävät huolta lääkinnällisten laitteiden lainsäädännöstä Yhdysvalloissa, joka on suurimpia yksittäisiä markkinoita. Myös jäljitettävyydestä puhutaan, sen roolista ohjelmistotuotannossa ja mitä hyötyjä se tuo.</p> <p>Työssä esiteltävät vaatimustenhallintaohjelmat ovat DOORS, HP ALM ja DOORS Next Generation.</p> <p>DOORS käyttäjille tehtiin kysely, jonka tulokset (n=21) antoivat ymmärtää, etteivät käyttäjät ole täysin tyytyväisiä ohjelmaan. Päivittäin ohjelmaa käyttävät ovat tyytyväisempiä ohjelmaan kuin muut. Kyselyssä kerättiin myös ohjelman pidetyimmät ja vähiten pidetyt ominaisuudet. Pidetyimpiä ominaisuuksia olivat jäljitettävyys ja muokattavuus. Vähiten pidetyt ominaisuudet olivat yhteysongelmat, hitaus ja taulukot. Kahta sellaista henkilöä haastateltiin, jotka ovat käyttäneet sekä DOORS:ia että HP ALM:ia. Näiden haastatteluiden mukaan miellettävämpi ohjelmisto olisi ollut DOORS.</p> <p>Suosituksia käytettävyyden parantamiseen koostuvat pienistä sekä suurista vaihtoehdoista. Pienimmät ja samalla helpoimmat parannukset löytyvät DOORS:n dokumentaatiosta, joilla saatettaisiin saada jonkinlaisia parannuksia tehokkuuteen. Parannukset, jotka vaativat enemmän resursseja kuten servereiden siirtäminen Suomessa sijaitsevalle työpisteelle vähentäisivät varmasti yhteysongelmia.</p>	
Avainsanat	jäljitettävyys, vaatimustenhallinta, IBM Rational DOORS, HP ALM

## Contents

### List of Abbreviations

1	Introduction	1
2	Background	2
2.1	Patient Monitors	2
2.2	Patient Safety	3
2.2.1	Therac-25	4
2.2.2	Device Safety	4
2.2.3	International Medical Device Regulators Forum	5
2.3	FDA	5
2.4	Traceability in Software Engineering	7
2.4.1	Good Traceability Practices	7
2.4.2	Benefits of Traceability	8
2.4.3	Traceability with Medical Devices	9
3	Usability	9
4	Requirement Management Tools	10
4.1	IBM Rational DOORS	11
4.2	HP ALM	13
4.3	IBM Engineering Lifecycle Management: DOORS Next Generation	16
4.4	Comparison	19
5	Research Methods	19
5.1	Survey	20
5.2	Interviews	21
6	Results and Analysis	21
6.1	Survey	22
6.2	Open-ended Questions	25
6.3	Interviews	28
6.4	Recommendations	29
7	Conclusion	33
	References	34

Appendices

Appendix 1. Survey

Appendix 2. Interview questions

## List of Abbreviations

ALM	Application Lifecycle Management
DOORS	Dynamic Object-Oriented Requirements System
DXL	DOORS Extension Language
FDA	Food and Drug Administration
HP	Hewlett-Packard
IBM	International Business Machines Corporation
ICU	Intensive Care Unit
ISO	International Organization for Standardization
NICU	Neonatal Intensive Care Unit
OR	Operating Room

## 1 Introduction

Requirement management is an integral part of any software development project. That is why there are tools which are meant to keep the requirements in check. In this study two requirement management tools: IBM Rational DOORS and HP ALM were compared. DOORS is used for requirements management and test execution in a patient monitor project at the moment and HP ALM is an interesting alternative for it. HP ALM is used on another, newer project within the same company. The third option looked into is DOORS Next Generation.

The theory section contains information about patient monitors and patient safety, because the commissioner of the study is in the patient monitor business. Patient safety is discussed as it is an important part of medical device development and traceability is a big part of it.

The requirement management tool currently in use is IBM Rational Dynamic Object-Oriented Requirements System, better known as DOORS. The considered replacement is HP ALM (Application Lifecycle Management) which is more of a set of tools used in software development. DOORS is used mainly for requirement management, writing and running manual tests. HP ALM offers other tools as well, for example defect management which is currently done on a different program.

Traceability is always important, but when working with medical devices it is part of legislation. Therefore, the used tool must be able to provide accurate and correct traceability. In this thesis traceability requirements are limited to FDA (Food and Drug Administration) legislation and guidance as USA has the largest and therefore the most important market.

The usability of both applications is studied through a survey, interviews and available literature. The idea of the survey was to find out the problematic features with DOORS, the requirements management tool currently in use, and then utilize these in the HP ALM related interviews. Also the good qualities of the application were of interest, and they were asked about in the survey.

The research questions are: Is there a difference in usability between the two software and are there ways to improve DOORS usability.

The conclusion was drawn using the gathered literature, survey and interview results. This should reveal the difference in functionalities of the programs and how they affect the way of working. The results of the interviews should provide an answer to whether the drawbacks of DOORS are present in HP ALM and are there new problem areas in HP ALM.

## 2 Background

This chapter first discusses patient monitors as well as patient safety and device safety. Then, International Medical Device Regulators Forum (IMDRF) and The Food and Drug Administration (FDA) are introduced. Finally, the chapter concludes with traceability's role in software engineering, good traceability practices and how its needed with medical devices.

### 2.1 Patient Monitors

Patient monitors are designed to measure patients' vital signs and organ functions. They also detect and alarm in case of abnormal or life-threatening conditions. The physiological parameters are monitored continuously. These measurements are shown as waveforms or numerical data. They can also be stored to observe the trends of the measured parameters. (Kramme, Hoffmann, & Pozos, 2011, p. 947)

Patient monitoring can be done for a shorter period, during surgery, or for a longer period, in intensive care unit. In both cases the goal is to alert the medical staff of unstable conditions as early as possible. (Kramme, Hoffmann, & Pozos, 2011, p. 947)

Different monitor types are used in situations where they fit. Transport monitor is a lighter model with a battery, so it can be used during transport. Newer models can be used as acquisition devices when connected to another monitor, but when they are removed from the host monitor they work as standalone monitors (GE Healthcare, 2018). Telemetry



devices are portable transmitters which the patient can carry themselves. These monitors do not usually have displays so the data they collect is observed with a different device for example a central station. (Kramme, Hoffmann, & Pozos, 2011, p. 949)

The monitors are scalable therefore measured parameters can be chosen according to the need and situation. For example, when conducting basic monitoring, invasive parameters are most likely not needed. If there is a need for invasive parameters or some other parameter, another module can be connected to the monitor to get this parameter. (Kramme, Hoffmann, & Pozos, 2011, p. 948)

The monitors can be set for specific care areas, so their default settings are more suited for patients in that care area. For example, GE's B650 multi-parameter monitor can be set with NICU (Neonatal intensive care unit) software package which uses neonatal algorithms. These algorithms work better with neonatal patients than the algorithms on the other software packages. (GE Healthcare, n.d.)

## 2.2 Patient Safety

Patient safety means that the patient gets the treatment they need with the least amount of harm possible. Safe treatment, medical and medical device safety are all parts of patient safety. (Terveyden ja hyvinvoinnin laitos, 2019)

A way to increase patient safety is monitoring the patient. Checking the patient personally is still one of the most important ways of making sure that everything is all right. Unfortunately, this one to one kind of monitoring is possible only in some situations like OR and ICU. When one person needs to tend to many patients the gap between the checks grows too big and this can lead to missing a critical situation. This is where patient monitors can help. They spot anomalies which people could miss and give out an alarm. The monitor also shows more detailed information about the patient than what can be get by observing the patient. (Salmenperä, 2014)

A medical device with a poorly done software can have devastating effects on the patients. An example of this is from the late 1980s when Therac-25 linear accelerator gave

patients massive X-ray overdoses. Everything that could have gone wrong with the software did go wrong. This case caused the FDA to tighten their legislation. (Vogel, 2011, p. 15)

### 2.2.1 Therac-25

With Therac-25 the software itself had problems such as coding errors, but this was not the only reason for the catastrophic failures. There was very little if any documentation about software specifications and testing plan. Gathered comments give a picture that unit and software testing was minimal which is concerning as the Therac-25's software was responsible about the security of the device. The previous versions, Therac-6 and Therac-20, were originally standalone machines which had safety features build in the hardware and the computer control was added later. Some parts of the software from the previous models were used in Therac-25 which could be the reason for some of the problems in the software. (Levensen & Turner, 1993, p. 20)

There was overconfidence towards the software which can be seen from the fact that the software was barely mentioned in the risk assessment and the software related risks considered were hardware related. In addition to this the hardware interlocks were removed as they were seen as unnecessary. (Levensen & Turner, 1993, pp. 21, 38)

### 2.2.2 Device Safety

The use of a medical device contains a degree of risk. In certain circumstances all devices may cause problems. Many of these problems cannot be detected until the device has been on the market for a while. The product might fail in a way that could not have been predicted when developing the device. To make the device safety better the potential hazards are estimated. This approach is known as risk assessment. (Cheng, 2003, p. 3)

The risks are measured as a combination of the hazard, the likelihood of it happening and the severity or overall impact of it. First the possible hazards are identified after which the risk of the hazard is evaluated. This risk assessment is based on experience, evidence and at some parts guesswork. (Cheng, 2003, p. 3)

Standard ISO 14971 has been produced to provide a framework risk management in medical device design. The framework includes risk analysis, risk evaluation and risk control. ISO 14971 is recognized as an acceptable risk management model by the FDA while European Union sees it as mandatory. (Cheng, 2003, p. 4)

### 2.2.3 International Medical Device Regulators Forum

International Medical Device Regulators Forum (IMDRF) is a group of device regulators who aim to harmonize the medical device regulation. The group consists of volunteers from around the world who continue the work started by the Global Harmonization Task Force (GHTF). Currently there are 10 members which include the United States of America, Europe and China. (International Medical Device Regulators Forum, n.d.)

Harmonization would lower the regulatory barriers, facilitate trade and improve access to new technologies (Cheng, 2003, p. 15). Currently there are significant differences in the legislation between countries, which means that getting a product to market in one country does not mean it is anywhere close to getting to the market in another.

## 2.3 FDA

FDA is a federal agency of the United States Department of Health and Human Services. They have a broad regulatory authority which includes multiple areas of which two are in the name, foods and drugs. One of the others under their regulation is medical devices. They make sure that the items under their surveillance are safe, effective and of certain quality. The premarket approval of medical devices is done by an FDA branch The Center for Devices and Radiological Health (CDRH). They also care for the safety and performance of the devices. (US Food and Drug Administration, 2018a) (US Food and Drug Administration, 2017)

Every region has their own regulations and the FDA is no different. The regulations are mandatory as they are there to ensure the safety and efficacy of medical devices. The regulations are based on The Federal Food, Drug, and Cosmetic Act or other federal laws. Additionally, the FDA has guidance documents which explain the requirements for

the FDA regulations. These documents are voluntary to use, but they do describe the agency's current thinking on the regulatory issues. (Regan, Caffery, Daid, & Flood, 2013, pp. 3-4) (US Food and Drug Administration, 2018b)

FDA offers guidance documents which are designed to help with gaining the approval for any medical device. They show one way to do this, but an alternative way can be used if it satisfies the requirements. One of these documents is "Guidance for the Content of Premarket Submissions for Software contained in Medical Devices" which provides information about the documentation that is recommended to be in premarket submissions for medical devices which have anything to do with software. (US Food and Drug Administration, 2005, pp. 1-2)

The recommended documentation depends on the Level of Concern of the device. The level can be Minor, Moderate or Major. This depends on the estimate of the severity of injury that the device could cause to the patient or operator because of device failure, design flaw or by its intended use. If the failure could lead to a serious injury or death the Level of Concern is Major. If there is a chance of minor injury, it is considered Moderate and if the chances of causing any injury are unlikely the Level of Concern is Minor. (US Food and Drug Administration, 2005, pp. 4-5, 9-10)

As mentioned above, the recommendations are given according to the Level of Concern. For example, Verification and Validation Documentation has different recommendations for each concern level, but Traceability Analysis has one common recommendation for all. (US Food and Drug Administration, 2005, pp. 9-10)

Traceability Analysis links together design requirements, design specifications and testing requirements. It also helps putting together the identified hazards with the implementation and testing of the mitigations. Traceability of the aforementioned activities is essential to product development and helps the reviewer to understand product design, development, testing and hazard mitigations. (US Food and Drug Administration, 2005, p. 13)

## 2.4 Traceability in Software Engineering

Traceability means that there is linking between system requirements and other system elements. These elements include other requirements, test cases and documentation among other things. The links allow navigation to both directions. From a specific requirement to the test results for it or the other way around. (Wieggers, 2013)

Traceability is a critical part of rigorous software development and it is required for the approval and certification process of most safety and security critical systems. These systems have a more and more important role in everyday lives. Although an important part of software development, traceability can be hard to achieve. The cost, effort and discipline required to make and maintain trace links can be very high. Therefore, getting a good grip of it can be difficult. (Mäder, Olivetto, & Marcus, Empirical studies in software and systems traceability, 2017)

Another problem with traceability is that the benefits can often go unrealized due to badly defined and ad-hoc traceability processes. Other reasons could be poor user training or missing tool support. This is a shame as traceability has been shown to reduce development effort and to increase development quality. (Mäder, Olivetto, & Marcus, Empirical studies in software and systems traceability, 2017)

### 2.4.1 Good Traceability Practices

Traceability can be a heavy, error-prone and costly process if it is not planned and implemented wisely. If the traces are done as an afterthought using brute force, the links will end up being less helpful and very difficult to maintain. (Mäder, Jones, Zhang, & Cleland-Huang, 2013, p. 58)

Planning traceability strategically when starting the project will provide instructions on how to make links in the project. One way is to use traceability information model (TIM) to plan which artifact types (requirements, testing, etc.) are linked. This model can then be used to make sure that the links are done between the artifacts their meant to. (Mäder, Jones, Zhang, & Cleland-Huang, 2013, pp. 59, 60)

The requirements need to have unique IDs to make sure there is no chance of mixing the requirements. The linking should use the IDs as reference points and not for example a certain row in a table. If a requirement points to a row in a table and there is a change with the table, a row is added or removed, the linking will be wrong. The IDs should also have prefixes that describe the artifact or at least are different between artifact types. (Vogel, 2011, p. 187) (Mäder, Jones, Zhang, & Cleland-Huang, 2013, pp. 62, 63)

It is also useful to do the linking while working on the requirement or test case etc. If the link traces are done at the end of the project, it can be hard to remember everything that should be linked. Especially as the people doing the trace linking at this point might not be the same developers, testers and requirement engineers. When the tracing is done during development, it can be utilized in the development. (Mäder, Jones, Zhang, & Cleland-Huang, 2013, p. 64)

A way to reduce the time used for maintaining, creating and using trace links is to use a program that will support them. One of these programs is Rational DOORS which allows creating, maintaining and using the links. These programs can usually show the trace information in different formats. (Mäder, Jones, Zhang, & Cleland-Huang, 2013, p. 60)

#### 2.4.2 Benefits of Traceability

As mentioned above, traceability can increase the quality of the software and reduce the development effort. It can also prove compliance with contracts and specifications. Couple of other situations where traceability can provide help are maintenance and testing. Maintenance is easier when the changed part, were it a requirement or part of code, can be traced to where the change would and could affect. (Wiegers, 2013)

A bug found during testing can be hard to pinpoint to the correct place in the code. If there are links between tests, requirements and code, finding the correct piece of code can be easier. This is especially true if there has been some change in personnel and the knowledge base is not quite as high yet as it were with the previous person. (Wiegers, 2013)

Many of the benefits of traceability are long-term. The initial development cost is increased due to managing the traceability information, but when there are modifications, additions or removals to the software the prior work with traceability will pay off. (Wieggers, 2013)

### 2.4.3 Traceability with Medical Devices

FDA guidance for premarket submission says about traceability: “Provide traceability to link together design, implementation, testing, and risk management” and “A Traceability Analysis links together your product design requirements, design specifications, and testing requirements.” (US Food and Drug Administration, 2005, pp. 8, 13)

Medical devices are part of a safety-critical industry, so traceability is used to prove that the development process has been rigorous and that there is evidence to prove the safety of the developed system (Rempel & Mäder, 2016). Traceability has a role in providing evidence that device specifications and implementations are connected to identified hazards. These hazards are used to make a risk analysis which produces the system-level requirements. Those requirements relate to for example test procedures and their results. (Mäder, Jones, Zhang, & Cleland-Huang, 2013, p. 58)

## 3 Usability

A usable product is usually something that can be identified with a lack of frustration when using the product. This is achieved when the user can do what they want the way they want, without hindrance, hesitation or questions. (Rubin, Chisnell, & Spool, 2008, p. 4)

Jakob Nielsen divides usability to 5 components making it less abstract:

- Learnability: When using the product for the first time, how easy is it to do basic tasks?
- Efficiency: When the product has been familiarized with, how quickly can the tasks be done?
- Memorability: When the product has not been in use for a while, how easy is it to start using it proficiently?

- Errors: When using the product, how many errors are done, how severe are these errors and how easily can they be recovered from?
- Satisfaction: How pleasant is it to use the product? (Nielsen, Usability 101: Introduction to Usability, 2012)

Good usability may also provide cost savings. As an example, a rotary dial phone company improved their product's usability, which sped up their users' dialing behavior by about 0.15 seconds. This led to a total annual savings of around \$1,000,000. (Nielsen, Usability Engineering, 1993, pp. 2,3)

Usability is not the only important attribute; another major attribute is utility. Utility means that the application provides features that the user actually needs. Usability and utility together will tell whether something is useful. There's no need for an application which is easy and nice to use, if it does not provide the correct features. And vice versa, if it offers the necessary tools, but using it is next to impossible it might not be worth the hassle. (Nielsen, Usability 101: Introduction to Usability, 2012)

If a user must use a tool which does not have a good usability, utility or either, the user is more likely to not want to use that tool. This could manifest as avoiding tasks that rely on using the tool. In more extreme cases having to use tools with bad usability may play a part when the user decides whether to change to another job.

#### **4 Requirement Management Tools**

Requirements management is an integral part of any development process and, different tools have been made to help. The tools should be able to manage changes to requirements, have version control and trace the requirements amongst other things. (Pandey & Pandey, 2012)

The tools mentioned here were chosen because IBM Rational DOORS is the requirements management tool in use at the moment, HP ALM is used on another project within



the same company and DOORS Next Generation is a requirements management application which has taken a lot of inspiration from DOORS and is part of IBM's Engineering Lifecycle Management toolset.

#### 4.1 IBM Rational DOORS

DOORS was published in 1991 and originally developed by Quality Systems and Software Ltd (QSS). In 2000 Telelogic bought QSS and 2008 IBM acquired Telelogic and named the application as IBM Rational DOORS. In April 2019 IBM announced that they would rename their portfolio and the products in it. The portfolio was known as Continuous Engineering or Rational, the new name is IBM Engineering Lifecycle Management. Rational DOORS was renamed to IBM Engineering Requirements Management DOORS Family, but the acronym stayed as DOORS.

DOORS is one of the most used requirements management tools on the market. It helps with different aspects of requirements management for example, links between requirements, test procedures and test results. These links are needed when checking whether all the requirements and tests that were supposed to be tested have a pass result.

DOORS is a client-server application and the company has decided to put the client on a Citrix server. This means that the user does not have the client installed on their computer, but they connect to a Citrix server and start the DOORS client from there. Reason for this is most likely that this way the latency between client and server would be lower than having the client on the user's own computer and connecting to a server in another country. Other reason could be that this way everyone will be using the same client version, even if connecting from another site. (IBM, a, n.d.)

DOORS' ability to create modules is utilized so that requirements and test procedures are divided into smaller modules instead of having one massive module which would then hold everything. The main view shows all the modules, choosing one will open it in a new window. Figure 1 shows what a DOORS module can look like. Module explorer can be used to quickly navigate to the wanted requirement or section. The main view shows the requirement, links and wanted attributes. Different products and software versions can be given as attributes and then filter out the unwanted steps. A module could

contain requirements from multiple projects especially if the projects are different versions of the same software and therefore share most of the requirements.

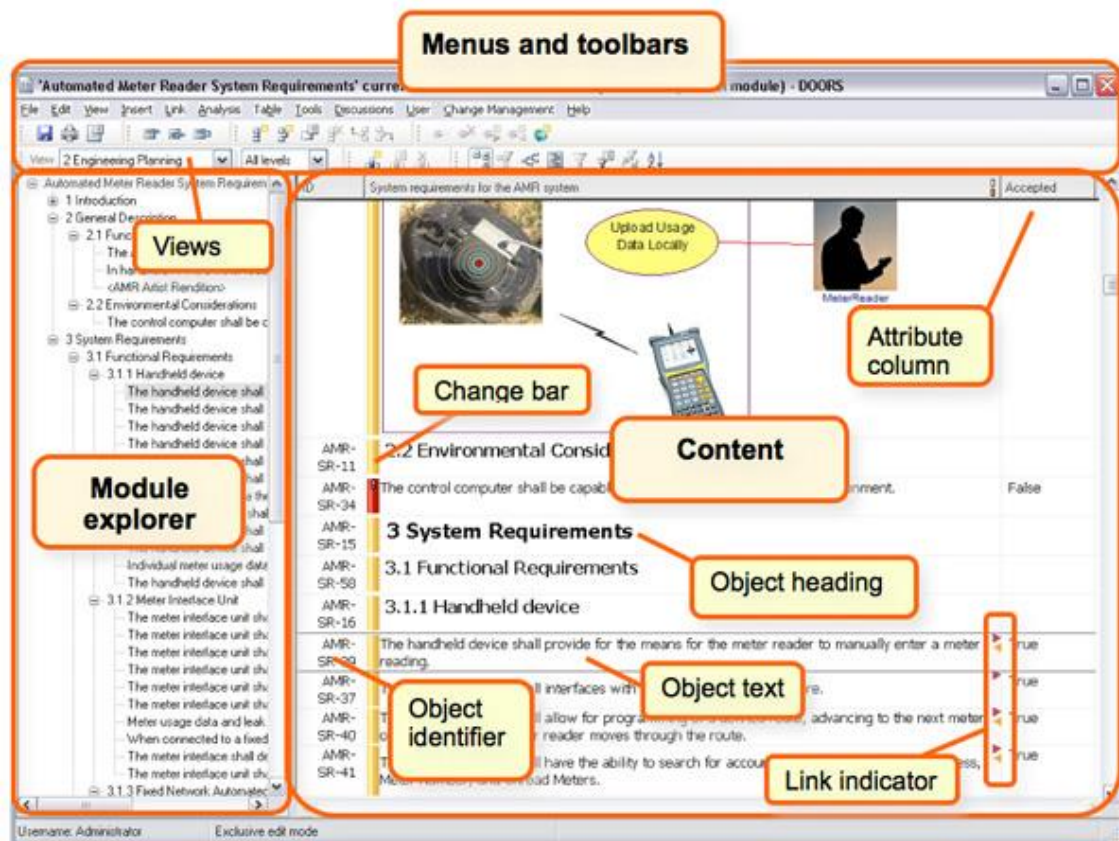


Figure 1. Example of a DOORS module (Papadakis-Kantos, IBM Rational DOORS getting started, 2013a).

DOORS also keeps track of the changes and saves the history. In Figure 1 steps which have a yellow change bar have been modified or added since the previous baseline. Red means that the change has not been saved yet. If a step has been modified the history tab will show who made the change, when it was done and what was done. It shows what the step contained before and after the change and which attributes were modified. Link changes are not shown in change history.

There are some known issues which may cause decreased performance such as connection between client and server. The data in DOORS is sent in small packages so good ping times are important for optimal performance. The latency should be under 50 milliseconds for normal performance. If it is more than 50 milliseconds, the performance

could decrease greatly. Pinging from a company laptop shows the latency to the Citrix server to be on average over 120 milliseconds. According to Citrix, latency under 150 milliseconds should provide great user experience. (IBM, 2013) (Contreras, 2017)

DOORS has 2 edit modes: exclusive and shareable. With exclusive edit the user locks the whole module so that they are the only one to edit the module. DOORS offers a possibility to create separate sections so that multiple users can work with same module simultaneously using shareable edit mode. These sections are controlled in different files in the database and must be loaded when the module is opened. Having too many of these sections will lower the performance of the program. Performance can also be decreased by large tables, large number of tables, OLE (Object Linking & Embedding) objects and pictures. (IBM, b, n.d.) (IBM, 2013)

## 4.2 HP ALM

The application has gone through a few name changes. Originally it was developed by Mercury Interactive and named as TestDirector, in version 8 it was renamed to Quality Center. Mercury Interactive was acquired by HP which rebranded the application as HP Quality Center. When version 11 was released the application was renamed to HP ALM. Now the application is developed and marketed by Micro Focus, but still uses the HP ALM name. (Guru99.com, n.d.)

HP ALM is a set of tools made for application development and testing. The tools include requirement management, testing and defect management. The idea is that instead of having most of the development tools on different platforms, they are brought to one. ALM can be used via a web browser. By default, the browser has to be Internet Explorer version 8-11, but plugins can be installed to the server which enable using other browsers as well.

There are four ALM editions which provide slightly different functionality. Table 1 below shows the features each edition contains.

Table 1. HP ALM editions (Guru99.com, n.d.).

Feature	HP ALM	HP ALM Essentials	HP QC Enterprise Edition	HP ALM Performance Center Edition
Release Management	✓	✓	✓	✓
Project Planning and Tracking	✓	✗	✗	✗
Risk Based Quality Management	✓	✗	✓	✗
Test Authoring and Execution	✓	✓	✓	✓
Test Resources	✓	✓	✓	✓
Test Configurations	✓	✗	✓	✓
Versioning	✓	✗	✓	✓
Base lining	✓	✗	✓	✓
Cross Project Customization	✓	✓	✗	✓
Sharing Defects	✓	✗	✗	✗
Hp Sprinter	✓	✗	✓	✗
Lab Management	✓	✗	✗	✓

Requirements are linked to tests creating a trace from the requirement, which is utilized when needing to know whether there is a test for each requirement. This will also tell if the tests for a requirement have been run and if they passed or failed.

Defects can be linked to other entities, like tests and requirements. This linking could be used when making tests for a defect or when a defect is connected to a certain requirement. (HP, 2010, p. 127)

There is an alarm feature in HP ALM which can be used to automatically inform when a change has been made. If a requirement was modified the tests related to this requirement would be flagged and the test designers would be notified. Clicking the alarm flag will open a dialog box which tells which requirement and change triggered the alert. (HP, 2010, pp. 135-139)

In HP ALM the requirements are in a folder structure as shown in Figure 2. A couple of levels down from root there are for example “User Requirements Specification” folder which contains the requirements for that document. Headings and other information are also held in the folder so that everything can be easily exported as a full document.

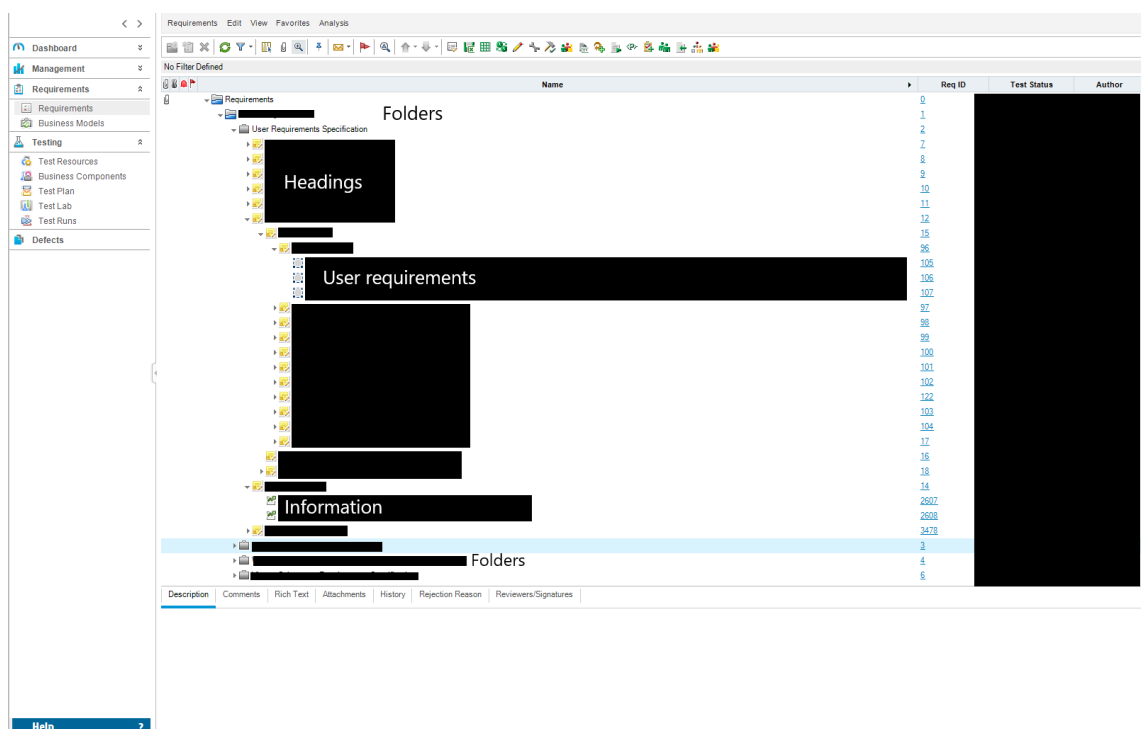


Figure 2. HP ALM's requirements folder structure.

Requirements and other objects can be given different attributes which can be used for filtering. In Figure 2 the two information steps have different version as an attribute so depending on the version exported, only one of these steps will be used in the exported document. Filtering can also be utilized on the main view for example to only show objects that belong to version 1.0. The attributes of the requirement can be seen when opening it in a detailed view. The user can also configure which attributes are shown as

columns on the main view. In Figure 2 the columns which are shown are Name, ReqID (Requirement ID), Test Status and Author. Every entity has an ID and it is unique. They are used for the traceability.

Functionality can be added to HP ALM via scripts. The language used is VBScript (Visual Basic Scripting Edition) and it is based on Microsoft's programming language Visual Basic. With the browsers it has built-in support only with Internet Explorer.

Migrating requirements from DOORS to HP ALM can be done. One way would be to export the wanted module as an excel and importing it to ALM. An excel add-in needs to be added to ALM. There are also some 3<sup>rd</sup> party applications such as agosense.symphony which helps with the integration of the two but can be also utilized for migration of requirements.

#### 4.3 IBM Engineering Lifecycle Management: DOORS Next Generation

A compelling alternative for DOORS could be DOORS Next Generation (DNG). DNG can be installed as part of IBM's Engineering Lifecycle Management, which is their set of application lifecycle management tools. DNG runs on IBM Rational Jazz platform and it includes a server application and a web client. (IBM, c, n.d.)

What makes DNG an attractive option is the fact that it is similar to DOORS and the migration can be done using ReqIF (Requirements Interchange Format). Although, the migration requires some modifications to the modules. Attributes in different modules need to be unified, meaning that attributes which do the same thing in different modules need to have the same name and options as possible values similarly to those shown in Figure 3. It is important as with DNG the attributes are defined on a project level, instead of in each module separately like with DOORS. During migration a new project level attribute is created for each different attribute.

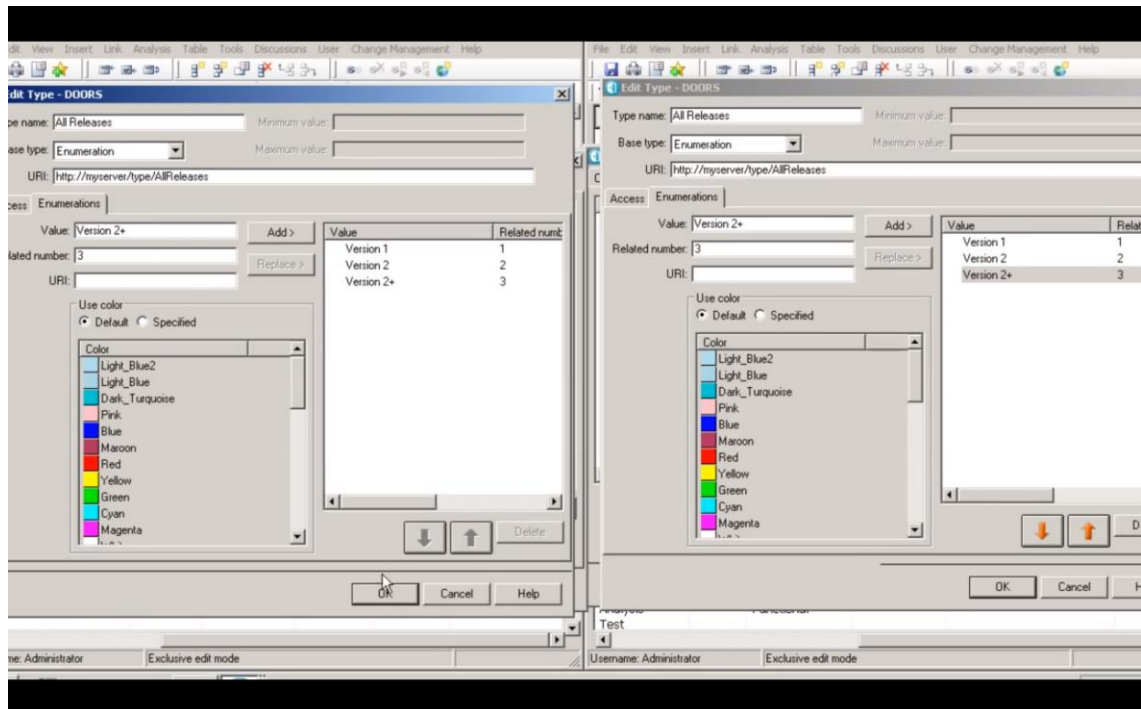


Figure 3. Unifying the attributes of two DOORS modules (Screen capture). (Papadakis-Kantos, Migrating IBM Rational DOORS project data to IBM Rational DOORS Next Generation or IBM Rational Requirements Composer by using ReqIF, 2013b)

The migration does not include historical data, but links are created to DNG that link back to DOORS. They provide access to history, baselines and non-migrated data. This means that DOORS is still needed for archiving purposes. Third party applications offer different kind of migration solutions. For example, Opshub's Integration Manager has two migration options. The first option migrates complete audit trail and history with other artifacts. With this option DOORS would not be needed after the migration. The other option is useful if there are DXL scripts in DOORS which are absolutely vital. Similarly, to the first option everything is migrated to DNG, but instead of being able to leave DOORS behind its archived and bi-directional integration between the two keeps them up to date. (Opshub)

DNG, shown in Figure 4, is used via a browser and it has a bit more modern look than DOORS. It does not have all the same functionality as DOORS has, but it is getting regular updates. DNG does not support DXL scripts which means that old scripts cannot be utilized just like that. But it does support scripting using JavaScript which is a more widely known language. This could be enough to inspire users to utilize scripting a bit

more. Similarly to DOORS, scripting can be used to add custom functionality to the application.

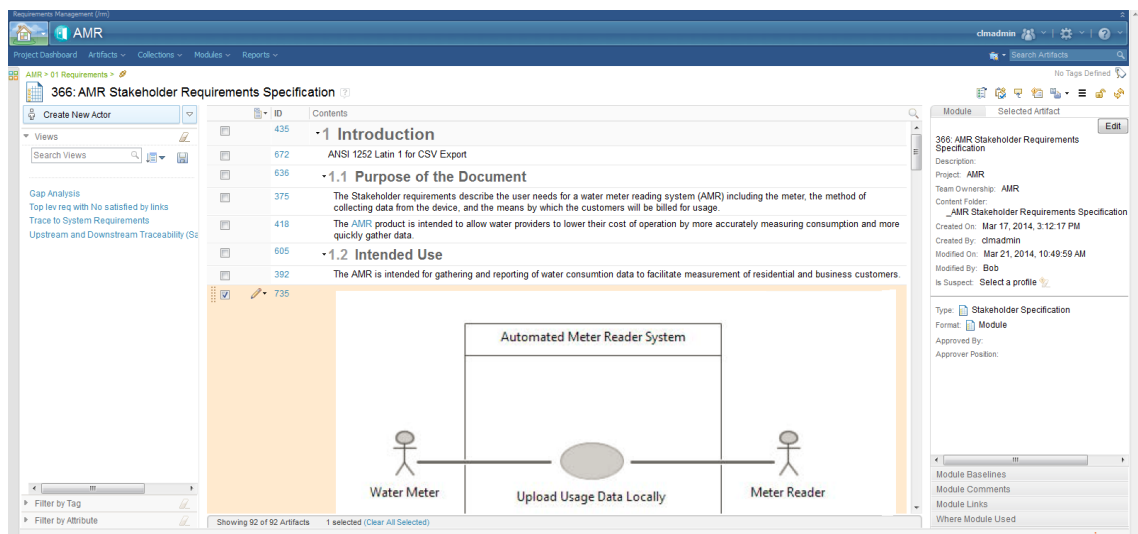


Figure 4. DOORS Next Generation. (IBM, d, n.d.)

Information on whether testing can be done on DNG similarly than with DOORS could not be found. DNG is usually used with Engineering Test Management as they can both be part of IBM Engineering Lifecycle Management. So, the testing related activities are usually done with Test Management.



#### 4.4 Comparison

Comparison of features between the applications is listed in Table 2. DNG is usually used as a part of IBM Engineering Lifecycle Management but in Table 2 it is considered as an individual and if functionality is received with another part of IBM ELM it is mentioned.

Table 2. Feature comparison.

	DOORS	HP ALM	DOORS Next Generation
Can be used via a Browser	No. (DOORS Web Access can be installed separately and has limited functionality.)	Yes (only Internet Explorer by default)	Yes
Defect management	No	Yes	No (With Test Management: Yes)
Testing	Yes	Yes	No definite answer. (With Test management: Yes)
Add functionality to UI with scripting	Yes (DXL)	Yes (VBScript)	Yes (JavaScript)
Supports ReqIF	Yes	Yes	Yes
Task tracking & Agile planning	No	No	No (With Workflow Management: Yes)
Marks related tests if requirement has been changed	No	Yes	Requires Test Management

All of the applications support ReqIF, which makes it easier to share the requirement data and integrate with other applications.

## 5 Research Methods

Both qualitative and quantitative methods were used for the research. For the quantitative method a survey was made for the people using DOORS at the time of the study. The data was used to assess how people find using the application. The results from that survey were utilized with the interviews (qualitative), which were done with people who have experience with both applications. The survey also contained open-ended questions with which the aim was to get also qualitative data.

## 5.1 Survey

A survey was made to find out how people felt about using DOORS, what were the features they liked or disliked and if they would like to switch from DOORS to some other requirements management program. The survey was sent to people who used DOORS in some aspects of their work. The email list contained 86 people and the number of answers received was 21, indicating a response rate of 24.4%.

The statements were taken from USE Questionnaire which has 30 statements. USE stands for Usefulness, Satisfaction and Ease of Use. Those 30 statements are divided to four categories Usefulness, Ease of Use, Ease of Learning and Satisfaction. For the survey two statements from each category excluding ease of learning were taken. The survey has been added as Appendix 1. (Lund, 2001)

The survey was done with the company's own survey tool and the answers were submitted anonymously, but the person answering had the opportunity to leave their email in the survey for possible follow up questions. Even if the survey itself did not save any personal information, to be able to answer the survey the participant had to login to the tool. This might have been an issue for some.

The survey consisted of six multiple-choice statements and some open-ended questions. Additionally, there were a few questions to find out how long and often participants use DOORS and what they usually use it for. There was also a yes/no question on whether the participant would like to change from DOORS to some other program. The statements had 5 choices for an answer from "Strongly disagree" to "Strongly agree" the middle option being "Neutral". The participants were asked to choose the option that best describes their feeling for each statement.

The statements were chosen to mirror some elements of usability as mentioned above. This was done either by directly asking whether DOORS is easy to use, or like with learnability and memorability by comparing the answers of those who use the application more to the answers from others.

The major reason for the open-ended questions was to map the liked and disliked features of DOORS. As an effort to raise the response rate the questions were left optional to ensure that people who were in a hurry would at least answer to the multiple-choice statements. Most participants answered to the open-ended questions especially to the questions about the liked and disliked features and the follow up question based on their answer whether they would like to change from DOORS to another program.

## 5.2 Interviews

The interviews were done 1-on-1 and recorded for later analyzing. The questions were decided before hand, but some extra questions that came up during the interview were also asked, meaning that the format was semi-structured. Half an hour was reserved for the interviews, but they only took around 20 minutes each.

The goal for the interviews was to hear about how the interviewees felt about using HP ALM and to compare it to DOORS. In addition, the problematic parts identified from the survey were discussed. The idea was to find out whether they had witnessed the same or similar issues with HP ALM. Both interviewees had previously used DOORS and were now working with HP ALM. They had used DOORS for years and had started with HP ALM around six months prior. Interview questions can be found from Appendix 2.

## 6 Results and Analysis

The survey's results are split to their own chapters where the statements are analyzed separately from the open-ended questions. The analysis for the open-ended questions was done mostly for each question, but there were some responses which answered to other questions as well, e.g. a disliked feature mentioned with the liked features, so these are taken into consideration with the question which they really answer.

## 6.1 Survey

21 people answered to the survey and 81% of them had been using DOORS for more than 2 years. Most of the participants (43%) used DOORS daily. 29% used it weekly and the rest used it either monthly or less. From the participants who had used the application for more than 2 years, eight used it daily. This was the largest group, when comparing the frequency and length of use. This group's answers are compared to the rest of the participants and they are referred to as the frequent users. Figure 5 illustrates the DOORS usage of the 21 participants.

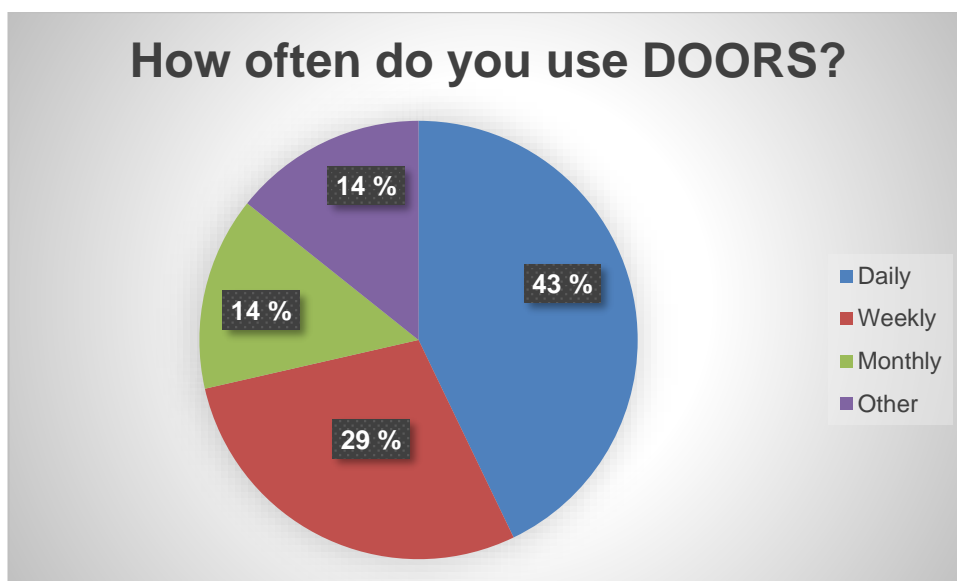


Figure 5. DOORS usage (n=21).

The statements were designed to be positive. For each statement the average score was less than 2,5, 1 being Strongly disagree, 3 Neutral and 5 Strongly agree. This means that the average answer to each of the statements was "Disagree". The frequent users' answers were significantly more positive than the answers from everyone not in that group. Only one of their scores was less than 2,75. The worst and the best scoring statements were the same for the groups, although with the frequent users there was a tie for the first place and with the others -group there was a tie for the last place. The averages for the groups can be seen in Table 3.

Table 3. Statements' average score for the most experienced, others and all.

	Frequent users (n=8)	Others (n=13)	All (n=21)
DOORS is easy to use.	3,25	1,92	2,43
DOORS helps me be more productive.	3,25	2,0	2,48
DOORS is efficient to use.	2,75	1,62	2,05
DOORS is user friendly.	2,25	1,62	1,86
DOORS works the way I want it to work.	3,0	1,77	2,24
I would recommend DOORS to a project.	2,88	1,85	2,24

For all the participants the worst score (1,86) was for statement “DOORS is user friendly”, where all the answers were neutral and below. When looking only at the answers from the frequent users, their score was 2,25. This was easily their lowest scoring statement, but it was also the closest one to the score of all participants. Figure 6 illustrates the user-friendliness of DOORS based on the survey answers.

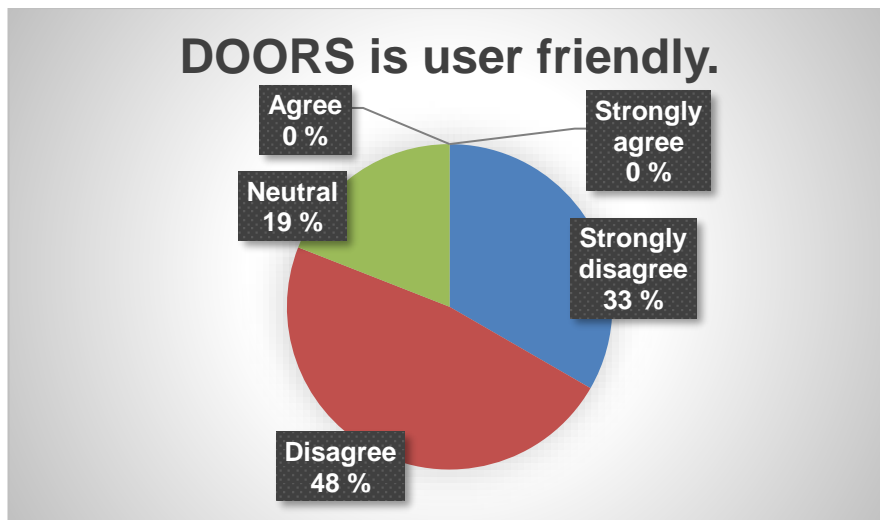


Figure 6. User friendliness of DOORS (n=21).

The best score (2,48) was for “DOORS helps me be more productive”, where other than strongly agree the answers were spread fairly even. The distribution can be seen in Figure 7. With the frequent users the score was 3,25 and with the others -group this statement was the only one with a score above two.

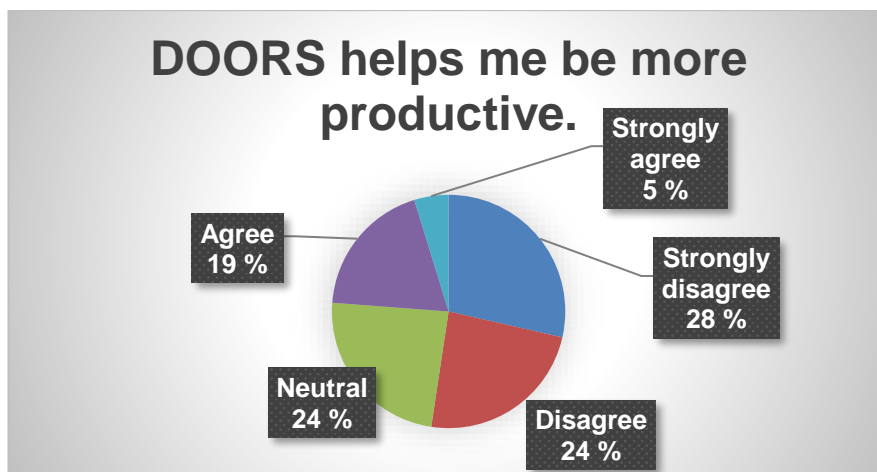


Figure 7. Productiveness of DOORS (n=21).

The biggest difference between the averages of the frequent users and everybody else was for the statement "Doors is easy to use". Where the average for others was 1,92 and the frequent users 3,25. This would indicate that using the application for long enough and frequently enough makes using a bit easier, which is not that surprising.

The most common usages for DOORS were reviewing requirements (66,7%) and updating attributes (61,9%). There were some answers in the other section which could be put to the reviewing requirements category, raising its percentage even higher. Reviewing might have been a bit unclear of a term here. Almost everyone uses the application for multiple things.

All in all, the groups of frequent users and others had similar results. Both had the same two as their highest scoring statements and the same happened with the two lowest scoring statements. The difference can be seen on how strongly they felt about each statement. Most of the statements received a neutral verdict from the frequent users. The results of the others -group were more on the negative side. What should be noted is that the best scoring statement was: DOORS helps me be more productive, which would indicate that the application does provide the tools for the users to complete their tasks.

The results would indicate that the users are not overly happy to use DOORS, as there was only one 'strongly agree' among all the answers. But there are significant differences

between the users who have used the application long and frequently compared to the others. Possible reasons for this could be that learnability and memorability are not high. Meaning that it takes time to learn to use the application and using the application again after a break means learning to use it again. Another reason could be that the users who use the application frequently have got used to it and its flaws. They have accepted and learned how some things are done and what kind of issues there can be.

The question about whether the participants would like to change from DOORS to some other program had a very definitive answer – 76 % would like to change and 24 % would not. Most of the participants who answered ‘no’ were not completely against the change. They were worried about how the transfer of data would work and how it would be done. The new program would have to be a lot better than DOORS and need a proper trial time. They were also worried about how much manual work there would be during the migration.

Participants who answered ‘yes’ to the question were asked: “What would need to be fixed for you to not want to change from DOORS?”. Most of the answers were also found from the disliked features question. Many of the answers can be put under enhancing usability and user experience, such as newer and clearer UI, better performance and speed and how images and tables work.

## 6.2 Open-ended Questions

There were some variety with the answers given to the liked features question. Some things were mentioned only once, but there were some features which were mentioned multiple times. A few of these are related to each other as one is needed to be able to do the other. The most liked features according to the open-ended questions were linking, traceability, filtering and attributes. Linking and traceability are related as linking is needed to provide traceability. Filtering and attributes also relate to each other. The users can create their own attributes to help filter the module the way they want. Many of the answers contained features which could go under customizability. The ability to create new attributes and scripts give the user some freedom on how to use the application. The most liked features could be categorized from the most popular down as follows:

- linking and traceability
- filtering
- customizability (attributes, views and scripts).

The number of disliked features seemed to be higher than that of the liked ones. There were some issues which were mentioned by multiple people e.g. the difficulty of using tables were mentioned in about third of the answers. Updating tables by adding rows/columns/cells and analyzing steps is very laborious. During export long header texts on the tables cause trouble. The UI received also some critique for how it looks and feels, old. It was also described as unintuitive, visualizing the structure is hard and finding anything can be difficult.

The slowness of the program and long load times have also been noticed. These might be related more to the fact that DOORS is used via Citrix and the server is located in the US, so the latency times might not be optimal. This is most likely the culprit to some of the connection issues which include Citrix crashes and connection losses. These can cause other problems e.g. the module to be unable to be edit, as it is already locked by the same user.

Pictures and other attachments may cause some problems with exporting. It requires special scripting, is slow and error-prone if different views are wanted. The fact that the scripting is done with DXL (DOORS Extension Language) was split with some of the answers. Some found it to be alright but most who mentioned the language found it complex or would like to be able to use some other language for the scripting.

Some issues or similar issues were mentioned by multiple people. These have been listed below:

- slowness, connection issues
- tables
- UI
- window management
- scripting (complex, poorly supported)
- exporting
- Are there features that are not known about?



- No notification for tests if linked requirement is changed.

When asked what would need to be fixed with DOORS for people to want to keep using it, many of the answers could be put under improving usability and performance of the application. Some of the usability issues would be fixed with improved performance e.g. slowness of the application, which was mentioned multiple times in the answers.

Other answers mentioned things that might be able to be done with new scripts or by improving the scripts in use, e.g. simplifying the exporting. The last group contains suggestions that cannot be done, at least with the DOORS version in use. These would be a modern and improved UI, easier way of attaching screenshots and enhancing the way tables work.

It seems that the high customizability is seen as a positive thing, but it also seems to cause some challenges. For example, the scripting is seen as a good thing and has potential to be very useful, but actually utilizing it is complex. Same with attributes and views which were some of the most liked features, but there is a downside to them as well. There might be attributes which are not used anymore or are used rarely, and their management can be error-prone. If no changes are made to the settings, a newly saved view will be made the default view for that module. This means that the view is made public and when this is done multiple times the view list gets crowded with views, that might not be used anymore or were meant for personal use from the beginning. One reason could be that the instructions are not clear or easily findable.

Usability issues can be given a severity rating. This rating is usually on a four or five-point scale from cosmetic to catastrophic, where cosmetic does not really cause any problems and catastrophic prevents the user from completing their tasks every time. Most of the usability issues found in the survey could be given a severity rating of minor or moderate, depending on whose rating is used, as they do not seem to prevent the users from completing their tasks. (Sauro, 2013)

The issues may slow down or frustrate the user, but other than crashes, there were not any mentions of loss of data. In some cases, the reason for loss of connection might have been due to idling for more than two hours, after which the user is disconnected.

This is to prevent users from accidentally occupying a DOORS license for excessive amount of time.

### 6.3 Interviews

The first impressions of HP ALM were that it is more complicated than DOORS. It seems to have a lot of features but getting to know them can be tricky. One reason for this could be that things are done differently between the applications and having got used to the DOORS way, making the switch can take some time. Also, the need to use Internet Explorer to use HP ALM was reported to feel a bit strange.

There has been some connection issues and the application can be a bit slow sometimes, but there has not been any crashes or timeouts. For that part the application was said to feel more stable.

Verification has not yet started on the other project so there is no first-hand experience how HP ALM will perform during it. Overall the plans are still somewhat open on how to do some things on HP ALM. One of these is how possible regression rounds are handled during verification.

There is a difference on how traceability between tests and requirements is handled. In HP ALM requirements are linked to the set of tests which were run. With DOORS each requirement is linked to verify the steps. This raises a question on how regression rounds are done with HP ALM. With DOORS the failed steps can be retested as all verify steps are linked to the requirements directly. For example, steps A and B are in the same module, but step B was failed. Instead of retesting everything in the module only step B is retested unless the previous steps are part of the testing flow and are required to get to step B. This can be done because that one failed step does not cause the whole module to fail and the requirements are linked to passed results. With HP ALM if one test fails in a set it shows that test set as failed. So, the requirements which tests passed are linked to a test set with failed result. If this is a problem that means the whole test set would need retesting, which would cause a lot of unnecessary repeated testing.

There are also significant differences with testing. Automated tests can be run from HP ALM, but this will most likely not be used and the tests are run some other way. For manual tests it seems that only one step at a time can be seen. This could be problematic in some cases especially if the test case is not familiar to the tester. With DOORS, the tester can see all the steps that fit to the screen which is helpful if the test requires quick actions e.g. changing values rapidly or timing something. Knowing what should be done next is very useful and helps maintaining a flow with the testing.

If new attributes are needed for HP ALM they must be requested through a ticket which can take some time. With DOORS the user can create the required attribute themselves.

In HP ALM data can be exported with scripts in a format that can be utilized to create graphs about the progression. For example, these graphs can be used to check the test coverage of requirements.

One of the reasons HP ALM was chosen is that it provides defect management. This is not used yet as the project is still in development phase, but the idea is to start using it during verification to link SPRs (Software Problem Report) from ClearQuest, which is the official defect management application.

For now, the preference would seem to be to use DOORS as it is the better known one and has a collection of tools e.g. scripts which were ready to be used. Work done with DOORS one way has to be done completely differently on HP ALM and some work requires exporting documents to Excel and the actual work is done there.

#### 6.4 Recommendations

Multiple issues were found with the survey. The most common issues were slowness and connection issues. Unfortunately, there is no magic bullet that would fix all of them just like that. The possible solutions are gathered underneath with their positive and negative sides and what needs to be taken into consideration with each solution.

The best solution would be to improve the usability of DOORS to such a level that the users would be happy or willing to continue using it. Then there would not be a need to

migrate to another application which has a change of something going wrong and the users would not need to spend time to learn to use another application. Changing to another application would also be a major investment for a company with all the licenses that would need to be purchased. Of course, there are issues which will displease some users, such as having to use DXL instead of some other newer or at least more common language. Another issue which was high on the list of the disliked features that cannot be changed if continuing to use the same application is the UI, at least it cannot be modernized. Scripts could add some functionality, but the appearance would stay the same. So, the way to enhance the usability of the application would be to fix the connection issues and increase performance.

In theory the connection to the Citrix server should be fine as the ping times are usually less than 150 milliseconds. According to their documentation the connection should provide great user experience. The next step would be to check the latency between the DOORS client and server to see whether it is under 50 milliseconds. Of course, it would be great if the latency could be brought even lower, but the 50 milliseconds is the high limit which is mentioned in DOORS documentation after which the performance may degrade. What was not tested was the impact of the firewalls and whether there would be a significant difference in latency, if users connected to Citrix via a VPN (Virtual Private Network).

One way to lower the latency times for users in Finland would be to host the servers from Finland, but this could be quite an expensive way to solve the problem, so it would be wise to try different methods first. Before doing this some comparison should be made with users who are in the same facility as the servers now. This would show if relocating the servers would provide any improvement. If the results would show only minor upgrade, it might not be worth it to relocate the servers.

Having the servers on-site should help with performance according to a previously done report. In 2013 a DOORS Health Check was conducted by IBM, which found that the local site had no performance issues, but the remote site did have them. According to the report there were no performance issues with DOORS, but the issues were due to the response times between Citrix server and client.

DOORS does not support mirroring of servers, so there would be a high chance of data corruption. The only way to use two servers would be to take them down for synchronization, which would mean that they could not be used at that time. This means that to improve the performance for users in Finland the server would need to be relocated there and the performance would drop for the users in the US. When deciding where to locate the server a couple of things should be considered. Firstly, where are the users that need to use DOORS the most. Secondly, which location can provide the fastest connections. Finally, would there be a better location that would be better suited to be the central from geographical point of view. As there are other locations than US and Finland which use DOORS, would the servers be closer to them if the server were located in Finland and would it help with DOORS' performance on their end.

There are newer versions of DOORS available for the client and server. The client version is 9.6.1.8 and server is 9.6.1.6. The newest version available is 9.6.1.11. These are fix packs, not completely new versions of the application. They contain a couple of fixes which are marked as performance fixes, but most of the fixes seem to be miscellaneous bug fixes or minor quality of life improvements. Most seem to be client-side fixes with only a few server-side fixes and with some the documentation does not mention which side the fix is for. (IBM, 2018)

Some DOORS elements may lower the performance of the application e.g. large tables or large number of tables and OLE objects. Already the instructions are to avoid adding tables to modules if possible. This is mostly because it is very time consuming to update and modify the attributes of the cells. It might be good to check if some of the tables could be changed into normal steps. In some cases, this is not feasible as the only efficient way to have the data is in a table. The number of OLE objects loaded at a time can be defined and thus improve the load times. The recommended value is 5 as any more than that probably would not fit on the screen simultaneously. This requires a change in the registry, so it should be done by someone experienced in the area as errors may require re-installation to be corrected. (IBM, 2013)

Using shareable edit too much can have an effect on the performance. Every shareable edit section creates a new file in the database that needs to be loaded. The shareable edit level should be 1, 2 or 3 especially in large modules. With test modules this is not

advisable as it would reduce the number of people who can work with the module at the same time. It would be good to open the modules in exclusive mode occasionally and save them. This will improve the performance by rolling the small files in the database into a larger file. (IBM, 2013)

Another solution would be to trial other requirement management applications. Especially if there is a need for a more complete lifecycle management solution. Most of the applications offer limited time free trials, usually from 30-60 days. During this time a smaller group of users would use the application to see whether it would offer a significant enough upgrade. One major part of this would be to test how the migration from DOORS is handled with the application. Trying to migrate everything is not necessary, but migrating some modules is needed. The modules should not all contain just normal objects but also tables and pictures to see what kind of issues could be met when migrating everything. Even if the applications were free for the trial it is an investment because the users need to spend time to test out the applications.

But which application to choose? The wanted toolset affects to the decision. Is it just a direct replacement for DOORS, which would offer only requirements management and testing or a single platform solution, like an application lifecycle management program? These provide most of the required applications without the need to jump from one application to another.

HP ALM offers an attractive combination of tools which should be able to provide the requirements management, testing and traceability that DOORS does now. Additionally, HP ALM also has defect management. Compared to other options, the fact that there are people in the same building who have used HP ALM for a while now and are more familiar with it, could be helpful in case there are problems or questions that rise. But the initial feeling from the interviews was that DOORS would be the preferred option. There will be a better control point after verification has been done once using HP ALM. Some of the open questions about the capabilities of the applications can be answered than.

Doors Next Generation is part of IBM's application lifecycle management toolset. It can collaborate with other tools to provide similar collection as HP ALM. The benefit would be that it has a lot in common with DOORS. Therefore, the transition from one to another

might be easier and the migration more straightforward. IBM offers solutions which include a certain set of tools. IBM Engineering Lifecycle Management Base includes three applications: Engineering Workflow Management, Engineering Test Management and Engineering Requirements Management DOORS Next Generation. Test Management and DOORS Next Generation should offer the same features as DOORS, plus defect management. Workflow Management helps with task tracking and agile planning which is done with another application at the moment. If Workflow Management is not needed maybe getting only Test Management and DOORS Next would be enough. If a move to a new requirements management application is done it would be recommendable to host the server on the local site to avoid connection issues.

## 7 Conclusion

The goal for this thesis was to find out how the users felt about using DOORS and what were the features they liked and disliked and whether there were some ways of improving the usability. DOORS was also compared to another requirements management application HP ALM. Additionally, DOORS Next Generation was also taken into consideration as a possible alternative. The goals were achieved on most parts. The users' satisfaction for DOORS and the liked and disliked features were mapped with a survey. Comparison with HP ALM and DOORS were done with literature and two interviews. Concrete tests between the two were not conducted as to for example whether there is a difference on how fast a requirement can be made.

The results confirm the presumption that DOORS users are not completely happy with the application. The opinion is lower with the people who use it less frequently. Most would be willing to change to another requirements management application, if there is enough of an improvement to make it worth it. Based on the interviews it seems that it might not be worth it to change to HP ALM. The number of interviews is so low that generalizing the results to all users has its obvious risks. What raises the confidence for this conclusion is that in some unofficial conversations the feelings for HP ALM have been similar.

The reliability of the survey is on a higher level as almost a quarter of the people who the survey was sent to answered to it. What could distort the conclusions is that the participants' job description was not asked (developer, tester, etc.) and the number of answers make it so that each answer has a large impact on the results. Therefore, there is a chance that some groups' opinions are underrepresented in the survey results or an outlier answer could have more impact than it should.

The next step would be to implement the "cheaper" solutions, which would not require relocating anything or changing applications and see whether they would provide enough of a performance boost. After that it would need to be decided if the results are good enough or are more extreme solutions needed such as relocating the servers, or, whether there is a need for an ALM like application which offers some additional functionality compared to DOORS, similar to the two mentioned in the study: HP ALM and IBM ELM.

## References

- Cheng, M. (2003). *Medical Device Regulations: Global Overview and Guiding Principles*. World Health Organization.
- Contreras, M. (2017). *How Network Latency Impacts User Experience*. Retrieved April 29, 2019, from <https://www.citrix.com/blogs/2017/09/25/how-network-latency-impacts-user-experience/>
- GE Healthcare. (2018, September 7). *CARESCAPE ONE | GE Healthcare*. Retrieved from <http://global.gehealthcare.com/en-gb/products/carescape-one?cookie=906&account=36813&>
- GE Healthcare. (n.d.). *CARESCAPE Monitor B650 | GE Healthcare*. Retrieved September 6, 2018, from <https://www.gehealthcare.com/products/patient-monitoring/patient-monitors/carescape-monitor-b650?account=36813&>
- Guru99.com. (n.d.). *Introduction to HP ALM(Quality Center)*. Retrieved from <https://www.guru99.com/hp-alm-introduction.html>
- HP. (2010). *HP Application Lifecycle Management: Tutorial*.



- IBM. (2013, November 12). *Tips to improve Rational DOORS performance*. Retrieved January 22, 2019, from <https://www-01.ibm.com/support/docview.wss?uid=swg21324458>
- IBM. (2018). *Fix list for Rational DOORS and DOORS Web Access*. Retrieved from <https://www-01.ibm.com/support/docview.wss?uid=swg27036720#96111>
- IBM, a. (n.d.). *Rational DOORS on Citrix*. Retrieved February 11, 2019, from [https://www.ibm.com/support/knowledgecenter/en/SSYQBZ\\_9.5.0/com.ibm.doors.install.doc/topics/c\\_installingcitrix.html](https://www.ibm.com/support/knowledgecenter/en/SSYQBZ_9.5.0/com.ibm.doors.install.doc/topics/c_installingcitrix.html)
- IBM, b. (n.d.). *Performance tuning for the Rational DOORS database and client*. Retrieved October 18, 2018, from [https://www.ibm.com/support/knowledgecenter/en/SSYQBZ\\_9.6.1/com.ibm.doors.administering.doc/topics/r\\_improve\\_perf\\_db\\_client.html](https://www.ibm.com/support/knowledgecenter/en/SSYQBZ_9.6.1/com.ibm.doors.administering.doc/topics/r_improve_perf_db_client.html)
- IBM, c. (n.d.). *Overview of Rational DOORS Next Generation*. Retrieved from [https://www.ibm.com/support/knowledgecenter/en/SSYQBZ\\_9.5.2/com.ibm.doors.requirements.doc/topics/c\\_overview\\_dng.html](https://www.ibm.com/support/knowledgecenter/en/SSYQBZ_9.5.2/com.ibm.doors.requirements.doc/topics/c_overview_dng.html)
- IBM, d. (n.d.). *Rational Doors Next Generation*. Retrieved May 15, 2019, from <https://jazz.net/products/rational-doors-next-generation/>
- International Medical Device Regulators Forum. (n.d.). *About IMDRF*. Retrieved September 10, 2018, from IMDRF: International Medical Device Regulators Forum: <http://www.imdrf.org/about/about.asp>
- Kramme, R., Hoffmann, K.-P., & Pozos, R. (2011). *Springer handbook of medical technology*. Springer-Verlag Berlin Heidelberg.
- Levensen, N. G., & Turner, C. S. (1993). An investigation on the Therac-25 accidents. *Computer (Volume: 26, Issue: 7, Pages 18-41)*.
- Lund, A. (2001). Measuring Usability with the USE Questionnaire. *Usability and User Experience Newsletter of the STC Usability SIG*. 8.
- Mäder, P., Jones, P. L., Zhang, Y., & Cleland-Huang, J. (2013). Strategic Traceability for Safety-Critical Projects. *IEEE Software ( Volume: 30 , Issue: 3 , May-June 2013 )*.
- Mäder, P., Olivetto, R., & Marcus, A. (2017). Empirical studies in software and systems traceability. *Empirical Software Engineering*, 963.
- Nielsen, J. (1993). *Usability Engineering*. Academic Press.
- Nielsen, J. (2012). *Usability 101: Introduction to Usability*. Retrieved October 3, 2018, from <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>

- Opshub. (n.d.). *How to migrate from IBM Rational DOORS to DOORS Next Generation (NG)*. Retrieved May 8, 2019, from <https://www.opshub.com/white-papers/migrate-from-doors-to-doors-next-generation/>
- Pandey, D., & Pandey, V. (2012). *Importance of Requirement Management*. Retrieved from <https://pdfs.semanticscholar.org/ad85/93ce74f392ce1924eadd420762d258711a8b.pdf>
- Papadakis-Kantos, Y. (2013a). *IBM Rational DOORS getting started*. Retrieved November 13, 2018, from <https://www.ibm.com/developerworks/rational/library/getting-started-ibm-rational-doors/index.html>
- Papadakis-Kantos, Y. (2013b). *Migrating IBM Rational DOORS project data to IBM Rational DOORS Next Generation or IBM Rational Requirements Composer by using ReqIF*. Retrieved May 16, 2019, from [https://www.youtube.com/watch?v=6GagDzEIXKc&list=PLZGO0qYNSD4W1a9eDXdhrzKlogfFWTP\\_t&index=2](https://www.youtube.com/watch?v=6GagDzEIXKc&list=PLZGO0qYNSD4W1a9eDXdhrzKlogfFWTP_t&index=2)
- Regan, G., Caffery, F. M., Daid, K. M., & Flood, D. (2013). Medical device standards' requirements for traceability during the software development lifecycle and implementation of a traceability assessment model. *Computer Standards & Interfaces (Volume 36, Issue 1, November 2013, Pages 3-9)*.
- Rempel, P., & Mäder, P. (2016). Continuous Assessment of Software Traceability. *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*.
- Rubin, J., Chisnell, D., & Spool, J. (2008). *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. John Wiley & Sons, Incorporated.
- Salmenperä, M. (2014). Potilasturvallisuuden perusteet: Potilasturvallisuuden käytön vaikutus potilasturvallisuuteen. In L.-M. Aaltonen, & P. Rosenberg, *Potilasturvallisuuden perusteet*. Duodecim.
- Sauro, J. (2013). *Rating the severity of usability problems*. Retrieved June 5, 2019, from Measuring U: <https://measuringu.com/rating-severity/>
- Terveyden ja hyvinvoinnin laitos. (2019). *Potilasturvallisuus*. Retrieved January 22, 2019, from <https://thl.fi/fi/web/sote-uudistus/palvelujentuottaminen/potilasturvallisuus>

- US Food and Drug Administration. (2005). *Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices*.
- US Food and Drug Administration. (2017). *CDRH Mission, Vision and Shared Values*. Retrieved September 20, 2018, from <https://www.fda.gov/AboutFDA/CentersOffices/OfficeofMedicalProductsandTobacco/CDRH/ucm300639.htm>
- US Food and Drug Administration. (2018a). *What does FDA regulate?* Retrieved September 20, 2018, from <https://www.fda.gov/AboutFDA/Transparency/Basics/ucm194879.htm>
- US Food and Drug Administration. (2018b). *Drug, and Cosmetic Act (FD&C Act), FDA regulations, and FDA guidance?* Retrieved June 17, 2019, from <https://www.fda.gov/about-fda/fda-basics/what-difference-between-federal-food-drug-and-cosmetic-act-fdc-act-fda-regulations-and-fda-guidance>
- Wiegers, K. (2013, January 16). *Requirements Traceability: Links in the Requirements Chain, Part 1*. Retrieved from <https://www.jamasoftware.com/blog/requirements-traceability-links-in-the-requirements-chain-part-1/>
- Vogel, D. A. (2011). *Medical Device Software Verification, Validation, and Compliance*. Artech House.

## Survey

### Background

I'm doing my Bachelor's Thesis and the subject is related to DOORS and how people find using it. With this questionnaire I'm trying to find out if people like using DOORS and what are the features they like or dislike.

Please answer to the questions in English or Finnish.  
Mandatory questions are marked with a \*.

**Disclaimer: This questionnaire is set to Anonymous, but if you choose to leave your email I can see which answers are yours.**

### \* How long have you used DOORS? ☺

- Less than a year.
- 1-2 years.
- More than 2 years.

### \* How often do you use DOORS? ☺

- Daily
- Weekly
- Monthly
- Other

If you chose Other, please write your answer here.

### \* What do you usually use DOORS for? ☺

- Writing requirements
- Reviewing requirements
- Writing test procedures
- Testing
- Reviewing test results
- Updating attributes
- Other

If you chose Other, please write your answer here.

**Statements**

Please choose the option that best describes how you feel about each statement.

\* DOORS is easy to use. ☺

Strongly disagree     Disagree     Neutral     Agree     Strongly agree

\* DOORS helps me be more productive. ☺

Strongly disagree     Disagree     Neutral     Agree     Strongly agree

\* DOORS is efficient to use. ☺

Strongly disagree     Disagree     Neutral     Agree     Strongly agree

\* DOORS is user friendly. ☺

Strongly disagree     Disagree     Neutral     Agree     Strongly agree

\* DOORS works the way I want it to work. ☺

Strongly disagree     Disagree     Neutral     Agree     Strongly agree

\* I would recommend DOORS to a project. ☺

Strongly disagree     Disagree     Neutral     Agree     Strongly agree

**Open-ended questions**

Please answer to the open-ended questions in English or Finnish.

What DOORS features do you like?

0% of 4000 characters

What DOORS features do you dislike?

0% of 4000 characters

The next questions are about using some other program than DOORS.

Depending on your choice, please answer to the follow up question.

\*Would you like to change from DOORS to another program? ☹

No  Yes

No -> Why not?

0% of 4000 characters

Yes -> What would need to be fixed for you to not want to change from DOORS?

0% of 4000 characters

Is there a feature in DOORS that you would like to have in the new program? Which one?

0% of 4000 characters

If there is something else you would like to say about DOORS that wasn't asked in this questionnaire you can write it here.

You can leave your email here if you would like to be contacted for possible follow up questions.

## Interview questions

What do you use HP ALM for?

How long have you used HP ALM and how long did you use DOORS?

How do you find using HP ALM? User friendly?

Can you see any differences with the efficiency of use?

How would you say using HP ALM compares to DOORS?

Are there DOORS features which you miss when using HP ALM?

Does HP ALM support scripting?

Pain points from survey: tables, UI, connection issues, scripting, window management.

Have there been any similar issues?

Did you have any issues with DOORS?