Software Quality and Testing

Exercise 1

Ida Lehtomäki

Autumn 2024

Prework

Software quality is measured with a set of quality attributes, which are described in the ISO 25000 Standard as functional suitability, performance efficiency, interaction capability, reliability, security, maintainability, flexibility and safety. Software quality that is considered good quality, matches the needs of stakeholders and therefore brings value to them. Prioritizing quality attributes needs to be personalized for each project, so that stakeholder needs are fulfilled. MoSCoW is a well-known method for prioritizing. General rule for software quality is that it needs to be reliable meaning the software works, functional so that the functions truly help execute a task and maintainable. Security means that unwanted and illegal activity in the software is prevented. Safety is a broader term, which evaluates whether the software is reliable as well as safe to use. It aims to tackle both deliberate harm and accidental harm.

Based on theory about Iron triangle, says that to build good quality software one needs either time or money. If the requirement is to build good fast, it costs. If there is time good quality can be accomplished cheap. To build good quality is cheaper in the long run than to later go back to do drastic changes on a bad system.

I have analyzed the OSL-app in my Human Centered Design -course, so I'm using It as an example of a bad software, although the application is not entirely bad. The application is used to buy tickets and search public transportation routes. In the app there are no additional information available while completing a task. Therefore, the functional suitability for the stakeholders is not matched, because the stakeholders of a public transportation system are a broad range of people from different technical backgrounds.

I personally think VR-system is a good software to use, because in my experience it is reliable and safe to use. I use the system frequently and only one time I remember to have had a difficulty with proceeding a payment. The ticket I purchased wasn't visible in the application and it was only sent to my email. Overall, I think the VR-application is quite easy and fast to use, so it fulfills the quality measurements performance efficiency and interaction capability, hence the payment happens in another system. I'm looking forward analysing it from a more critical perspective later in this excersize.

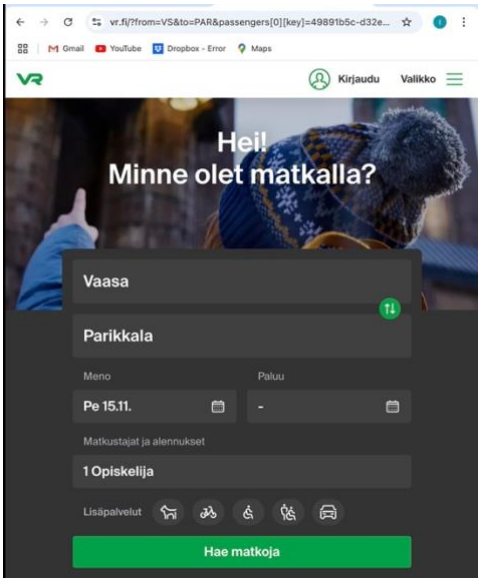
References 14.11.2024:

https://www.perforce.com/blog/kw/software-safety-vs-security-whats-different

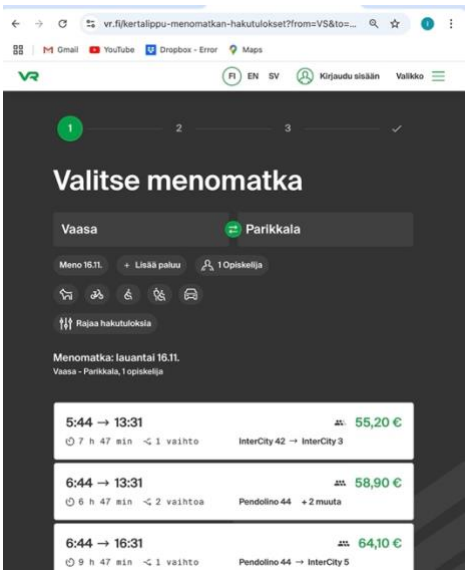
https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

Exercise 1

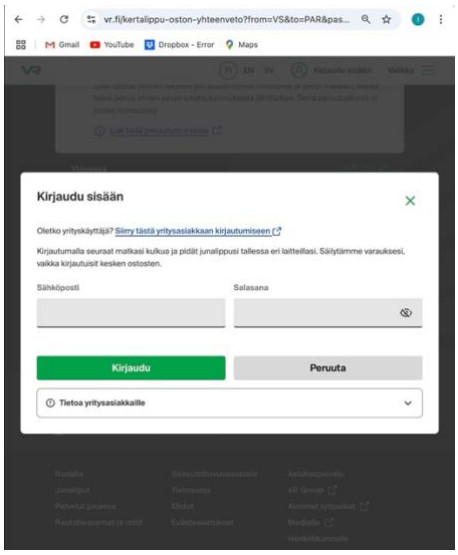
Description of the software



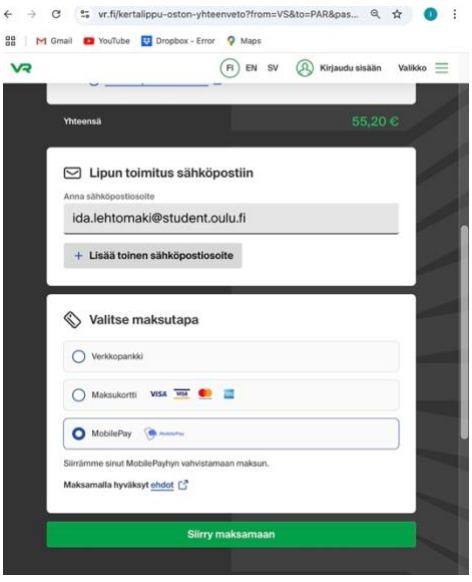
The VR web applications main functional task is the purchase of tickets, which is done by choosing the start and destination, travel date and other factors that contribute to the price of tickets like a student status as shown in the screenshot.



After setting up the filtering factors the system works as a search engine for available routes. In this frame the user can browse available routes and pick the most suitable one to continue.



It is possible to log in before proceeding to payment. Logging in functionality allows the ticket to be stored in the VR-account in addition to ticket being sent to email. By logging in it is possible to activate a business status, if purchasing tickets with a specific company deal.



If ticket is purchased without logging in, the ticket is simply sent via email. After choosing the payment method the website transfers the payment to the chosen company's site and after a successful payment directs the transaction back to the VR page.

What qualities are important and why?

For any system the basic important qualities are that a system must work and not crash. The system must be stable and be able to execute a task. The system must show accurate data.

For a government owned railway network -system it is essential to be reliable and usable. The ticket booking system is widely used nationwide and therefore it needs to always be usable and available for everyone. Users are a various group with a broad range in both age and technical backgrounds. Because VR (Finnish railway network) is the only nationwide network there is no other options in case of a system crash. Fault tolerance and recoverability standards are high for this system.

The booking system has an algorithm, which calculates the route options and recommends the fastest route. It is crucial for the system's reliability that the algorithm works correctly and calculates the train timetable accurately when choosing connecting trains.

Ticket booking system is connected to different payment methods and security in payments is an important attribute. Integrity of the databases is also important with any booking system, ensuring one ticket is sold only once and there is no overlapping with other bookings.

For a widely used system maintainability is also important. Especially for a public sector system, which aims to be affordable and usable for a long time. Public sector doesn't often sell away their systems, so maintainability is important.

Functional Suitability attribute analyzing

Name	Functional Suitability
Description	Functional suitability measures how well a system meets stakeholder needs. Does the system have functions it was supposed to have, and does it work the way it is supposed to?
Current State	If the VR systems goal is to show accurate routes and allow users to purchase tickets and set different price attributes the system meets the functional suitability measures and therefore is good.
Benefits	Creating a simple system with only the essential functionalities is cost efficient and user-friendly, because the system stays simple to use if there are no “extra functionalities”.
Risks	When creating a simple functionally suitable system the requirements engineering is highly important. If the requirements engineering fails, so does functional suitability of the system. Good testing and under involvement is essential in the design process.
Quality indicators / metrics	Requirements coverage = number of implemented requirements / total requirements, gives a percentage, which says how many engineered requirements have been successfully implemented. Feature utilization rate = actively used features / implemented features, this measures whether all implemented features are being used, which brings insight if the stakeholder needs have been understood properly and only useful features implemented. User satisfaction score, average rating from users about the functionalities of the software.
Ways to achieve and ensure each quality attribute	Successfully executed requirements engineering with adequate amount of user engagement and enough of testing of the features.

The most important quality attribute: Reliability

Name	Reliability
Description	Reliability measures how well a system completes a task and how much a system can be trusted. Reliability tackles the time when system can be used (availability), fault tolerance and recoverability.
Current State	Reliability is good, because there are very few errors. The booking system shows the routes and connects to different payment methods without errors.
Benefits	Higher user trust and repeat use. Less revenue loss, because higher availability and less errors in a booking system means less failed bookings, so less lost money.
Risks	Hardware or network failures, not enough server capacity, faulty implementation of failover mechanisms, Data inconsistency due to lack of transactional integrity in databases.
Quality indicators / metrics	Defect Density = amount of defects / size of the software (for good quality systems defect density should be < 0.5 per thousand lines of code Accuracy rate = correct outcomes / outcomes * 100, gives the percentage of correct outcomes. Ticket booking platforms accuracy rate considered good if from 98%-99.9%.
Ways to achieve and ensure each quality attribute	Automatic transfer to backup components when primary fails. Modular software architecture. Automated testing, stress and load testing, fault injection testing.

The least important quality attribute: Aesthetics

Name	Usability → Aesthetics
Description	Aesthetics means the visual representation of the system. Colors, fonts, icons and how well they represent the brand image etc.
Current State	VR-has a very recognizable brand image with the bright green color. The system is in harmony with the color and images. Good stage.
Benefits	Clear and organized design improves user-experience and makes it easier to navigate in the system. Recognizable visual brand image can improve trust and bring revenue with improved retention.
Risks	Focus on visual design elements could take away from other core functionalities, so there need to be clear priority of these. Focusing on beautiful design could take away from easy usability. Very artistic or interesting visual design might not be the easiest to navigate.
Quality indicators / metrics	User feedback surveys, usability testing, task completion success rate, Design audits, which test the consistency of visual brand like fonts and color.
Ways to achieve and ensure each quality attribute	User-centered design, comprehensive branding, prioritizing accessibility in design - less is more sometimes.