

Project Plan

Company 1 - TDDC88

Contents

1	Introduction	3
1.1	Background	3
1.2	Purpose	3
1.3	Goal	3
2	Scope	3
3	Delivery to customer	3
4	Organization and employees	4
4.1	Roles	4
4.2	Cross-functional teams	6
5	Processes	6
5.1	Communication and meeting processes	6
5.2	Development and GitLab processes	7
5.3	Regulatory documents	8
5.4	Requirements	8
5.5	Risk process	9
5.6	Change of processes	9
6	Milestones and timeplan	9
7	Risks	14
A	Cross-functional teams	17
B	Development process	17

Version	Author	Updates	Reviewed by	Date
0.1	A. Nilsson & S. Gharedaghi	Initial version	D. Ma	2021-09-16
0.2	A. Nilsson & S. Gharedaghi	Updated from feedback and added SOP&Risks	D. Ma	2021-10-15
0.3	D. Ma	Minor fixes in section 4	E. Sköld	2021-11-02
0.4	S. Gharedaghi	Adding roles description in section 4	D. Ma	2021-11-06
0.5	D. Ma	Developed section 5.1, 5.3, added subsection 5.5	E. Sköld	2021-11-08
0.6	D. Ma	Developed section 5.1-5.3	A. Nilsson	2021-11-17
0.7	A. Nilsson	Added risk process in chapter 6, updated CFT description in section 4.2	D. Ma	2021-11-19
0.8	F. Dolk	Added requirements process in 5.4	D. Ma	2021-11-20
0.9	E. Sköld	Updated section 4 with role changes	D. Ma	2021-11-24
1.0	D. Ma	Updated Git process in section 5.2	E. Sköld	2021-11-30
1.1	S. Gharedaghi	Updated Risks context and table in section 7	A. Nilsson	2021-12-03
1.2	S. Gharedaghi	Updated Milestones in section 6	A. Nilsson	2021-12-05
1.3	S. Gharedaghi	Grammatical Review	A. Nilsson	2021-12-07
1.4	S. Gharedaghi	Removed part about customizability in section 1.1	D. Ma	2021-12-07

1 Introduction

1.1 Background

The regional health care in Östergötland uses many different systems and services to keep track and get information about patients. This causes some problems since it is a time-consuming process and can, in the worst case, cause a lack of safety for the patient if the information can not get delivered in time. Patient information is documented on a paper journal that is only available for the person that happens to have the journal at the time. This also causes problems with sharing and getting the correct information. We have been tasked to digitize their journals and present all relevant information on a user-friendly application, which should be easy to use and understand. The system shall also be mobile and work with different units. For more requirements, please view the requirements document.

1.2 Purpose

The project aims to make patient information more accessible for the employees at the regional health care in Östergötland.

1.3 Goal

The project goal is to develop a system that provides the regional health care in Östergötland with relevant and accessible information in a user-friendly and mobile way.

2 Scope

The following Work-Breakdown-Structure (WBS) shows an overview of what the project aims to fulfill and how we will achieve it.

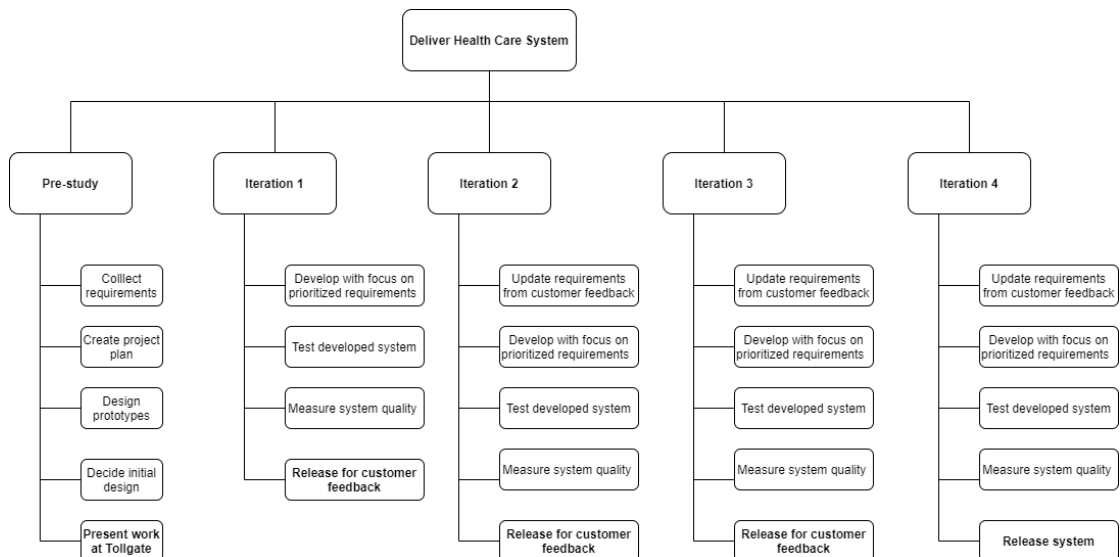


Figure 1: Work Breakdown Structure

3 Delivery to customer

The system is going to be developed in four iterations that all end with delivery to the customer. After each delivery, we seek feedback and thoughts from the customer to get an idea in what direction we should continue with the project. The feedback will help us develop new or update

our existing requirements. After the fourth iteration, we will present the system at VSSE'21 - Valla Software and System Expo on the 16th of December 2021.

4 Organization and employees

The company is divided into two departments, Product & Sales (P&S) and Research & Development (R&D). The focus of the P&S department is mainly customer contact, requirements, and testing. The R&D department mainly focuses on topics regarding design and implementation. From these teams, we have created cross-functional teams with specific tasks. These teams have developed during the project to focus on different tasks as the project's needs have changed, see more under 4.2

4.1 Roles

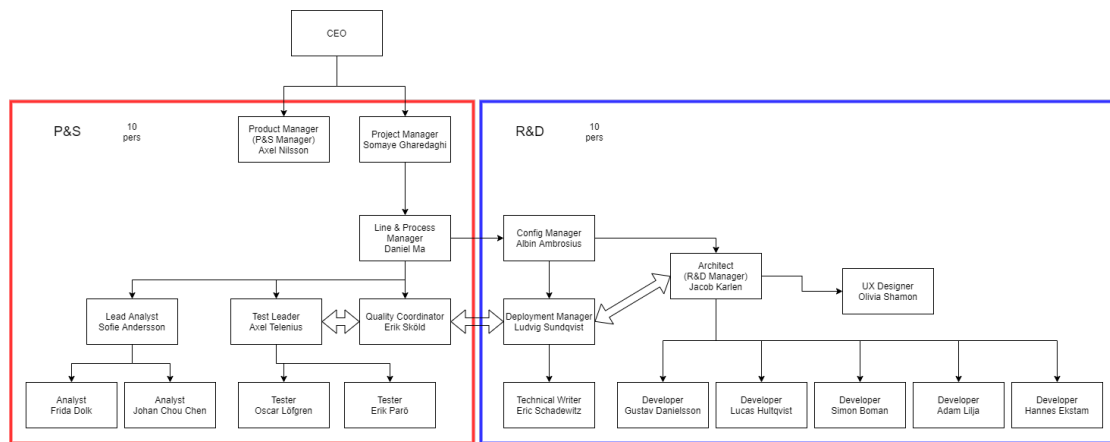


Figure 2: Company Flowchart

Product Manager – Axel Nilsson

A strategic product owner is a link between the customer and the developers. This means handling the product backlog and prioritizing requirements in the best way possible to match customer needs. Also, creating a vision for the product so the whole team knows what we are working towards.

Project Manager – Somaye Gharedaghi

Breaking down the work by creating the project plan, making sure that everyone has something to do in a timely manner, making sure that everyone has the view of the plan to know how they can contribute and collaborate, running the meetings, and keeping the project organized. These all are achievable through regular managers meetings or one by one, keeping in touch with managers, listening to their concerns, and setting agenda for meetings according to following up the progress of the project and managers' concerns. As well as presenting the weekly report to the CEO at CEO Meeting and sending weekly status reports to the CEO, the consulting supervisors, the examiner, and the company's employees.

Line & Process Manager – Daniel Ma

Ensures suitable working environments and sees that everyone feels involved and content. Handles formal communication with the company leadership. Develops and handles the company's processes along with the Quality coordinator.

Lead Analyst - Sofie Andersson

The main responsibility is to contact our customer and understand what they want from our system and what they need. Also responsible for making progress in the analyst work and communicating to the manager group about it.

The analyst team is responsible for defining requirements and user stories so that the development group knows what to do and has customer meetings.

Analysts - Frida Dolk & Johan Chou Chen

Analysts are responsible with the Lead analyst to structure the work with the customer and align the customer needs with the rest of the company. Shall understand customer's desires and put these into requirements in a concrete, detailed and organized way and deliver to the developer team.

Test Leader - Axel Telenius

Handling testing operations towards the customer, following up work by the test team, and coordinating with the quality coordinator to ensure software meets specifications from the testing point of view.

Testers - Oscar Löfgren & Erik Parö

Testers assist the test leader with the test, assist the test leader with coming up with a clear plan for the testing, and report the test results. They are responsible for creating relevant test documents and making sure to follow them.

Quality Coordinator - Erik Sköld

Measures and monitors the product quality and initiates necessary changes of product and process. Responsible for the explicated plan for quality assurance. Works together with the testing team on the software quality assurance plan. Reviews decisions on code conventions, test tools, and reporting. Collects means of quality work, verifies traceability, and makes sure the work fits together.

Configuration Manager – Albin Ambrosius

Ensures that all tools, software or hardware, are being utilized and progress is being tracked. This role will see that everyone is notified of changes in the tools or assets that we create. Handles the set-ups for new tools and communicates with both the development team and company leadership through R&D reports.

Architect – Jacob Karlén

Specifies and decides on high-level architecture, target environment, and components to be used. Ensures that functional and non-functional requirements are met and coordinates with other teams on technical matters. Responsible for the Architecture Notebook.

R&D Manager – Ludvig Sundqvist

Responsible for scheduling, sending out agendas and moderating R&D meetings and coordinating the work within the department.

UX Designer – Olivia Shamon

Specializes in setting targets and realizing the user experience of the system. Creates prototypes of the system for the developers.

Deployment Manager – Ludvig Sundqvist

Makes sure the product is made available to the customer and plan and prepare for continuous deployment. Works as a middleman between the testers and developers works closely with the architect and test leader. This role is the responsible manager for Docker, GitLab, and containers.

Technical Writer – Eric Schadewitz

Ensures that the output of the project is accessible to our customer. Produces instructions on how to use the system the way it was developed to be used. Also documents suggestions for further development.

Developers – Gustav Danielsson, Lucas Hultqvist, Simon Boman, Adam Lilja & Hannes Ekstam Ljusegren

Have the main responsibility of realizing requirements into functions of the software solution.

4.2 Cross-functional teams

The company is further divided into three cross-functional teams (hereinafter CFT), with members from both departments and with different roles. See appendix A for the team structures. The idea with the CFTs is to speed up development by ensuring that the right competence is present in each team and that a wide range of competencies is present. One key area of the project is the requirement specification. The company's analysts carry the expertise of this area, and thus a decision was made to have one analyst in each CFT so that this knowledge is dispersed throughout the whole company. The members and structures of the CFTs may change during the development phase (between iterations) depending on customer needs, course requirements, or any other reasonable reason.

For each iteration, each CFT shall have different responsibilities as planned by the managers. During the pre-study, each CFT shall develop a unique prototype to showcase to the customer. One prototype shall be chosen from the customer feedback, and this prototype shall be the basis for the development. In the first iteration, one team develops the software's back-end, and two teams focus on the front-end (software overview and patient journal, respectively). In the second iteration, the back-end team shall swap focus to front-end development if back-end development is finished.

To address slow development progress and isolated developers and testers (feedback received from, e.g., complaints, retrospectives, and company surveys), the CFTs are restructured for iterations three and four. During these iterations, the CFTs have the following focus areas: CFT1 shall be developer-heavy and focus on completing the issues with the highest rank; CFT2 shall focus on UX design, provide the development team with prototypes when needed and also test the implemented component to verify that they work in the intended way UX-wise; CFT3 shall focus on testing and quality, with tasks such as implementing a pipeline in GitLab with automated tests and accepting merge-requests.

5 Processes

In this section, the different processes of the company are described. This section acts as a guideline for company members on how to conduct specific tasks and allows for the course responsible (examiner, CEO, supervisors) to get an overview of the company's processes.

5.1 Communication and meeting processes

Meeting protocols will be created beforehand, and anyone can add to the topics to discuss at the meeting. Regarding the CEO meetings, a deadline to add main topics is set on Wednesdays at 12.00 in order to allow time for the project manager to create a PowerPoint presentation and

send the agenda to the CEO. Smaller topics can still be added after the deadline but will be brought up at the meeting only if time allows.

Meetings shall be scheduled in Teams using the calendar function, and both required and optional attendees shall be added to the meeting (exception for company-wide meetings). This way, individuals will have a better overview of meetings in place and which meetings they have to attend. Company-wide or large meetings (e.g., CFT meetings) shall be announced 24 hours beforehand. Exceptions can be made, e.g., for crisis meetings.

Managers have a weekly meeting each Monday to discuss different topics added to the agenda. These meetings also serve as a way to delegate tasks and make decisions.

Each Friday, the team leaders for each CFT will meet with the person responsible for the status reports to update them on progress. This achieves three things: first, the progress of each CFT can be documented in the status reports; secondly, the progress can be analyzed and compared with the set milestones to see if the teams are on track or if something needs to change in order to reach the milestones in time; and thirdly, it creates a tighter integration between the teams as the team leaders can sync up and discuss needs. These meetings are complemented by a shorter stand-up meeting each Tuesday (applicable in iteration 3 and 4).

Each CFT shall have at least two stand-up meetings per week, and it is up to each CFT to decide when these meetings shall take place. To start each iteration, a sprint planning meeting shall be held within each CFT. To end an iteration, each CFT leader shall plan and hold a retrospective to gain insights into how the teams have performed and what can be done better for the next iteration.

5.2 Development and GitLab processes

To track development and other tasks, boards and issues on GitLab are used. For example, every requirement is reflected as the issue on GitLab, which developers shall assign themselves to when working on implementing a requirement. One issue is to be created for each requirement – for the sake of traceability. Tasks are then created under each issue if needed (i.e., if the requirement was large enough). With this process, the company faces the problem of infrequent merges as some issues contain several sub-tasks. To address this problem, this process is updated: requirements shall be split into multiple issues instead, if needed, to create a better continuous integrative process. Traceability is achieved by a requirement id as a part of the issue name/description. When all parts are implemented, the original requirement issue is closed.

Issues/features shall be developed in individual branches, *feature branches*, which are then merged to the development branch. The issues shall then be closed, and the requirement shall next be reviewed. If the implementation passes the review and the testing phase, then the issue shall be remained closed, otherwise reopened. See appendix B for an example of a GitLab board used in the company. Furthermore, the issues shall be connected to a milestone, e.g., priority 1 requirements, in order to create a burn-down chart. To differentiate issues on GitLab, id's are used to specify the type:

- bg: issues related to bugs.
- dc: issues related to documents.
- fc: issues related to features not specified as a requirement.
- nuc: issues related to non-functional requirements.
- rc: issues related to functional requirements.
- te: issues related to testing and quality control.

The development process is further described in the Software Quality Assurance Plan, section 3, and in the README of the company GitLab.

Table 1: The table shows the regulatory documents of the project. Listed are also the author(s) of the document and the project member(s) who are the main reviewer(s) the documents.

Document name	Author	Reviewed by
Architecture notebook	Jacob Karlén	Developers
Customer requirements specification	Sofie Andersson	Analysts
Education plan	Axel Nilsson & Daniel Ma	Project manager, Quality coordinator
Project plan (this document)	Axel Nilsson & Somaye Gharedaghi	Managers
Software quality assurance plan	Erik Sköld	Process manager
Test plan	Axel Telenius	Testers
User manual	Eric Schadewitz	Deployment manager, Product manager

5.3 Regulatory documents

The documents listed in table 1 shall be seen as living documents which are continuously updated throughout the project.

Regulatory documents shall be written in LaTeX, preferably with the help of Overleaf. The documents are made available to everyone in the company through link sharing. These links can be found among the company's files in Teams. Each document shall be reviewed after it has been updated and documented in the version table on the document's first page. This review shall be done by a member with relevant expertise, e.g., a tester can review updates to the Test Plan, and a developer can review the Architecture notebook. The grammar of each document can be reviewed by any company member. Each document shall be uploaded to GitLab (including all Latex files) to version-control. Each document (PDF) shall be uploaded to the Teams folder "Output" for the course responsible (examiner, CEO, supervisors). There is a secondary folder called "Old versions" in the Output folder, where old versions shall be moved to whenever a new version of a document is uploaded in the Output folder.

To track when work is being done in the documents, there is a board on GitLab. When working on a document, an issue shall be created on GitLab and have a member assigned to it. When a document is ready to be reviewed, the issue shall be moved to the correct list of the board, under "Ready to be reviewed". These issues shall also be connected to a relevant milestone, e.g., "everything that shall be done until the end of the project". Additionally, issues not directly connected to requirements, such as issues connected to documents, shall be assigned to an epic on GitLab. Process for regulatory documents:

1. Create issue on GitLab, under list "Backlog" or "Doing", when planning to work on a document (author)
2. Change document by editing or adding information (author)
3. Move issue to the list, "Up for review", and assign reviewer to the issue (author)
4. Review and comment changes (reviewer)
5. Address comments and edit if necessary (author)
6. When a document has been reviewed, the new version shall be uploaded to Teams and GitLab (author)

5.4 Requirements

The work with the requirements shall be conducted by the analyst team. A first draft of the requirements is to be created in the first iteration, before the tollgate meeting. The requirements shall be reviewed by the analyst team during every iteration to ensure that they are updated according to both external feedback from customers and internal feedback from the company's different departments. After feedback from the internal and external sources, the analyst team shall review the feedback and change all requirements according to what is considered valid feedback both from the customer perspective and the company perspective. The feedback from

customers is collected at customer meetings, which are scheduled at least once per iteration. The feedback from the company shall be collected every week from each CFT at the weekly company meetings. This feedback can be related to re-prioritization due to time and difficulty, as well as improving language for clarity. Every requirement shall be conducted from and be traceable to a use case, which originates from customer meeting data. Each use case shall have a version history and a note of who the author is of that version. Every requirement shall have an ID that gives the requirement a unique identifying name. The ID shall also be connected to the use case. In all places where requirements are going to be used, the ID of the requirement shall be clearly noted.

When a requirement is changed after feedback, it shall first be discussed in the analyst team to decide if the change is of significant or minor importance. If the team decides that a change is of little importance and is needed, a note will be made in an internal changelog and changed in relevant places such as GitLab and SRS. Examples of minor changes are spelling and grammar mistakes.

If the requirement is of more significant importance and a more considerable change is required, the procedure is similar to small changes: a note is made in an internal changelog. Additionally, a requirement change is made that explains why a change was made, what the current situation for that requirement is, and who is responsible for that change. This is later communicated out to relevant departments and changed in internal documents.

All company members shall know where the requirements can be found and what they mean for the customer. Therefore, the analyst team shall not be the only company representatives at customer meetings and on-site visits. Instead, team members from each workgroup shall be present along with the analysts. The analyst team will also ensure this by clearly presenting the background and key takeaways from the customer meetings on the company meeting at the beginning of iteration 2.

5.5 Risk process

To identify risks within the project, risk identification workshops have been held to get a list of all relevant risks discovered. The risks were then assessed and given a probability factor from 1 to 4 and an impact factor of 1 to 4. Multiplying these two factors results in the risk management indicator, from which the risks are then ranked. A short description of how to handle the risk if it occurs and how the company should work to lower the probability of the risk occurring was then added to each risk.

At a later part of the project, a process to update the company employees on relevant risks and monitor risks was implemented. This process was to have a portion of each CEO meeting be reserved to go through the most relevant risks at the moment and go through the risks which have been updated with a new probability or impact factor. New risks discovered throughout the project were added using the process described above.

5.6 Change of processes

Changing of processes can be made and decided upon verbally between manager-level members. The new processes need to be documented in this document. If these changes concern specific members, these members need to be notified and preferably included in the discussion. Changes concerning the whole company shall be announced in the Teams channel, General.

6 Milestones and timeplan

Tables 2, 3, 4, 5, 6, and 7 indicate the milestones per iteration.

Table 2: Milestones For Pre-study

Date	Description
13/Sep, 2021	Set milestones for tollgate meeting
17/Sep, 2021	CFT2: Internal deadline for first draft for Minimalistic, well presented prototype for second meeting with the customer
20/Sep, 2021	Signing the Company Contract by staff CFT1: Internal deadline for first draft for Quantity in a compact format prototype for second meeting with the customer CFT3: Internal deadline for first draft for Dashboard style layout prototype for second meeting with the customer
21/Sep, 2021	Second meeting with the customer for feedback
22/Sep, 2021	Analysts internal deadline for requirements list
23/Sep, 2021	Tollgate Meeting

Table 3: Milestones For Iteration 1

Date	Description
24/Sep, 2021	Workshop with analyst team
27/Sep, 2021	Final decision on tools to assign requirements in backlog
28/Sep, 2021	Developer Workshop
30/Sep, 2021	Hospital visit in person (Motala) Final set up milestones for each CFT
01/Oct, 2021	Hospital visit in person (Linköping)
04/Oct, 2021	Company website Up Hospital visit in person (Norrköping)
08/Oct, 2021	Friday, October 8, 2021 CFT3 done header, patient overview etc.
09/Oct, 2021	Internal Deadline for Iteration 1
11/Oct, 2021	Sprint Planning for iteration 2 External Deadline for Iteration 1

Table 4: Milestones For Iteration 2

Date	Description
14/Oct, 2021	Feedback from the customer
15/Oct, 2021	CFT1: Finish all routes, services, information, testing CFT2: Finish overview, migration to GitLab CFT3: Finish UC11/12 (priority 1)
19-31/Oct, 2021	Exam Period
04/Nov, 2021	Update Output Document CFT1: Finish RoS-integration UC-010-002 and Contagious UC-014 in the patient journal CFT2: Finish in/out-flow (cards 3 and 4 in figma) UC-005 and UC-016 CFT3: Finish ecg/ekg UC-007 Internal Deadline for Iteration 2 (Finish priority 1 requirements) Feedback from the customer
07/Nov, 2021	External Deadline for Iteration 2 (Have a working product to show)

Table 5: Milestones for Iteration 3

Date	Description
08/Nov, 2021	Creating priority 2 milestone table
09/Nov, 2021	finish Automated testing and pipeline Finished with acceptance testing structure and tasks Finish new version of project plan SQ assessment Software Quality Assessment for the work until iteration2 Finish the priority 1's leftover from iteration 2
10/Nov, 2021	Make sure that all priority 1 are closed Start working on priority 2 tasks Soft deadline finishing reviewing priority 1 requirements Started prototyping for unfulfilled requirements
11/Nov, 2021	Translate SUS-test New version of requirements, non-func. requirements, tech. requirements and visual requirements
12/Nov, 2021	Fix boards so developers can simpler pick tasks Finished reviewing all priority 1 requirements Finish Selenium tests for everything in iteration 1 and 2
15/Nov, 2021	That every developer have a priority 2
16/Nov, 2021	Update the worksheet with the new requirements and keep all data about the reviews that have been done Look through the summarizing of the customer test result and add what is relevant to relevant issues and reviews Summarize results from customer meetings
17/Nov, 2021	List updated requirements (number of requirements for each priority) Finish at least 10 priority 2 issues Merge a stable develop to main
18/Nov, 2021	Update Output Document Internal Deadline for Iteration 3 (Finish priority 2 requirements) review all priority 1 and 2 issues that where done by 17/Nov Analyst Leader Schedule customer meeting Finish testing main branch
21/Nov, 2021	R&D manager (make sure everything is merged) External Deadline for Iteration 3(Have a working product to show)
22/Nov, 2021	Push stable main branch Testers Review tasks for acceptance testing
23/Nov, 2021	Acceptance testing with customer (eventual) Testers Finish test plan v2.0 for reviewing

Table 6: Milestones for Iteration 4

Date	Description
24/Nov, 2021	Testers Address feedback on test plan v2.0 (finished version) Acceptance testing with customer (eventual) Quality Coordinator Quality assessment on work done in previous iteration
25/Nov, 2021	Summarize data from acceptance testing (eventual) Finish all priority 2 requirements Postponed External Deadline for Iteration 3
26/Nov, 2021	Product owner, lead analyst (Eventual) Customer meeting addressing SRS Testers Report all inspect element bugs Feedback from Customer
29/Nov, 2021	Deployment issue should be done Quality Coordinator Software Quality Assessment for the work until iteration 3
30/Nov, 2021	Address review feedback and test feedback on priority 2 requirements Testers Finish test plan v3.0 for reviewing 140 hours for developers and finishing all coding
01/Dec, 2021	Major bug fixes Testers Test plan V2.1 finished Update Output Document
02/Dec, 2021	Comprehensive system review review all review-able requirements Testers Comprehensive system testing Quality Coordinator Publish Software Quality Assurance Plan v1.4 Internal Deadline for Iteration 4 (Finish priority 3 requirements)
03/Dec, 2021	Feedback from Customer Testers Comprehensive usability testing Send test plan for feedback from supervisor Code stop
05/Dec, 2021	Authors Finished regulatory documents + other documents needed in the output folder External Deadline for Iteration 4 (Have a working product to show)

Table 7: Milestones for Release

Date	Description
06/Dec, 2021	Comprehensive documents review Quality Coordinator Last version of Software Quality Assurance Plan ready to be reviewed Testers Push pipeline to main
07/Dec, 2021	Final system review Testers Final system testing Authors Documents fix Authors Documents stop and hand-in Quality Coordinator Software Quality Assessment for the work until iteration 4 Quality Coordinator Publish last version of Software Quality Assurance Plan Review , Final testing
08/Dec, 2021	Testers Final acceptance testing Final bug fixing
09/Dec, 2021	Testers Acceptance testing report CFT1: Address the last feedback from final acceptance testing Delivery and conclusion internal deadline
14/Dec, 2021	VSSE rehearsal
16 /Dec, 2021	VSSE

7 Risks

Table 8 below shows the risks we have identified for the project. They have been ranked on how significant the impact of the risk occurring would be and how likely the risk could happen. This table has been updated over the project by finding and adding new risks as well as changing the indicator according to changes of probability and impact over time, and the history has been recorded in Table 9.

Table 8: Identified Risks

ID	Risk	Probability	Impact	Risk Management Indicator (Probability e times Impact)	Risk Measure
6	Wrong time estimation	4	4	16	Constant communication and checking progress against schedule
9	Not meeting the external deadline	4	4	16	Transparent communication with the customer
20	Delay in deployment which affects acceptance testing	4	4	16	Assign someone to help the one who works on it
4	Not having buffer times and internal deadlines	3	4	12	Make sure to have internal deadlines a couple of days before the iteration ends
12	CFT dependencies	3	3	9	Communication between teams, weekly team leader meetings to update each other
1	Loss of code due to GitLab issues	2	4	8	Commit often so you can revert back if problems occur, ask if unsure when merging
11	Not reaching 160 hours	2	4	8	Keep tracking the time-sheet to make involve members who cannot reached the 160h in priority 4 tasks.
18	Having a lot of failed reviews and needing to spend reworking and reprogramming issues, leading fail for deadlines	2	4	8	Being stricter internal deadlines so we have a buffer to the customer deadlines
21	few comments per function and a lot of commit messages not following the company guidelines which makes it difficult to maintain and understand the code	4	2	8	Add comments and follow the company guidelines
3	Staff sickness	3	2	6	Documentation for trace-ability
17	manage to complete all the top priority issues, 1 and 2	1	4	4	1 issue per day or 5 per hour
13	Not having enough time to handle responsibilities	2	2	4	Splitting up the responsibilities using other staff with lower burden
2	Lack of knowledge about Angular within CFTs	1	4	4	Have a plan for educating ourselves in development, assess competence regularly
8	Misunderstanding in requirements	1	4	4	Getting feedback from the customer regularly
15	Trouble with merging	1	4	4	More resources, such as time
19	Trouble with deploying the product	1	4	4	More resources, such as time
7	Customer's new request	1	3	3	Getting feedback from the customer regularly
14	Long lasting branches that exist for a long time	2	1	2	terminate them when configuration manager decides
10	Losing critical members (ex: somebody drop the course)	1	1	1	CFTs and Documentation
5	Technical problems (computer stops working)	1	1	1	Use computer in computer lab if your computer stops working, save often
16	Zombie code (code that is old and not in use but still are on GitLab)	1	1	1	check compare old branch with new branch when merging, check affected files to prevent this

Table 9: Risks History

Date	Description
09/Nov, 2021	<p>Risk 1: Changed probability from 4 to 2 due to implementation of pipeline and better merge request reviews</p> <p>Risk 2: Probability changed to 1 from 4 since developers have self-studied and a workshop has been held</p> <p>Risk 3: changed impact to 2 from 4 since we work in teams where people can take over responsibilities when someone is sick.</p> <p>Risk 4: lower the probability since we have implemented buffer in the time plan</p> <p>Risk 7: Lowered impact to 3 from 4 since a change from the customer is not going to affect the company too much.</p> <p>Risk 10: The company could adapt if someone quit since we know what people are responsible for.</p> <p>Added risk 11, 12, 13</p>
11/Nov, 2021	<p>Risk 6: Higher probability 2 to 4 since we could not meet the external deadline in iteration3</p> <p>Risk 4: Higher probability 1 to 3 since the remained time is tight.</p> <p>Risk 9: Higher probability 1 to 4 since we are not meeting the external deadlines.</p> <p>Risk 11: Higher probability 1 to 2 since the developer have a short time to cover the 160 hrs.</p> <p>Added risk 14, 15,16,17,18</p>
15/Nov, 2021	<p>Risk 15: Split up into 2 different risks. The first part kept in risk 15 and the other part is added as risk 19.</p> <p>Risk 14: Probability changed from 4 to 2.</p> <p>Risk 17: Probability changed from 4 to 1.</p>
02/Dec, 2021	<p>Risk 10: impact changed from 3 to 1</p> <p>Added risk 20 and 21</p>
04/Dec, 2021	<p>Risk 6: Probability changed from 4 to 1</p> <p>Risk 9: Probability changed from 4 to 1</p> <p>Risk 11: Probability changed from 2 to 1</p> <p>Risk 13: Probability changed from 2 to 1</p>

A Cross-functional teams

	CFT1		CFT2		CFT3	
<i>Team leader</i>	Axel Telenius	Test leader	Erik Parö	Tester	Ludvig Sundqvist	Deployment manager
	Daniel Ma	Line manager	Albin Ambrosius	Config manager	Adam Lilja	Developer
	Eric Schadewitz	Technical writer	Frida Dolk	Analyst/UX	Axel Nilsson	Product manager
	Erik Sköld	Quality coordinator	Gustav Danielsson	Developer	Johan Chou Chen	Analyst
	Jacob Karlén	Architect	Hannes Ekstam Ljusegren	Developer	Lucas Hultqvist	Developer
	Simon Boman	Developer	Sofie Andersson	Lead analyst	Olivia Shamon	UX Designer
					Oscar Löfgren	Tester

Figure 3: The cross-functional teams in iteration 1 and 2.

Team leaders in bold	CFT1		CFT2		CFT3	
	Development focused		UX focused		Testing and quality focused	
	Jacob K	Architect	Olivia S	UX lead	Axel T	Test leader
	Lucas H	Developer	Albin A	Config. manager	Oscar L	Tester
	Simon B	Developer	Frida D	Analyst/UX des.	Erik P	Tester
	Adam L	Developer	Eric S	Technical writer	Erik S	SQA coordinator
	Hannes E	Developer	Axel N	Product manager	Johan	analyst
	Sofie	L Analyst	Somaye G	Project manager	Daniel M	Line manager
	Ludvig S	Depl. manager				
	Gustav D	Developer				
	Will focus on realizing the requirements into features of the software solution. Developer-heavy, but have an analyst to clarify what the true customer needs are		Focuses on updating the prototype with views for components to be implemented (so CFT1 has something to base component implementation on). Reviews components produced by CFT1 and gives feedback and actions to take to improve UX.		Testing and quality assurance, along with input from lead analyst to make sure that customer needs and wants are met.	

Figure 4: The cross-functional teams in iteration 3 and 4.

B Development process

<p>CFT1 Backlog</p> <p>All future events scheduled or planned shall be visualized as data points on the timeline</p> <p>#36</p> <p>The timeline shall clearly visualize where the current position on the timeline is (i.e. what time it is right now)</p> <p>#37</p> <p>The application shall close the box with specific information by either clicking a cross on the box or somewhere on the screen that is not the box</p> <p>#42</p> <p>On the human body silhouette, all the in/out-farter currently on the patient should be visible and placed on the body part that they are currently put on</p> <p>#122</p>	<p>Prio 2 Backlog</p> <p>Whenever the user is not exposed to the full timeline, the timeline shall show an arrow that point to the left and says "Det finns fler händelser (länge bak i tiden)".</p> <p>#39</p> <p>The application shall be able to show more specific information about each event by user tapping on the icon. (E.g. user who was responsible for the event, all relevant lab results or "utlåtanden" from doctors)</p> <p>#41</p> <p>The timeline shall have a feature that enables scrolling back or forth on the timeline by swiping the user's finger to the right or left</p> <p>#43</p> <p>The application shall show the "orsakskod" as plain text and not the numerical code Example: Kontakt med dokt, luv</p> <p>#48</p>	<p>Selected issues</p> <p>The menu shall be able to hide/show by clicking on the menu button in the header</p> <p>#22</p> <p>There shall be a patient overview page - "Enhetsöversikt" that shall show all the patients currently in the ER in a list</p> <p>#25</p> <p>The application shall show the patients triage color on the team's overview</p> <p>#23</p> <p>The application shall have a header</p> <p>CFT1 Done</p> <p>#21</p> <p>The header shall contain Region Östergötland-icon, button for menu, search field, button for "all patients", "Start view", date and time, message icon, button for the logged in user</p>	<p>CFT1 Story</p> <p>On the patient view, there shall be a widget displaying a silhouette by a human body</p> <p>#123</p> <p>The widget should request data about "in-ut-farter" from backend (based on patients "personnummer")</p> <p>#127</p> <p>On the widget, there shall be a button called "Visa historik"</p> <p>#124</p> <p>When a user clicks "Visa historik" the widget should display a list of all "in/ut-farter" that the patient has had during the ER visit as well as start time and removal time</p> <p>#125</p> <p>When a user clicks "Släng" on the display list of historical data, the list will disappear and the user will be back on looking at the widget</p>	<p>CFT1 Done</p>
--	---	--	--	-------------------------

Figure 5: GitLab Board