SDP - Software Development Plan

Team 1 - ETSN05

Revision History

Date	Version	Description	Author
2021-09-14	1.0	First draft and outlines of all sections	Linnea Johnsson
2021-09-25	1.1	Updated according to review comments	Linnea Johnsson
2021-09-26	1.2	Updated time allocation and Gantt chars	Linnea Johnsson
2021-09-27	1.3	Uupdated risks based on team feedback	Linnea Johnsson & David Ravanelli

Referenced Documents

1. Programvaruutveckling för Stora System - Projekthandledning 2021 (PH)
2. Product Requirements Document - PUSS21412 (PRD)
3. Software Product Features Backlog(SPF) - PUSS21413
4. Sprint log/plan (SP) - PUSS21414
5. Software Top Level Design Document (STLDD) - PUSS21415
6. Software Detailed Design Document (SDD) - PUSS21416
7. Software Verification and Validation Report (SVVR) - PUSS21417
8. System Specification Document (SSD) - PUSS21418
9. Project Final Report (PFR) - PUSS21419

Contents

1 Introduction

This Software Development Plan (SDP) details the development plan produced for the development of the web-app based carpool service called ETSN05_PG1 Carpool Webb-app (ECW). The description of this product can be found in the PH document chapter 9. Furthermore, the SDP derives mostly from the PH and includes the process model with the project's different phases and their corresponding time-estimates and major events, and the tailoring of the project. The SDP also includes a description of the project's organisation, time plan, configuration handling, and quality review, as well as a a risk analysis of the project.

2 Project Organisation

The project is organized into a Project Group (PG), Developing Group (DG), and a System Group (SG). The PG consists of the Project Owners (PO) and Scrum Master(SM), the DG consists of developers and the architects, and the SG consists only of the architects. Within the DG there are different areas of responsibilities. There are mainly four areas of responsibilities that corresponds to three different groups, and one sole role. There is the Algorithms Group (AG), the Frontend Group (FG), and the Backend Group(BG). There is also one role responsible for testing (TR).

Besides the above mentioned internal actors the project has an External Group (EG), which include a section chief, a customer, a reviewer, and three different experts roles. The experts have different areas of expertise. There is one requirement expert, one test expert, and one design expert.

2.1 Project team members

The project's team member and their corresponding roles can be seen in table 1. All internal team members are enrolled in the course ETSN05 and participating in the project for educational purposes. The members have been given their different groups and roles based on interests and experience. The external team members are educational staff members in the course.

Group	Name	Role
PG	Alexander Ekman	PO
	Linnea Johnsson	PO
	David Ravanelli	SM
AG	Max Fogwall	Architect & TR
	Emil Friberg	Developer
	Gustav Broman	Developer
FG	Erik Kullberg	Architect
	Axel Domell	Developer
	Erik Busk	Developer
	Johannes Bastmark	Developer
BG	Anton Håkansson	Developer
	Elsa Lindhé	Developer
	Eric Schyllert	Developer
EG	Alma Orucevic-Alagic	Customer, Chief & Reviewer
	Rasmus Ros	Design expert
	Sergio Rico	Requirements & Test Expert

Table 1: The ECW project's team members

3 Process model

The process model mostly follow the directions from the PH chapter 2. The things that differ, and/or need more explanation, are described below.

3.1 Phases

The project is divided into four phases, where the last three are part of sprints. All phases include milestones in the form of documents to produce, deadlines to meet and meeting to have.

The meetings that are held in the different phases include the Sprint Planning Meeting (SPM), the Daily Stand-up Meetings (DSM), the Reflection Meeting(RefM), the Review Meetings(RevM), Internal Project Meetings(IPM), and External Project Meetings (EPM). The latter refer to meetings between the project's internal groups having meetings with the project's external actors. The person responsible for the different meetings is presented in table 2. Being responsible for a meeting includes calling it and assigning someone or oneself as responsible of note taking and action list creation. One exception is RevMs, where by default the PG has this responsibility. Another exception is in EPMs, where someone of the internal actors in order of first PG, then SG and lastly DG has this responsibility.

RefMs and SPMs will be held in connection to either Mondays ' or Fridays' DSMs in order to ensure high attendance. These weekdays' DSM are held over Zoom in order to better connect to the team and more easily raise issues or concerns. The rest of the DSMs are held on a designated Slack channel in order to accommodate the team members conflicting schedules.

The sprint phases requires time for development of the ECW, but all phases includes some level of mandatory coursework that require time from the project members. The time allocation will be further explained in section Time Plan.

Meeting	Head responsible
SPM	PG
DSM	SM
RefM	SM
RevM	Reviewer & PG
IPM	PG, DG or SG
EPM	Anybody

Table 2: The different meetings that are held within the project and role that is head responsible for the meetings

3.1.1 Phase 1

In table 3 the major events from the first phase are listed in order to create an easily accessible overview. This phase focuses on specification.

Event	Deadline	Responsible/Attending
SDP	2021-09-15	PO/-
PRD	2021-09-15	PO/-
SPM1	2021-09-10	PG/All
SPF1	2021-09-10	PO/-
SP1	2021-09-10	SG & DG/-
RevM 1	2021-09-17	PG/All

Table 3: The major events occurring in phase 1

3.1.2 Phase 2

Table 4 shows the seconds phase's main activities. This phase focuses on the first sprint and high-level design.

Event	Deadline	Responsible/Attending
Sprint 1	2021-09-26	DG & SG/-
RefM	2021-09-24	SM/All
DSM	-	SM/ SG & DG
STLD	2021-09-29	SG/-
SPM2	2021-09-24	PG/All
SPF2	2021-09-14	PO/-
SP2	2021-09-14	SG & DG/-
RevM 2	2021-10-01	PG/All

Table 4: The major events occurring in phase 2

3.1.3 Phase 3

The thirds phase's major event are included in table 5. The third phase focuses on sprint 2 and low-level design.

Event	Deadline	Responsible/Attending
Sprint 2	2021-10-10	DG & SG/-
RefM	2021-10-08	SM/All
DSM	-	SM/ SG & DG
SDD	2021-10-13	DG/-
SPM3	2021-10-08	PG/All
SPF3	2021-10-08	PO/-
SP3	2021-10-08	SG & DG/-
RevM 3 (informal)	2021-10-15	PG/All

Table 5: The major events occurring in phase 3

3.1.4 Phase 4

The last phase's major events can be viewed in table 6, including and ending with the final review. The last phase focuses on sprint 3, integration, system testing as well as checking, and producing, the last documentation.

Event	Deadline	Responsible/Attending
Sprint 3	2021-10-22	DG & SG/-
RefM	2021-10-22	SM/All
DSM	-	SM/ SG & DG
SVVR	2021-10-22	SG & DG/-
SSD	2021-10-22	PO/-
PFR	2021-10-22	PO/-
RevM final	2021-10-22	PG/All

Table 6: The major events occurring in phase 4

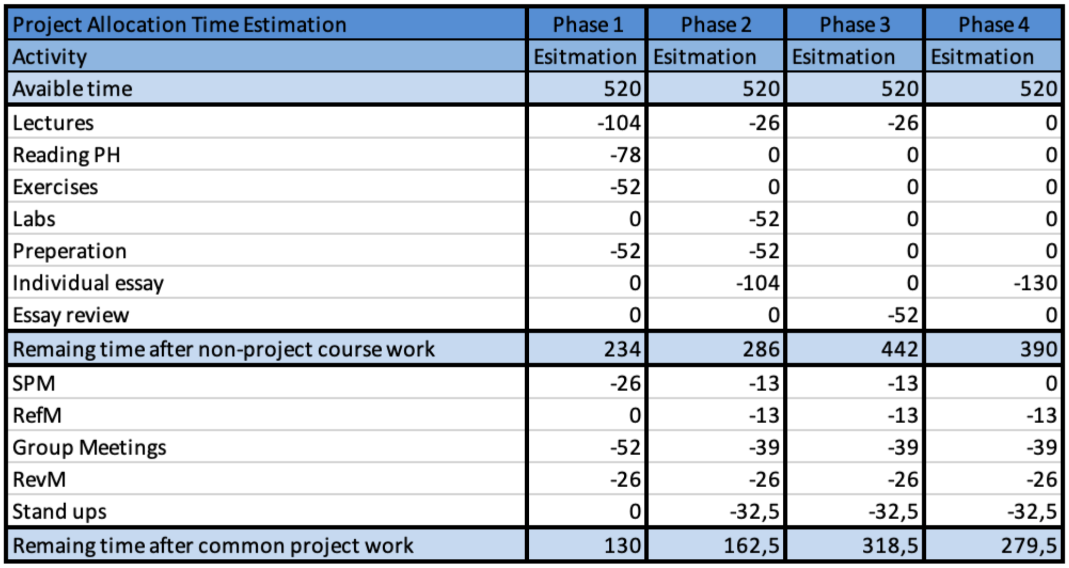
4 Configuration Handling

This project consists of several documents of different character, varying from source code to text documents such as this. There are two main libraries for the different types of configuration units. The first one is GitLab that contains the project's source code, the backlog, the sprint logs, and the SG's & DG's drafts of the STLDD, SDD, SVVR, & SSD. The main text documents such as the SDP, PRD, and PFR are stored in a GitHub repository in phase one and two, with the hope that we can integrate these to GitLab as well.

The configuration units are numbered according to the PH, and a list of the different document numbers can be seen in section Referenced Documents. Baseline versions are created prior to review hand ins.

5 Time plan

The project will run in parallel with other mandatory coursework. Below follows an estimation of the time allocation for the project, as well as Gantt charts for the main activities and the development process.



Project Allocation Time Estimation	Phase 1	Phase 2	Phase 3	Phase 4
Activity	Esitmation	Esitmation	Esitmation	Esitmation
Avaible time	520	520	520	520
Lectures	-104	-26	-26	0
Reading PH	-78	0	0	0
Exercises	-52	0	0	0
Labs	0	-52	0	0
Preperation	-52	-52	0	0
Individual essay	0	-104	0	-130
Essay review	0	0	-52	0
Remaing time after non-project course work	234	286	442	390
SPM	-26	-13	-13	0
RefM	0	-13	-13	-13
Group Meetings	-52	-39	-39	-39
RevM	-26	-26	-26	-26
Stand ups	0	-32,5	-32,5	-32,5
Remaing time after common project work	130	162,5	318,5	279,5

Figure 1: Time allocation estimation for the major course- and project events that are mutual for all internal team members. The presented available time is per the entire internal team, i.e. for 13 team members.

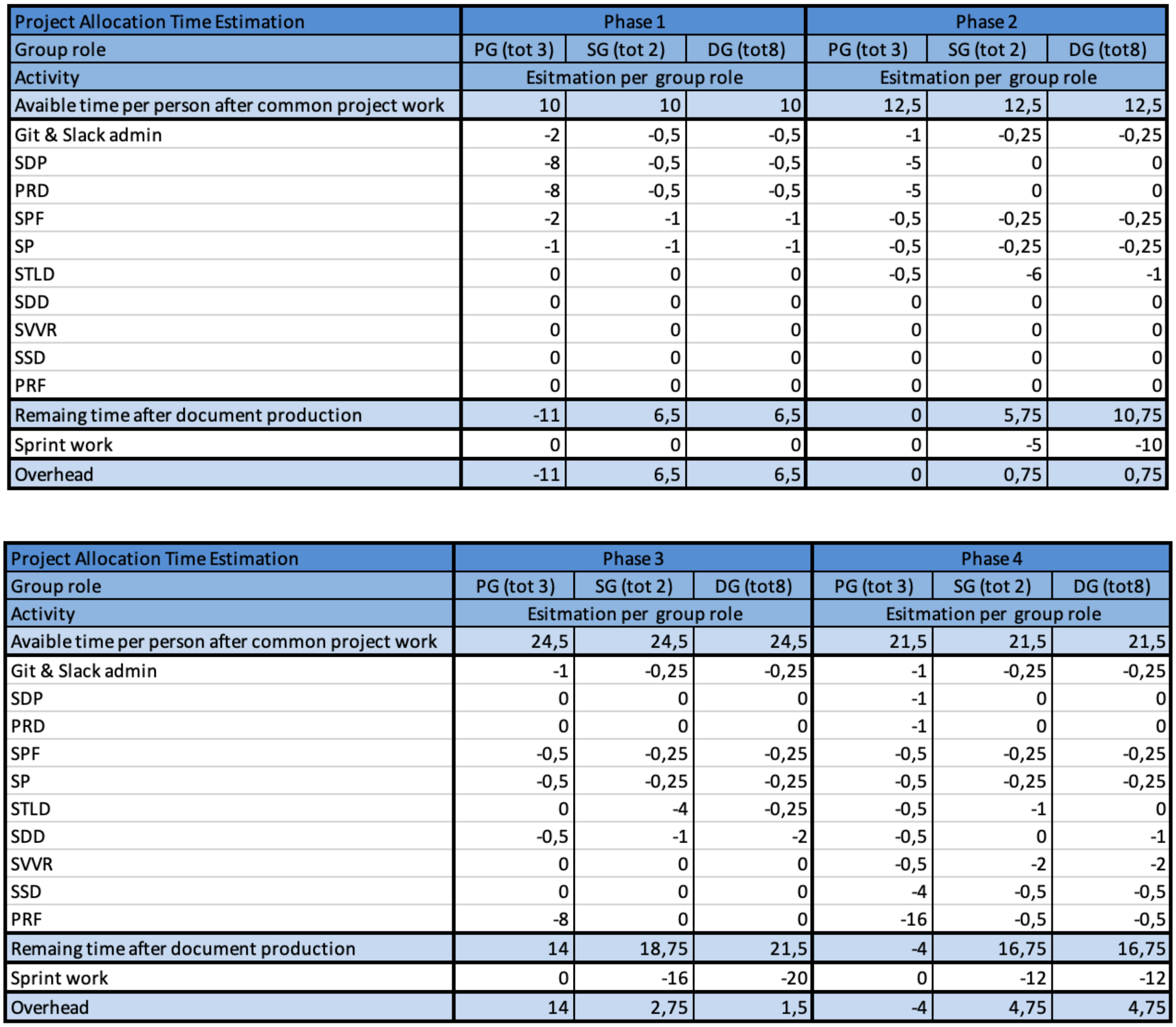
5.1 Time Allocation Estimation

In figure 1 the major course- and project events that are common for the entire project team, with their estimated time allocation needed for them, are presented. The time allocation estimation for the role specific project events are found in figure 2. The estimation in figure 1 takes into account that each team member has 20 hours per week for the course and that the different phases have a different amount of mandatory course work. The result from figure 1 is then divided by 13 in figure 2 in order to represent the remaining time available to allocate per person. This time is to be used in their group role specific tasks.

The DG is divided into different subgroups, but will work with a hands-on approach. If the FG and BG has a lot to do while the AG does not, the AG members help out where needed. E.g. in the first sprint more focus is put on the front end issues since the AG and BG would need more time to plan and draw out their implementations. When the planning is done, they are to help out with available issues or create new one if needed.

From figure 2 it can be extracted that phase three have the most time available for the sprints. A reason for this is that phase one and two has a greater focus on common coursework and on getting started with the project, i.e. figuring out how to correctly formulate and estimate issues and more. While phase four includes more documents needing to be produced, and loose strings needing to be tied up in order to deliver a working system.

An overhead has also been included, since allocating every possible unit of time does not allow for the flexibility needed in the agile way of working. For the PG this overhead is used to compensate for an uneven workload over the different phases. The PG have more to do in the beginning and end of the project, and a little less in the middle.



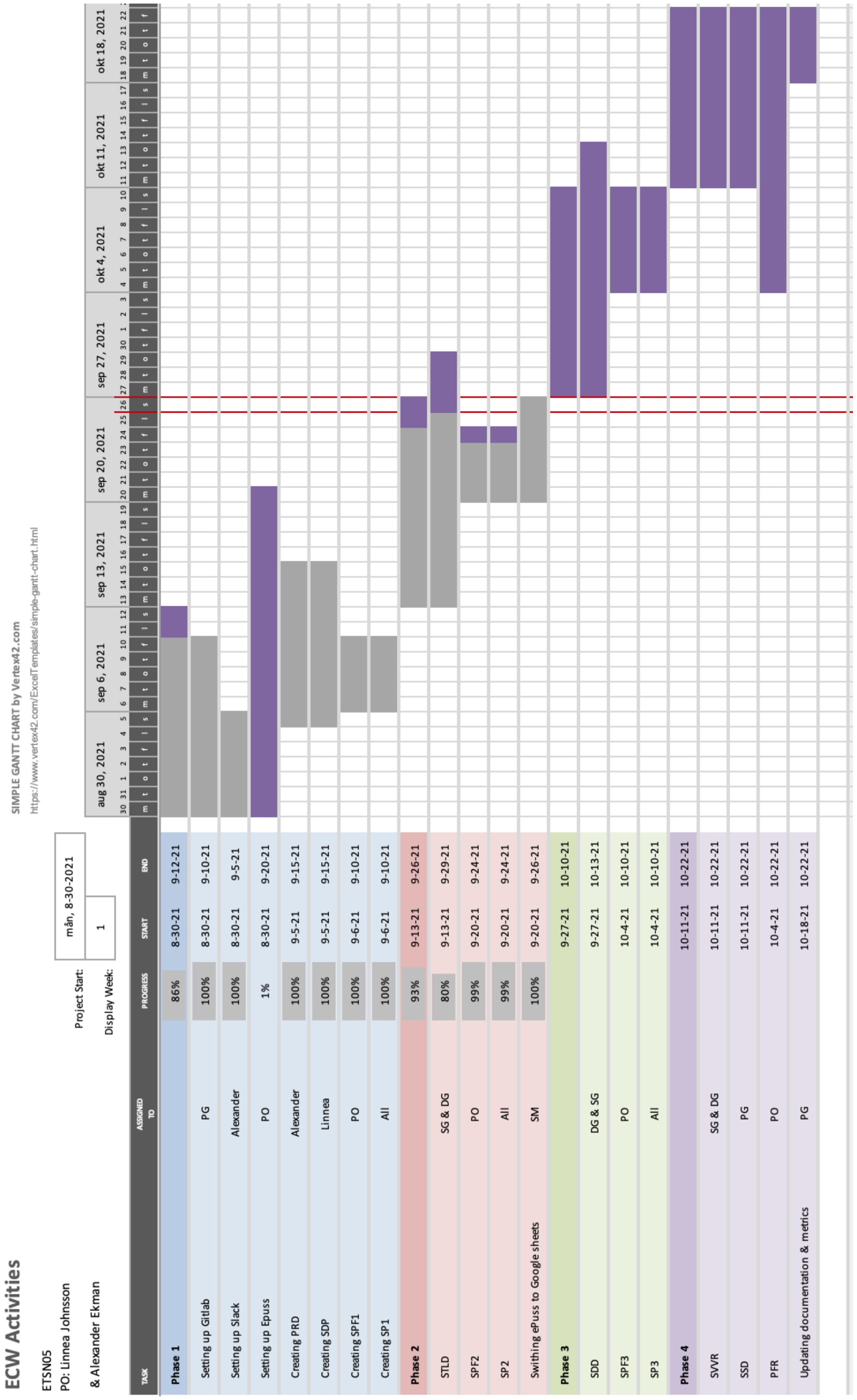
Project Allocation Time Estimation	Phase 1			Phase 2		
Group role	PG (tot 3)	SG (tot 2)	DG (tot8)	PG (tot 3)	SG (tot 2)	DG (tot8)
Activity	Esitmation per group role			Esitmation per group role		
Avaible time per person after common project work	10	10	10	12,5	12,5	12,5
Git & Slack admin	-2	-0,5	-0,5	-1	-0,25	-0,25
SDP	-8	-0,5	-0,5	-5	0	0
PRD	-8	-0,5	-0,5	-5	0	0
SPF	-2	-1	-1	-0,5	-0,25	-0,25
SP	-1	-1	-1	-0,5	-0,25	-0,25
STLD	0	0	0	-0,5	-6	-1
SDD	0	0	0	0	0	0
SVVR	0	0	0	0	0	0
SSD	0	0	0	0	0	0
PRF	0	0	0	0	0	0
Remaing time after document production	-11	6,5	6,5	0	5,75	10,75
Sprint work	0	0	0	0	-5	-10
Overhead	-11	6,5	6,5	0	0,75	0,75

Project Allocation Time Estimation	Phase 3			Phase 4		
Group role	PG (tot 3)	SG (tot 2)	DG (tot8)	PG (tot 3)	SG (tot 2)	DG (tot8)
Activity	Esitmation per group role			Esitmation per group role		
Avaible time per person after common project work	24,5	24,5	24,5	21,5	21,5	21,5
Git & Slack admin	-1	-0,25	-0,25	-1	-0,25	-0,25
SDP	0	0	0	-1	0	0
PRD	0	0	0	-1	0	0
SPF	-0,5	-0,25	-0,25	-0,5	-0,25	-0,25
SP	-0,5	-0,25	-0,25	-0,5	-0,25	-0,25
STLD	0	-4	-0,25	-0,5	-1	0
SDD	-0,5	-1	-2	-0,5	0	-1
SVVR	0	0	0	-0,5	-2	-2
SSD	0	0	0	-4	-0,5	-0,5
PRF	-8	0	0	-16	-0,5	-0,5
Remaing time after document production	14	18,75	21,5	-4	16,75	16,75
Sprint work	0	-16	-20	0	-12	-12
Overhead	14	2,75	1,5	-4	4,75	4,75

Figure 2: Time allocation estimation for the major project events per group role and phase. The starting value of available time comes the last row in figure 1 per phase divided by 13 team members

5.2 Gantt Activity chart

The planning of the main activities of the project in relation to each other, excluding the development of the product, is shown in figure 3's Gantt chart.



ECW Activities

ETSN05
PO: Linnea Johnsson
& Alexander Ekman

Project Start: mån, 8-30-2021
Display Week: 1

SIMPLE GANTT CHART by Vertex42.com
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

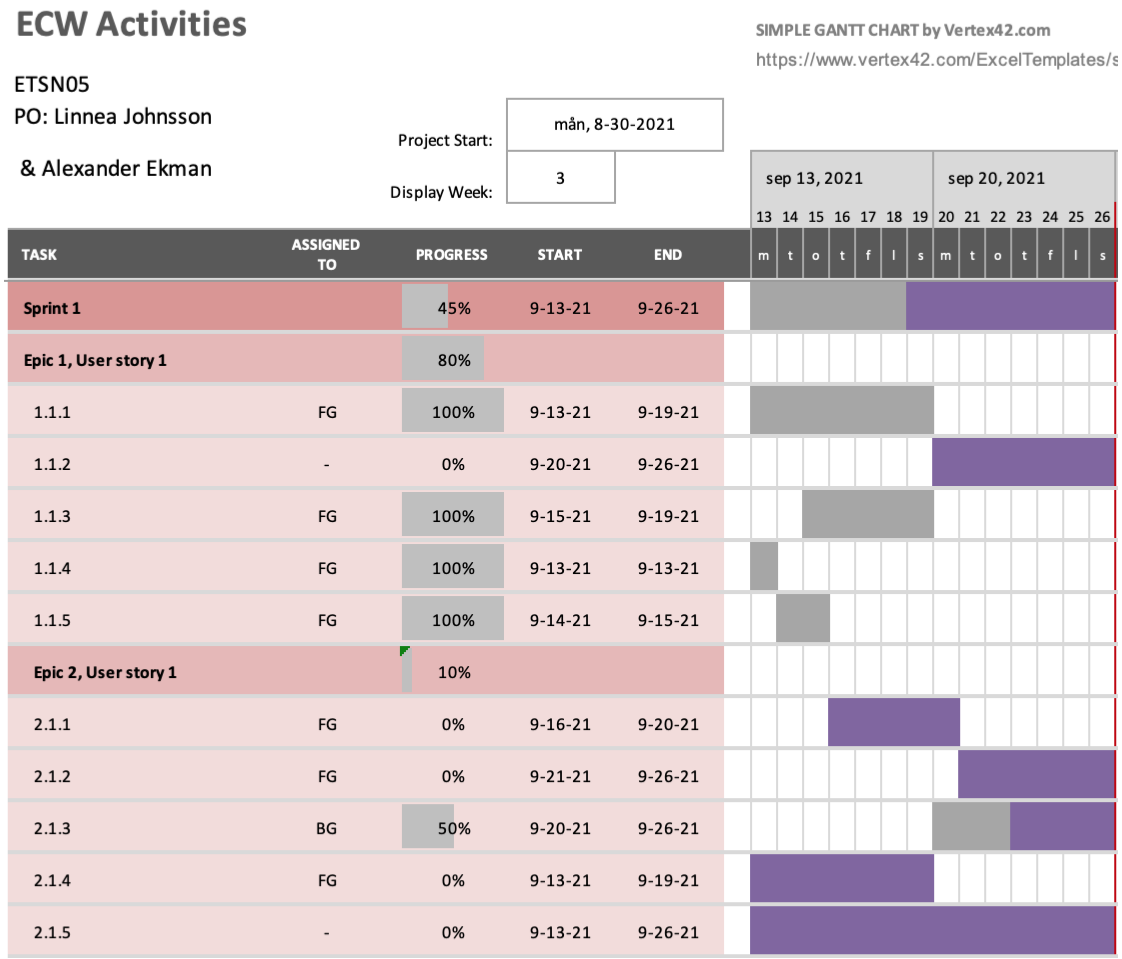
TASK	ASSIGNED TO	PROGRESS	START	END
Phase 1		86%	8-30-21	9-12-21
Setting up Gitlab	PG	100%	8-30-21	9-10-21
Setting up Slack	Alexander	100%	8-30-21	9-5-21
Setting up Epuss	PO	1%	8-30-21	9-20-21
Creating PRD	Alexander	100%	9-5-21	9-15-21
Creating SDP	Linnea	100%	9-5-21	9-15-21
Creating SPF1	PO	100%	9-6-21	9-10-21
Creating SP1	All	100%	9-6-21	9-10-21
Phase 2		93%	9-13-21	9-26-21
STLD	SG & DG	80%	9-13-21	9-29-21
SPF2	PO	99%	9-20-21	9-24-21
SP2	All	99%	9-20-21	9-24-21
Swithing ePuss to Google sheets	SM	100%	9-20-21	9-26-21
Phase 3			9-27-21	10-10-21
SDD	DG & SG		9-27-21	10-13-21
SPF3	PO		10-4-21	10-10-21
SP3	All		10-4-21	10-10-21
Phase 4			10-11-21	10-22-21
SVVR	SG & DG		10-11-21	10-22-21
SSD	PG		10-11-21	10-22-21
PFR	PO		10-4-21	10-22-21
Updating documentation & metrics	PG		10-18-21	10-22-21

Figure 3: A time plan represented in a Gantt chart over the main deliverable of the project, besides the code base development.

5.3 Gantt Development chart

Gantt charts are also used for the sprint planning. They are used to figure out which issues to start with, and which issues can be carried out in parallel, and which need to be developed sequentially during each sprint. This means that a new Gantt chart is developed for each sprint after the corresponding SPM based on the SFP and SP.

This decision was made due to the time allocation needed in order to produce a Gantt chart over all issues over all three sprints. Especially when taking into account the time needed to update the Gantt charts to include changes in future plans and the SPF due to missed issues or such when laying out the first plan. Having a detailed plan over issues too far into the future would need a compromise on the agile aspect of the project. The Gantt chart for sprint 1 is presented in figure 4.



ECW Activities

SIMPLE GANTT CHART by Vertex42.com
https://www.vertex42.com/ExcelTemplates/s

ETSN05
PO: Linnea Johnsson
& Alexander Ekman

Project Start: mån, 8-30-2021
Display Week: 3

sep 13, 2021 | sep 20, 2021

13 14 15 16 17 18 19 20 21 22 23 24 25 26
m t o t f l s m t o t f l s

TASK	ASSIGNED TO	PROGRESS	START	END
Sprint 1		45%	9-13-21	9-26-21
Epic 1, User story 1		80%		
1.1.1	FG	100%	9-13-21	9-19-21
1.1.2	-	0%	9-20-21	9-26-21
1.1.3	FG	100%	9-15-21	9-19-21
1.1.4	FG	100%	9-13-21	9-13-21
1.1.5	FG	100%	9-14-21	9-15-21
Epic 2, User story 1		10%		
2.1.1	FG	0%	9-16-21	9-20-21
2.1.2	FG	0%	9-21-21	9-26-21
2.1.3	BG	50%	9-20-21	9-26-21
2.1.4	FG	0%	9-13-21	9-19-21
2.1.5	-	0%	9-13-21	9-26-21

Figure 4: A Gantt chart over SP1 as a time plan for development in sprint 1

6 Time plan follow-up and quality review

The idea at first was to use the course recommended software EPUSS for time reporting. This plan changed due to information that the software not meeting school requirements. Time reporting will instead be done using a reprogrammed google sheets document where each team member has their own sheet to put in their time spend on different tasks.

The reported time is tagged with the codes given in the PH section 8, these include codes for working on documentation and meetings. The DG and SG will also use the git command spend when working on issues in order to track their progress. Both of these above mentioned ways of reporting time will then be used to follow up on the time plan and to measure how much we had over- and underestimated.

The follow-up of the time plan is carried out in two more ways. One is measuring the number of pushed up and/or added issues. Pushed up issues refers to issues being moved from one sprint to a later one, either due to time constraints or dependency miss calculations. Added issues instead refers to a user story having more issues added to it after the SPM due to a failure to include sufficient issues to cover the story. The latter can also include one issue being split up into more issues in order to better represent its work load. Another way to follow up on the time plan is by using a burn down chart and a Gantt chart. These are used in order to easily see the actual progress or time spent in contrast to the plan for each sprint.

The overall quality of the project is measured in two main ways. The first is measuring how well we estimate and plan the different parts of the projects, i.e. how well we follow the time plan. Secondly, code and deliverables are reviewed in many steps. One of the reviews are made by the SG. This include reviewing developed code before accepting merge requests into the main branch. This is done in order to ensure that the code developed follow the architectural plan of the project. Other, more informal reviews are done continuously both on the code and deliverables by IPMs and EXPs where cooperation and shared knowledge increases the quality. Furthermore, the RevMs provides the team with feedback on the deliverables so that they can be updated and improved before the final hand in, and the final review.

Finally, the quality of the delivered system is validated and documented in the SSD and SVVR that are produced and handed in during the last phase of the project.

7 Risk analysis

The risk analysis presented below was created by using input from the DG and SG. The members gave input from their area of expertise. I.e. they tried to formulate risks based on issues and difficulties they had run into in prior projects.

7.1 R1

Description: Issues being underestimated
Likelihood: 4/5
Severity: 2/5
Indication: More time is spent on issues than planned
Prevention: Basing estimates on experience and not overcrowding the schedule. Split big issues into smaller sub-issues, this makes it easier to estimate each sub-issue correctly. Analyze on sprint retrospective why we overestimated last sprint and avoid to make the same mistake for the upcoming sprint.
Damage Control: Move less prioritized issues to the a later sprint or bring in more developers in order to have more time to allocate

7.2 R2

Description: Issues being overestimated
Likelihood: 2/5
Severity: 3/5
Indication: Issues are getting done faster than planed and developers being unnecessarily non allocated
Prevention: Making educated guesses for the estimates and have an updated and prioritized SPF on hand.
Damage Control: Move up more issues from SPF to SP, ask developers if there are bugs needing to be fixed or code to be refactor and allocate the resource to that

7.3 R3

Description: Team members getting sick
Likelihood: 4/5

Severity: 2/5
Indication: No good indications, since sickness and other not-planed interference can occur from one day to another, especially in pandemic times.
Prevention: Utilize the digital tools for meetings instead of being in crowded rooms and follow the health departments recommendations
Damage Control: Re-prioritize the SPF and see what can be moved from the current sprint to the next. Use the Scrum Master to help re-organize the work that will not be finished.

7.4 R4

Description: Sever bugs appear in phase three
Likelihood: 2/5
Severity: 4/5
Indication: Little time spent on code review and low-coverage test written, the testing written do not acting as suspected
Prevention: Thorough code reviews and high coverage testings
Damage Control: Cut out affected functionality in order to deliver a functioning system to the final review

7.5 R5

Description: Sever bugs not appearing before hand-in
Likelihood: 1/5
Severity: 5/5
Indication: Little time spent on code review and low-coverage test written
Prevention: Thorough code reviews and high coverage testings
Damage Control: Ask the customer for an extension to bug-fix

7.6 R6

Description: Misunderstanding of specifications
Likelihood: 3/5
Severity: 4/5
Indication: Poor communication with client, long times in between feedback, and ambiguous/vague requirements
Prevention: Ask early-on for clarification, proactively reach out if we find flaws or holes in the current requirements, and/or prototype a solution in the early deliverables and emphasize that we need the customer's feed back on it. Use stand-up meetings to identify any misunderstandings.
Damage Control: Schedule time to refactor current designs, documents, code, and test cases to match the correct specifications while pushing back the development of new functionality.

7.7 R7

Description: Miss-scheduling of highly dependant issues
Likelihood: 2/5
Severity: 2/5
Indication: Scheduled issues, for which no progress is made
Prevention: Take extra cation when planning issues that could have high dependencies. Highlight progress of coupled issues on the DSM.
Damage Control: Allocate more resources to the blocking issue and prioritize highly if it is essential for the projects progress. Otherwise push it to a later sprint.

7.8 R8

Description: Conflicting schedules giving meetings low attendance
Likelihood: 3/5
Severity: 1/5
Indication: Few and small windows where meetings can be scheduled
Prevention: Schedule the most important meetings far in advance before the team members schedule fills up. For example plan the most important stand-ups on zoom ahead of time and have the less important stand-ups on slack, where it is easier to attend sometime during the day by text.
Damage Control: Have meeting with few attendees and share the information discussed and action list created with ones missing the meeting. Ask if they have something to ad