

# **Software Engineering Group Project Project Plan**

Author: David Fairbrother, Joshua Doyle  
Config Ref: SE\_05\_PM\_01  
Date: 2015-10-29  
Version: 1.2  
Status: Release

## CONTENTS

CONTENTS .....	2
1. INTRODUCTION .....	3
1.1 Purpose of this Document .....	3
1.2 Scope.....	3
1.3 Objectives.....	3
2. DELIVERABLES .....	3
2.1 List of deliverables and their deadlines .....	3
2.2 Additional important dates .....	3
2.3 Individual deliverable requirements .....	4
3. TASK STANDARDS .....	5
3.1 Naming convention .....	5
3.2 Task groups .....	5
4. LIST OF TASKS .....	5
4.1 Project Management Tasks .....	5
4.2 Test Specification Tasks .....	6
4.3 Design Specification Tasks .....	6
4.4 1 <sup>st</sup> Prototype Tasks .....	6
4.5 Software delivery Tasks .....	7
4.6 Documentation handover Tasks .....	7
5. MONITORING.....	7
5.1 Review meetings .....	7
5.2 Informal meetings .....	8
5.3 Formal meetings.....	8
5.4 Minutes .....	8
5.5 Time sheets .....	8
5.6 Blogs .....	8
6. GANTT CHART .....	9
7. COMPLETION DATES .....	10
8. RISK ANALYSIS.....	11
REFERENCES .....	12
DOCUMENT HISTORY .....	12

# 1. INTRODUCTION

This document outlines deadlines for deliverables and their requirements. It breaks these deliverables into tasks and creates a plan for their completion.

## 1.1 Purpose of this Document

The purpose of this document is to describe how tasks will be created, named and monitored to ensure deliverables are created within set deadlines and are of a high quality.

## 1.2 Scope

This document is written to assist in planning, timing and management of the project. It also lists tasks, deadlines and deliverables and creates a standard naming scheme and plan for their completion.

## 1.3 Objectives

This document aims to inform the reader of upcoming deliverables and their deadlines, it will break the deliverables down into assignable tasks with a standard naming convention. The project plan will also provide a schedule to detect problematic tasks and remediate the situation before time slippages occur. The reader should understand the timeline of the project and tasks required to complete it.

# 2. DELIVERABLES

## 2.1 List of deliverables and their deadlines

To ensure the project is progressing correctly and to a high standard, several deliverables will be presented at a review meeting with the project manager. These must be submitted via Blackboard before the meeting.

These deliverables and their deadlines [1] are:

1. Interaction and high level design for the system – October 30<sup>th</sup>
2. Test specification for the system – November 13<sup>th</sup>
3. Design specification for the final system – November 27<sup>th</sup>
4. 1<sup>st</sup> Prototype demonstration – December 11<sup>th</sup>
5. Delivery of software – January 29<sup>th</sup> by 16:00
6. Handover of all documentation – February 15<sup>th</sup> by 16:00

## 2.2 Additional important dates

There are other important dates within this project. These are listed below:

- Integration and testing week - January 25<sup>th</sup> – January 29<sup>th</sup> (Inclusive)
- Acceptance testing – Week commencing February 1<sup>st</sup> (times to be announced)

## **2.3 Individual deliverable requirements**

### **2.3.1 Interaction and high level design for the system**

Sections 4.1, 4.2, 5.1 and 5.2 within SE.QA.05A [2] are required for this deliverable. This includes:

- Applications in the system
- Application interactions
- Use-cases
- User interface design

### **2.3.2 Test specification for the final system**

This deliverable requires the test specification to be completed as per SE.QA.06 [3] the documentation should include:

- Test specification

### **2.3.3 Design Specification**

A full design specification must be created conforming to SE.QA.05A [2]. The following sections can be reused from Interaction and high level system design deliverable

- Applications in the system
- Application integrations
- Use-Cases
- User interfaces design

The following documentation must be created in addition:

- Component descriptions
- Significant classes
  - Interface design for these classes

To ensure a complex class or interaction is implemented correctly it also needs to be broken down into:

- Sequence diagrams
- State diagrams
- Activity diagrams
- Significant data structures

### **2.3.4 1<sup>st</sup> Prototype demonstration**

A demonstration of the final product needs to be created; this is to allow the client to review and change elements of the product early in development. The demonstration must have all its major components completed however it does not need to be fully complete or polished.

### **2.3.5 Delivery of software**

The software must contain all functional requirements and be polished and ready for customer delivery. The software will be delivered on a CD to the client. The client will then perform acceptance testing.

### 2.3.6 Handover of all documentation

All documentation must be completed and reviewed to ensure a consistent high standard across documents. All documents must also conform to the layout in SE.QA.03 [4]. Once finalised the documents will be exported to PDF and delivered to the client. The documentation includes [5]:

- Design specification
- Final report
- Maintenance manual
- Project plan
- Test report
- Test specification

## 3. TASK STANDARDS

### 3.1 Naming convention

All tasks will follow a standard naming scheme to allow identification of the deliverable targeted this is described in SE.QA.02 [6]. The format will be SE\_05\_XXXX\_YY.

XXXX refers to the deliverable in a shorthand code such as PM for project management. YY is a numerical value starting and 01 and increasing for tasks which are large enough to warrant being broken down into subtasks. Tasks which are not suitable to be turned into sub tasks will have a subtask value of 01. [6]

### 3.2 Task groups

The following codes will be used for identifying tasks:

- DEGN – Design Specification
- MAINT – Maintenance Manual
- PM – Project Management
- QA – Quality Assurance
- TCLI – Tasker Client
- TEST– Testing Reports
- TEST\_SPEC – Testing Specification
- TMAN – Tasker Manager
- TSRV – Tasker Server
- DEL – Complete Deliverable

For example a design document might have the code SE\_05\_DEGN\_07 (if it was subtask 7)

## 4. LIST OF TASKS

### 4.1 Project Management Tasks

SE\_05\_PM\_01 – Create Project plan documentation (This document)

SE\_05\_PM\_02 – Create Gantt chart for project plan

Se\_05\_PM\_03 – Create risk analysis for project plan

## 4.2 Test Specification Tasks

SE\_05\_TEST\_SPEC\_01 – Create test specification for TaskerCLI

SE\_05\_TEST\_SPEC\_02 – Create test specification for TaskerMAN

SE\_05\_TEST\_SPEC\_03 – Create test specification for TaskerSRV

SE\_05\_DEL\_02 – Complete test specification deliverable

## 4.3 Design Specification Tasks

SE\_05\_DEGN\_01 – Create system applications documentation

SE\_05\_DEGN\_02 – Create application interaction documentation which lists to format data will be transmitted between software in

SE\_05\_DEGN\_03 – Create use-case diagrams for the application

SE\_05\_DEGN\_04 – Design and create mock-up of user interfaces

SE\_05\_DEL\_01 – Interaction and high level design deliverable

SE\_05\_DEGN\_05 – Document component level design

SE\_05\_DEGN\_06 – Design and document significant classes within TaskerCLI

SE\_05\_DEGN\_07 – Design and document significant functions and/or classes (where applicable) in TaskerMAN

SE\_05\_DEGN\_08 – Design scheme and indexes for TaskerSRV database

SE\_05\_DEGN\_09 – Break down complex designs and algorithms into UML sequence diagrams, state diagrams and activity diagrams where appropriate for TaskerCLI

SE\_05\_DEGN\_10 – Break down complex designs and algorithms into UML sequence diagrams, state diagrams and activity diagrams where appropriate for TaskerMAN

SE\_05\_DEGN\_11 – Spike work and document methods such as threading to conform to requirement PR1 [7] (all user interactions must take < 1 second to reflect their changes)

SE\_05\_DEGN\_99 – Review all design documentation clarify and ambiguity and release

SE\_05\_DEL\_03 – Complete design specification deliverable

## 4.4 1<sup>st</sup> Prototype Tasks

### 4.4.1 TaskerCLI Tasks

SE\_05\_TCLI\_01 – Implement JUnit tests from specification

SE\_05\_TCLI\_02 – Create application interfaces

SE\_05\_TCLI\_03 – Implement classes from design specification and adhering to interfaces

SE\_05\_TCLI\_04 – Implement task synchronisation logic

SE\_05\_TCLI\_05 – Implement GUI

SE\_05\_TEST\_01 – First prototype integration and application testing

SE\_05\_TCLI\_06 – Performance and bug fixing

SE\_05\_DEL\_04 – Complete first prototype deliverable

### 4.4.2 TaskerMAN Tasks

SE\_05\_TMAN\_01 – Implement PHPUnit tests from specification

SE\_05\_TMAN\_02 – Implement HTML, CSS on static content

SE\_05\_TMAN\_03 – Implement basic editing of tasks

SE\_05\_TMAN\_04 – Implement sorting and filtering

SE\_05\_TMAN\_05 – Implement advanced editing of tasks (batch editing)

SE\_05\_TEST\_01 – First prototype integration and application testing

SE\_05\_TMAN\_06 – Performance and bug fixing

#### **4.4.3 TaskerSRV Tasks**

SE\_05\_TSRV\_01 – Setup development database

SE\_05\_TSRV\_02 – Create schema and indexes

SE\_05\_TSRV\_03 – Establish connections and populate with test data

SE\_05\_TSRV\_04 – Create automated package to setup database

SE\_05\_TEST\_01 – First prototype integration and application testing

SE\_05\_TSRV\_05 – Performance fixes and schema tweaks

#### **4.5 Software delivery Tasks**

SE\_05\_TCLI\_07 – Incorporate client feedback and fixes

SE\_05\_TMAN\_07 - Incorporate client feedback and fixes

SE\_05\_TSRV\_06 - Incorporate client feedback and fixes

SE\_05\_TEST\_02 – Testing for all targeted platforms and bug fixes

SE\_05\_MAINT\_01 – Create maintenance documentation

SE\_05\_TCLI\_08 – Package software ready for handover

SE\_05\_TMAN\_08 – Package software ready for handover

SE\_05\_TSRV\_07 – Package database scripts ready for handover

SE\_05\_PM\_04 – Deliver software to client on CD

#### **4.6 Documentation handover Tasks**

SE\_05\_QA\_01 – Review testing documentation

SE\_05\_QA\_02 – Review design documentation

SE\_05\_QA\_03 – Review maintenance manuals

SE\_05\_QA\_04 – Review miscellaneous documents

SE\_05\_DEL\_05 – Testing report

SE\_05\_DEL\_06 – Handover all documentation

## **5. MONITORING**

### **5.1 Review meetings**

Review meetings will be conducted as per SE.QA.07 [8]. Before the meeting the QA manager will inform all relevant members of the team the location and time of the meeting. They will also ensure the correct version of the document is being used and distribute this copy before the meeting. During the meeting a brief version of minutes may be taken and distributed subject to a decision between the QA manager and project leader. The QA

manager will take any points that arise and create issues on GitHub and ensure they are assigned. A document must go through a review before it can have its status changed to review

## **5.2 Informal meetings**

There will also be informal meetings held on a regular basis. These are to co-ordinate and assign tasks, check progress and raise points within the group. Two informal meetings are planned every week to allow constant communication within the group [8]. These will be held Tuesday at 6PM and Wednesday at 2PM in the Think Tank. In the event of a meeting time being changed, cancelled or moved the project manager will email the group and inform and confirm with the group the alternative arrangements. Brief minutes and actionable points will be taken by the QA manager and distributed to the group. Any issues arising will be opened and assigned on GitHub.

## **5.3 Formal meetings**

Formal meetings will take place once per week with the project manager. In these meeting minutes must be taken by any member of the group present, and then officially typed up by the QA manager. The project manager will review minutes from the previous meeting and create and assign any actions to ensure the project continues correctly and on time. The project manager will also ensure progress has been made on previously assigned actions.

## **5.4 Minutes**

All minutes must conform to the layout specified in SE.QA.03 [4] these minutes should be uploaded to the repository as soon as possible. Each set of minutes should be in an individual file with the name style [9] of yyyy-mm-dd\_minutes where yyyy represents year, mm represents month, dd represents day in numerical form.

## **5.5 Time sheets**

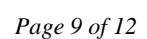
Time sheets will be created and conform to the layout provided by the project leader. Within these all project members will estimate the time required to complete a task, the time taken to complete the task, the task ID and name and any comments about the task. These will be submitted by Wednesday evening, they must show the tasks completed from the previous Wednesday to the current Wednesday.

These will be used to calculate hours available to the project, hours completed and any over or under run.

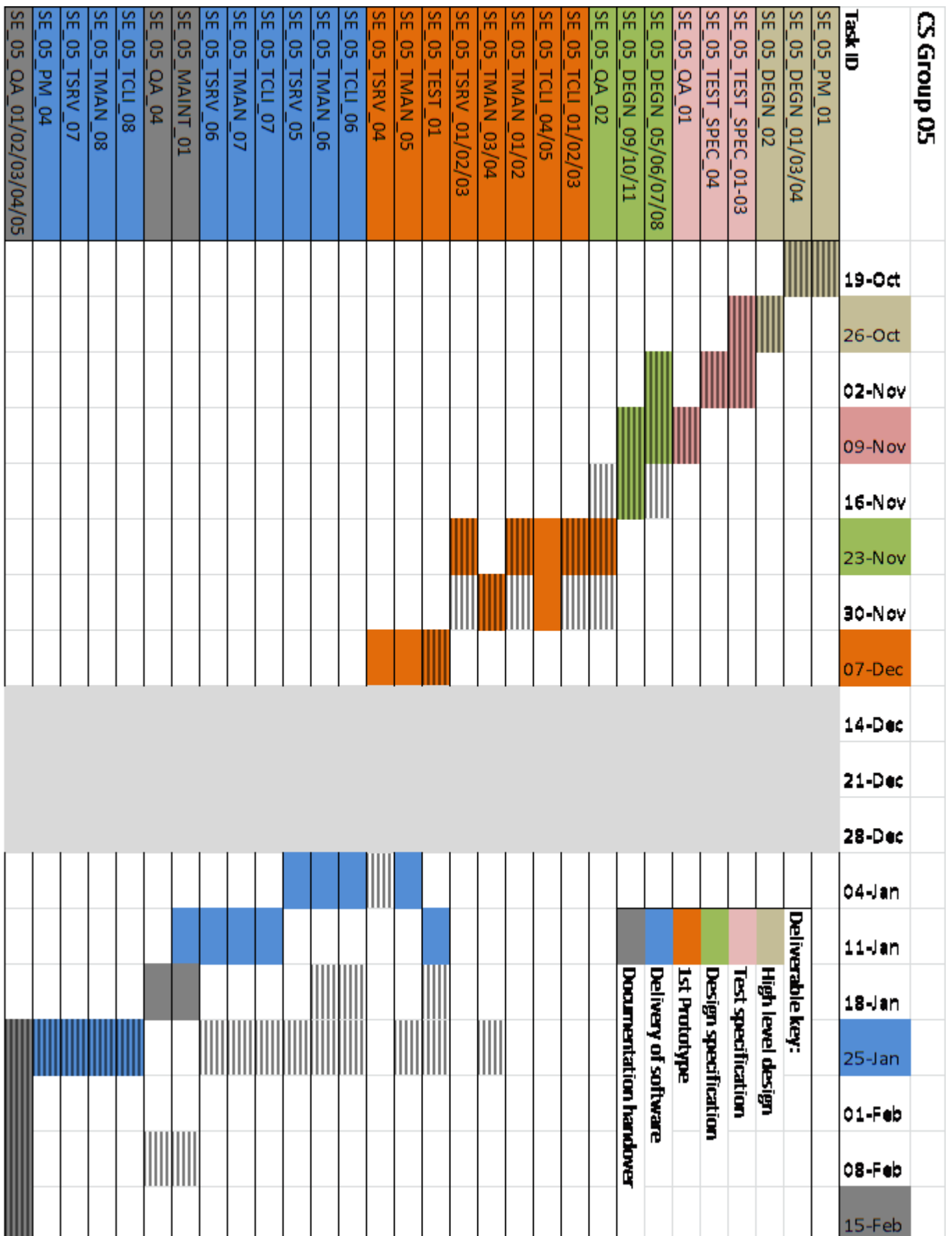
## **5.6 Blogs**

All project members will create and maintain a blog. This will list the task they are currently working on, any comments about the task and difficulties encountered during that task. The project manager will monitor these to view the project member's progress.





## 7. COMPLETION DATES



## 8. RISK ANALYSIS

Risk ID	Risk Description	Impact	Probability	Mitigation
001	Work not in on schedule	Medium	Medium	Monitoring of assigned tasks and regular feedback from team members
002	Project leader missing	Medium	Low	Deputy leader to take over when project leader missing, project leader must keep deputy leader informed
003	QA lead missing	Medium	Low	Deputy QA to take over. QA lead to keep deputy informed of documents pending review and review dates
004	Work corrupted	Medium	Low	Revert to previous commit, if still corrupted pull from server repository
005	Local work deleted	Medium	Low	Pull latest commit from GitHub and recreate local copy
006	Remote work deleted	High	Low	Revert latest commit. Upload latest version to GitHub from a team members local computer (determined in meeting) recreated repository
007	GitHub temporarily offline	Medium	Low	Watch status page at <a href="https://status.github.com/">https://status.github.com/</a> and work locally
008	GitHub permanently offline	V. High	V. Low	Consult with project manager, migrate hosts to alternative Git host. Upload most recent local copy of work (determined in meeting)
009	Project member illness	Low-Med	Medium	Depending on severity reallocate tasks to another member, if possible get existing work off project member
010	Client requirements change	Med-High	Low	Consult regularly with client and ensure project is progressing with their satisfaction. Meeting with team and client to compromise on any requirement changes
011	GUI implementation problems	Medium	Medium	Spike work will be created early in project to create code that produces output like GUI design. In event this is not easy to implement the GUI design will be altered.
012	Client synchronisation issues	High	Medium	Spike work will be conducted early in the project to view ways of dealing with intermittent connections and sync issues

## REFERENCES

- [1] Aberystwyth University, "Project Timetable: The Software Development Life Cycle (2015-16)," [Online]. Available: [https://blackboard.aber.ac.uk/webapps/blackboard/content/listContent.jsp?course\\_id=\\_12897\\_1&content\\_id=\\_557616\\_1](https://blackboard.aber.ac.uk/webapps/blackboard/content/listContent.jsp?course_id=_12897_1&content_id=_557616_1). [Accessed 19 10 2015].
- [2] N. W. Hardy, C. J. Price and B. P. Tiddeman, *SE.QA.05 A 1.8 - Design Specification Standards*, Aberystwyth University: Software Engineering Group Project, 2015.
- [3] N. W. Hardy, C. J. Price and B. P. Tiddeman, *SE.QA.06 1.8 - Test Procedure Standards*, Aberystwyth University: Software Engineering Group Project, 2015.
- [4] N. W. Hardy, C. J. Price and B. P. Tiddeman, *SE.QA.03 1.8 - General Documentation Standards*, Aberystwyth University: Software Engineering Group Project, 2015.
- [5] N. W. Hardy, C. J. Price and B. P. Tiddeman, *SE.QA.01 1.10 - Quality Assurance Plan*, Aberystwyth University: Software Engineering Group, 2015.
- [6] C. J. Price, *SE.QA.02 - Project Management Standards 1.9*, Aberystwyth University: Software Engineering Group Project, 2015.
- [7] N. W. Hardy, *Tasker Team Tasking System - Requirement Specification 1.1*, Aberystwyth University: Software Engineering Group Project, 2015.
- [8] C. J. Price, N. W. Hardy and B. P. Tiddeman, *SE.QA.07 - Review Standards 1.6*, Aberystwyth University: Software Engineering Group Project, 2015.
- [9] N. W. Hardy, C. J. Price and B. P. Tiddeman, *SE.QA.08 1.8 - Operating Procedures and Configuration Management Standards*, Aberystwyth University: Software Engineering Group Project, 2015.

## DOCUMENT HISTORY

Version	CCF No.	Date	Changes made to document	Changed by
1.0	N/A	19/10/2015	First draft ready for review	DAF5
1.1	N/A	29/10/2015	Corrected spellings of JUnit and PHPUnit	JOD32
1.2	#18	30/10/2015	Corrected spelling in risk analysis and added GUI and sync logic to risk analysis	DAF5
1.3	188	13/02/2016	Added document versions to reference list	DAF5
1.4	189	13/02/2016	Added completion dates to Gantt chart	DAF5