SOFTWARE SOLITIONS

SOFIA

Planning and Progress Review

Progress and Planning Review and Reflection

Document Control

Editor	Version	Date	Update
Alex Cash	1.0	03/06/2015	Created Document

Table of Contents

Planning and Progress Review

1.0 Introduction

This document is a retrospective review of our planning and management processes throughout the project in regards to implementation and testing alongside creating other deliverables. This report shall discuss all three iterations, the user stories they covered, how the code was implemented and assigned, and how we used our iterative process to make progress on coding whilst still delivering the required external deliverables. This review shall critique our methods and aim to suggest areas which could have been improved.

Sub Heading:

2.0 Team Meetings

As we were operating an agile team methodology, we wanted to replicate the concept of "standups" as best we could. In case the reader is not familiar, in a real agile environment, teams will meet daily for a short meeting (anywhere between 5 and 20 minutes) whilst standing to talk about progress and work to pick up. This promotes communication and by standing forces participants to focus and become actively involved. Due to the nature of the project, it was not possible to meet on a daily basis (nor were short meetings suitable). A good compromised was reached by conducting two formal meetings a week, enabling progress updates and planning to take place. These meetings were all recorded in our minutes documents. The organisation of these meetings was the responsibility of the Project Manager and a sample of the organised meetings can be seen below:

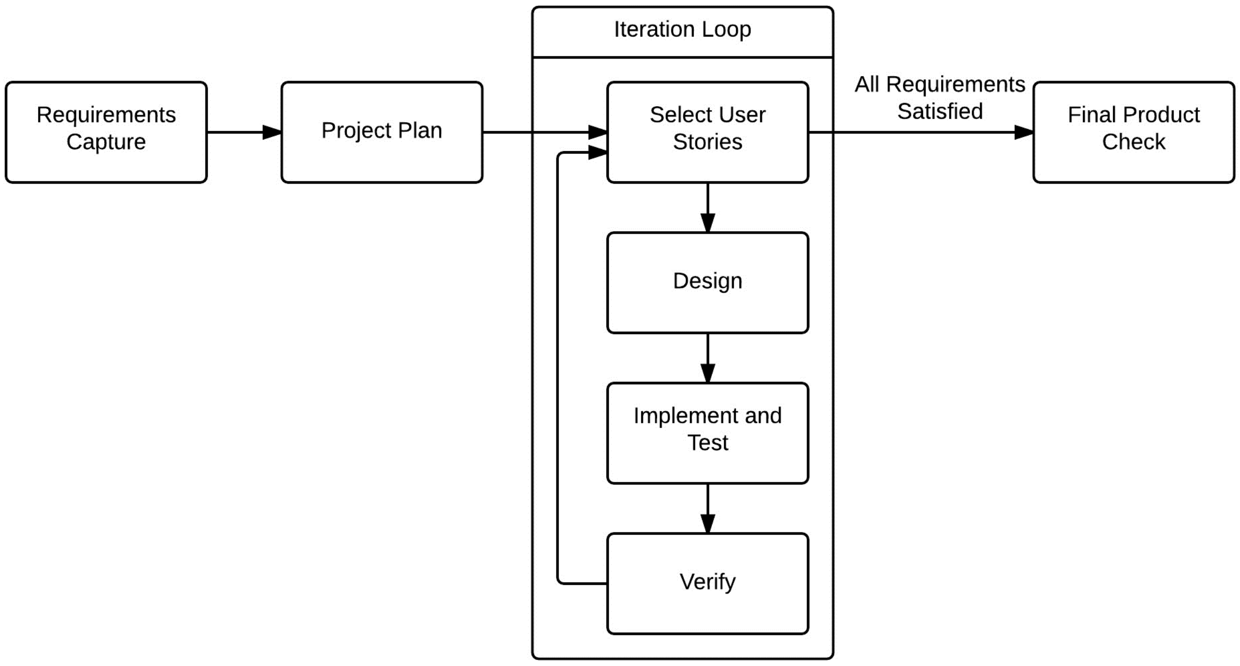
DAY	DATE		ROOM	TIME
Monday	23/02/2015		JP002	12
Wednesday	25/02/2015		JP003	2
Monday	2/3/2015		JP002	12
Wednesday	4/3/2015		JP003	2
Monday	9/3/2015		JP002	12
Wednesday	11/3/2015		JP003	2
Monday	13/04/2015		JP002	12
Monday	20/04/2015		JP002	12
Friday	24/04/2015		JP005	12
Monday	27/04/2015		JP002	12
Friday	1/5/2015		JQ001	12
Monday	4/5/2015		JP002	12

Friday	8/5/2015	G013	12
Monday	11/5/2015	JP003	12
Friday	15/05/2015	JP005	12
Monday	18/05/2015	JP002	12
Friday	22/05/2015	JP005	12
Monday	25/05/2015	JP002	12
Friday	29/05/2015	JQ001	12

Although our code development process was agile, we can describe our work outside of code development as being a more traditional and non-agile style of work. The other tasks, which shall be discussed later, had pre-defined deadlines and were straightforward to assign staff to, so were handled separately from code development.

3.0 The Adopted Iterative Process and Initial Planning Stages

As can be seen in the Iterative Review Document, we used the following iterative process to complete our project:



The first process seen above was the "requirements capture"; this involved deciding on a product that fit the customer specifications and generating a rough idea of how the product should behave, captured as user stories. These user stories shall be discussed later in this document.

3.1 Requirements Capture

During the requirements capture phase, we worked as a team to define our product into user stories. The initial product idea was suggested by our Contracts and Documentation manager and was welcomed by the team who believed it had real potential. We then came to the conclusion as a team that we would like to create a suite of products, one for students and one for teachers. This would require a large amount of development but we felt that the team possessed a number of very strong programmers (primarily the Lead Software Developer and the Specialist Software Developer) and could handle the task.

Through our meetings we managed to develop a general idea of our product which we captured in the following user stories for the TeachEasy client:

> "As a teacher, I can provide any number of my students with access to a lesson I created previously"
>
> "As a teacher, I can create a lesson comprised of a number of discrete pages, each of which I can customise"
>
> "As a teacher, I can pick a page category from a number of pre-defined templates (e.g. video, quiz, etc.)"
>
> "As a teacher, I can include multimedia objects in the lessons I create"
>
> "As a teacher, I can save lessons I am working on then access them again and edit them at a later date"
>
> "As a teacher, I can assign marks to exercises on each page, where necessary"

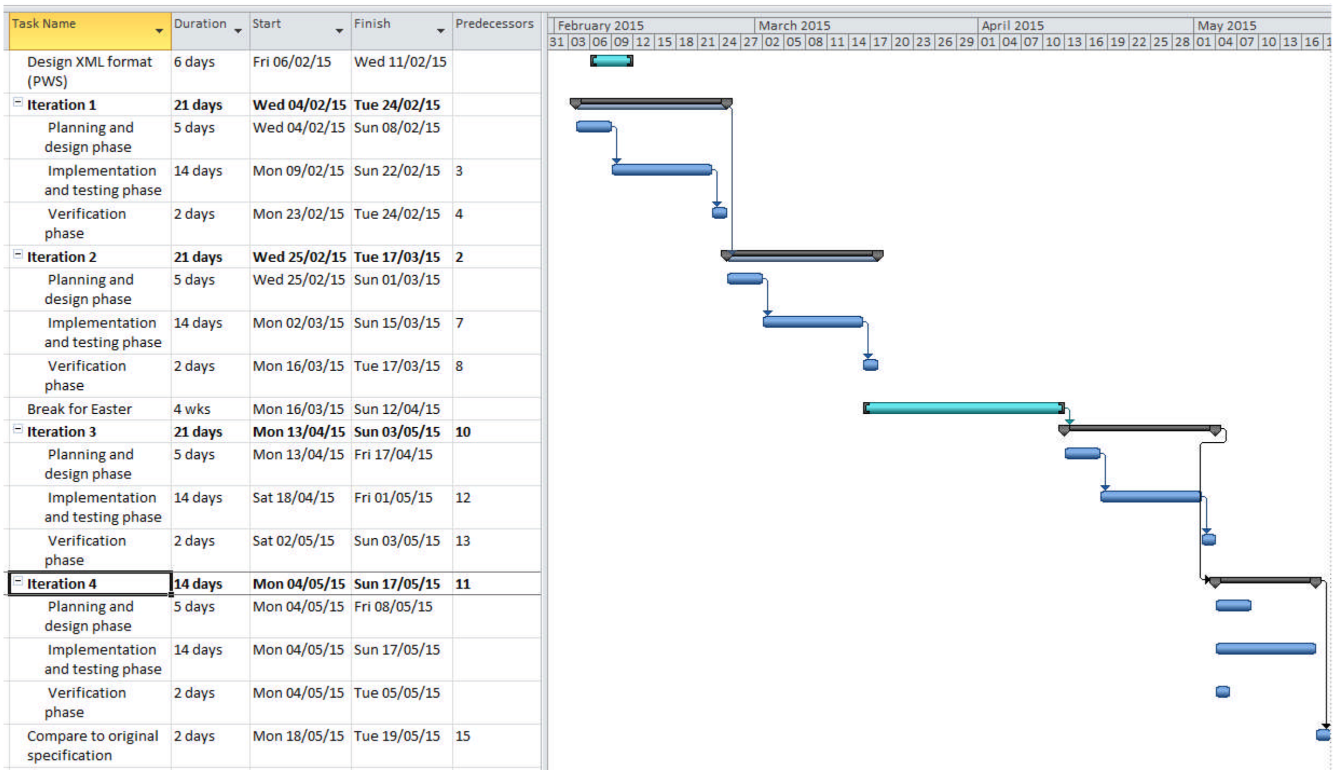
And the following user stories for the LearnEasy client:

> "As a student, I can view a lesson created by my teacher"
>
> "As a student, I cannot edit a lesson"
>
> "As a student, I can choose the lesson I want to work on from a selection of lessons provided to me by my teacher(s)"
>
> "As a student, I can pause and resume lessons"
>
> "As a student, I can view all forms of multimedia that my teacher has included in the lesson"
>
> "As a student, I can interact with all suitable forms of multimedia (e.g. pause a video)"
>
> "As a student, I am provided with a record detailing my lesson completion and level of achievement, which I can choose to print out"

These turned out to be surprisingly good user stories as they did not need modification throughout the project, and accurately described the end products.

3.2 Project Plan

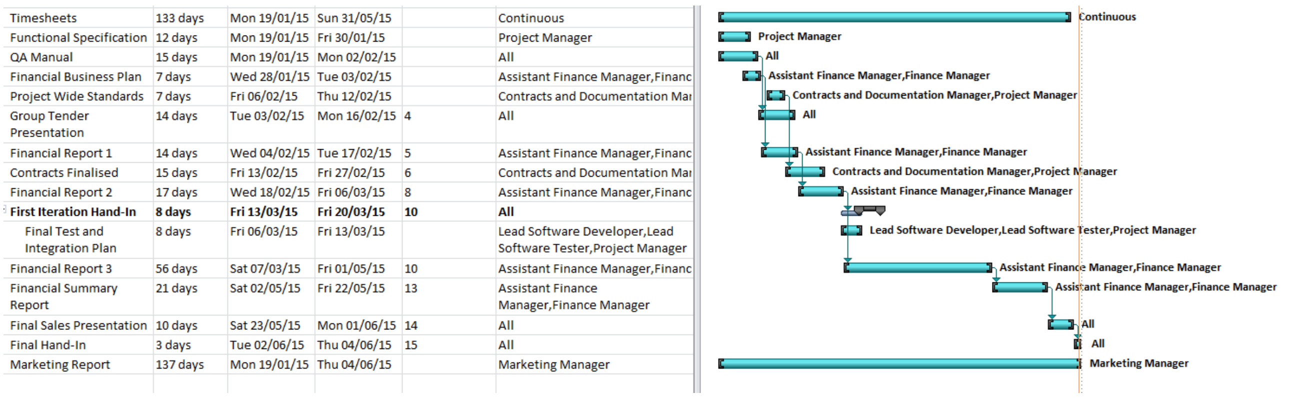
As can be seen, our first step once requirements were captured was to create an overall project plan. This was meant to be a very general overview of the timings associated with the project, and was not designed to describe actual tasks. As we were using a very agile methodology, it would not have made sense to try and plan anything past our first iteration until the iteration was complete. As such, the "project plan" stage that occurs before the iteration loop began was a very short process led by the Project Manager. This resulted in the production of a very general Gantt chart used to loosely plan our iteration time frames. This can be seen below:



Task Name	Duration	Start	Finish	Predecessors
Design XML format (PWS)	6 days	Fri 06/02/15	Wed 11/02/15	
Iteration 1	**21 days**	**Wed 04/02/15**	**Tue 24/02/15**	
Planning and design phase	5 days	Wed 04/02/15	Sun 08/02/15	
Implementation and testing phase	14 days	Mon 09/02/15	Sun 22/02/15	3
Verification phase	2 days	Mon 23/02/15	Tue 24/02/15	4
Iteration 2	**21 days**	**Wed 25/02/15**	**Tue 17/03/15**	2
Planning and design phase	5 days	Wed 25/02/15	Sun 01/03/15	
Implementation and testing phase	14 days	Mon 02/03/15	Sun 15/03/15	7
Verification phase	2 days	Mon 16/03/15	Tue 17/03/15	8
Break for Easter	4 wks	Mon 16/03/15	Sun 12/04/15	
Iteration 3	**21 days**	**Mon 13/04/15**	**Sun 03/05/15**	**10**
Planning and design phase	5 days	Mon 13/04/15	Fri 17/04/15	
Implementation and testing phase	14 days	Sat 18/04/15	Fri 01/05/15	12
Verification phase	2 days	Sat 02/05/15	Sun 03/05/15	13
Iteration 4	**14 days**	**Mon 04/05/15**	**Sun 17/05/15**	**11**
Planning and design phase	5 days	Mon 04/05/15	Fri 08/05/15	
Implementation and testing phase	14 days	Mon 04/05/15	Sun 17/05/15	
Verification phase	2 days	Mon 04/05/15	Tue 05/05/15	
Compare to original specification	2 days	Mon 18/05/15	Tue 19/05/15	15

This Gantt chart ended up being updated part way through the project, which shall be discussed later.

Outside of our iterative process for code implementation there were a number of defined deliverables that were handled using a different method. For these deliverables (such as the QA manual) we had defined requirements and deadlines. This enabled us to use a non-agile system with more predefined work schedules and objectives. For this, a Gantt chart was created to track these tasks and provide clear working times, this is as follows:



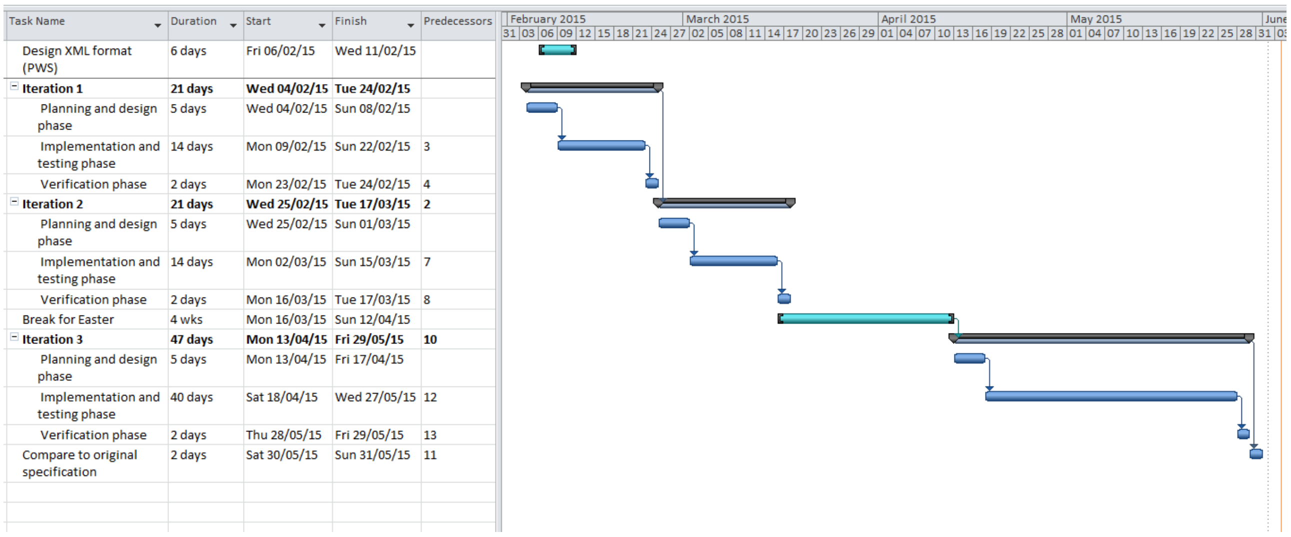
Timesheets	133 days	Mon 19/01/15	Sun 31/05/15		Continuous
Functional Specification	12 days	Mon 19/01/15	Fri 30/01/15		Project Manager
QA Manual	15 days	Mon 19/01/15	Mon 02/02/15		All
Financial Business Plan	7 days	Wed 28/01/15	Tue 03/02/15		Assistant Finance Manager,Financ
Project Wide Standards	7 days	Fri 06/02/15	Thu 12/02/15		Contracts and Documentation Mai
Group Tender Presentation	14 days	Tue 03/02/15	Mon 16/02/15	4	All
Financial Report 1	14 days	Wed 04/02/15	Tue 17/02/15	5	Assistant Finance Manager,Financ
Contracts Finalised	15 days	Fri 13/02/15	Fri 27/02/15	6	Contracts and Documentation Mai
Financial Report 2	17 days	Wed 18/02/15	Fri 06/03/15	8	Assistant Finance Manager,Financ
First Iteration Hand-In	**8 days**	**Fri 13/03/15**	**Fri 20/03/15**	**10**	**All**
Final Test and Integration Plan	8 days	Fri 06/03/15	Fri 13/03/15		Lead Software Developer,Lead Software Tester,Project Manager
Financial Report 3	56 days	Sat 07/03/15	Fri 01/05/15	10	Assistant Finance Manager,Financ
Financial Summary Report	21 days	Sat 02/05/15	Fri 22/05/15	13	Assistant Finance Manager,Finance Manager
Final Sales Presentation	10 days	Sat 23/05/15	Mon 01/06/15	14	All
Final Hand-In	3 days	Tue 02/06/15	Thu 04/06/15	15	All
Marketing Report	137 days	Mon 19/01/15	Thu 04/06/15		Marketing Manager

This can be seen more clearly in the Microsoft Project file named Non Agile Deliverables.

4.0 Code Development Iterations

For a breakdown of each iteration and their reviews, please see the Iterative Review Document. Additionally, for tracking of development tasks by iteration including specific tasks, assignees, and the produced java files, please see the Implementation Breakdown Spreadsheet (Excel file).

User stories were chosen at the beginning of each iteration and used to create a software implementation plan based on fulfilling those user stories. This process was used to great effect for iterations 1 and 2. However, part way through iteration 3 it became clear that the iteration was going to be quite quick and would leave too much to complete for iteration 4 in order for us to hit our deadline. As a team we decided to incorporate iteration 4 into iteration 3, thus making release 2 equivalent to iteration 3. As such, the original Gantt chart was updated to:



Task Name	Duration	Start	Finish	Predecessors
Design XML format (PWS)	6 days	Fri 06/02/15	Wed 11/02/15	
Iteration 1	**21 days**	**Wed 04/02/15**	**Tue 24/02/15**	
Planning and design phase	5 days	Wed 04/02/15	Sun 08/02/15	
Implementation and testing phase	14 days	Mon 09/02/15	Sun 22/02/15	3
Verification phase	2 days	Mon 23/02/15	Tue 24/02/15	4
Iteration 2	**21 days**	**Wed 25/02/15**	**Tue 17/03/15**	2
Planning and design phase	5 days	Wed 25/02/15	Sun 01/03/15	
Implementation and testing phase	14 days	Mon 02/03/15	Sun 15/03/15	7
Verification phase	2 days	Mon 16/03/15	Tue 17/03/15	8
Break for Easter	4 wks	Mon 16/03/15	Sun 12/04/15	
Iteration 3	**47 days**	**Mon 13/04/15**	**Fri 29/05/15**	10
Planning and design phase	5 days	Mon 13/04/15	Fri 17/04/15	
Implementation and testing phase	40 days	Sat 18/04/15	Wed 27/05/15	12
Verification phase	2 days	Thu 28/05/15	Fri 29/05/15	13
Compare to original specification	2 days	Sat 30/05/15	Sun 31/05/15	11

Overall our implementations were successful and managed to create excellent products that the team as a whole are proud of.

5.0 Methodology Summary

In summary, the methods used to track and plan work proved very effective. Our agile development system worked particularly well to allow flexibility and quick turnaround of iterations, resulting in high quality products that met requirements. Splitting the team's workload into two different management styles also proved effective; an agile methodology simply would not have worked well for managing deliverables outside of code. This enabled us to be very flexible with who was working on what, allowing the Project Manager to assign staff to produce deliverables when necessary and easily swap them back into the implementation program when required.

Our meeting style also proved to be beneficial, resulting in frequent and thorough minutes, and a team that was able to stay on track throughout the project. We made do with timing restrictions due to other requirements to meet frequently as a whole team which most definitely enhanced collaboration as well as team morale and bonding.