

Software Development Plan
for
University X Library System

Table of Contents

1. Introduction

2. Project organisation

3. Managerial process

4. Software engineering process

5. Work packages (WPs), schedule and budget

Appendix A: Work package [and main activity] descriptions

1. Introduction

1.1 Project overview

This document is the plan for development of a library system for University X. The system consists of both computer software and hardware, excluding the network infrastructure [1], [2].

1.2 Project deliverables and milestones

Work Package	Milestone	Deliverable
	Kick Off (KO)	
WP1: Requirement Analysis	End of WP1	Use Case Model
WP2: First Iteration	End of WP2	- Software for some uses cases - Server + 4 PCs
WP3: Second Iteration	End of WP3	- Software for all uses cases - Remaining PCs
WP4: Validation & Acceptance	Project End	"Sign Off"

Table 1.2-1: Main Deliverables & Milestones

1.3 Evolution of SDP

This document will be issued before KO (this issue), at the end of WP1 and at the end of WP2.

1.4 Reference materials

[1] Contract document, including Statement of Work

[2] Invitation to Tender

[3] Company Quality Management System (including templates, standards, guidelines, tool manuals, etc)

1.5 Definitions & acronyms

Self-explanatory. For example

COTS Commercial off-the-shelf software (e.g. Microsoft Office)

KO Kick Off or start point (of project)

QMS Quality management system

SDP Software development plan

WP Work package

2. Project organisation

2.1 Process model

This plan covers the development of the library system for University X and consists of the software development processes

requirements analysis

design

coding

as well as software verification, configuration management, quality assurance and project management. It also covers acquisition of computer and system installation at the University X's site.

Figure 2.1-1 provides an overview of the overall system development, installation and maintenance. The scope of the present plan is depicted by the boxed area (heavy dotted lines) in the figure.

As suggested by Table 1.2-1 the life-cycle model is an iterative one consisting of a requirements analysis phase for the complete system, followed by two iterations of development. The first iteration delivery, intended for customer familiarisation and feedback, will be a reduced system which implements a small number of key use cases installed on a limited number of computers, whereas the second iteration will provide the complete, fully tested system.

There will be a major project review at

- end of the requirements phase
- after delivery of the first iteration system
- on completion of system testing of the (complete) second iteration system
- after acceptance testing at the customer's site

Each review will have to be completed satisfactorily before authorisation to proceed to (i.e. make the transition to) the succeeding activities is given.

2.2 Organisational structure/2.3 Organisational boundaries & Interfaces

The following figure depicts the organisational structure of the project.

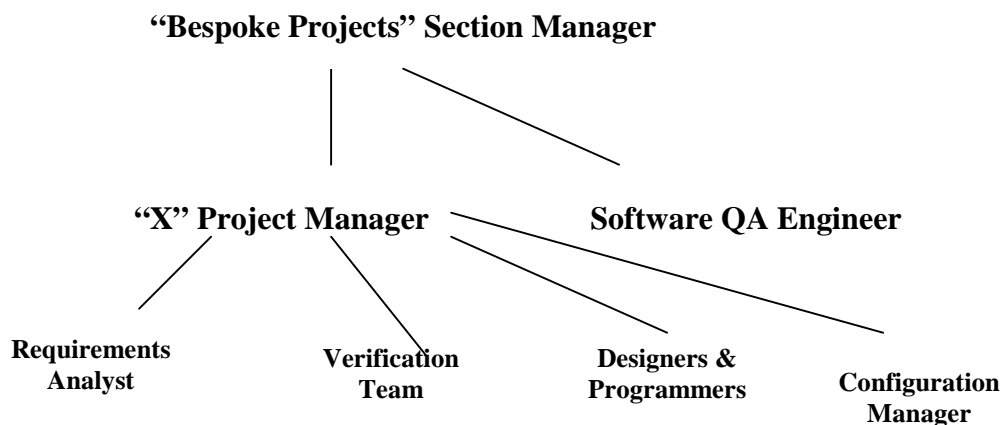


Figure 2.2-1: Project “X library System” organisational structure

2.3 Organisational boundaries & Interfaces

The following figure depicts where the project fits into the overall company structure, within the “Bespoke Projects” section of the Software Engineering division. In addition, the external interfaces of the project are identified by the dotted boxes and connecting lines.

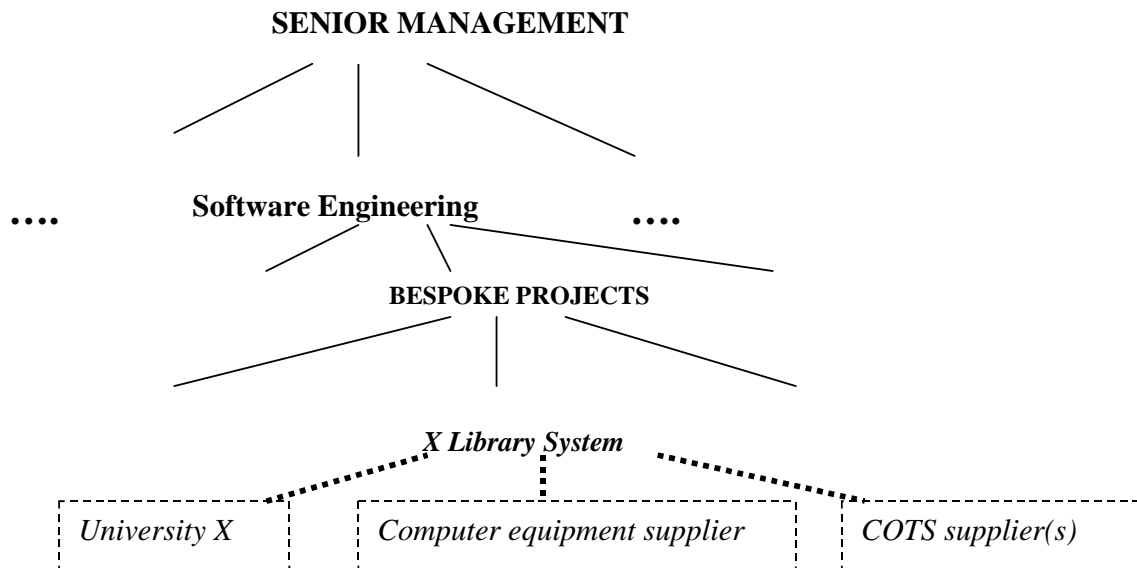


Figure 2.3-1: Project “X library System” context

2.4 Project responsibilities

Refer to Figure 2.2-1. All team members, apart from the independent Software Quality Assurance Engineer, report to the Project Manager.

The Requirements Analyst is responsible for WP1 (see Table 1.2-1).

A Senior Software Designer will be responsible for the design, coding and internal verification (including unit and integration testing) aspects of each of WP2 and WP3.

The Verification Team leader will be responsible for validation, particularly system testing, in WP4.

The Project Manager will be responsible for the installation aspects of WP2 and WP3, as well having an overall responsibility for configuration management.

The Project Manager and Software Quality Assurance Engineer will each have a key role in the Acceptance element of WP4.

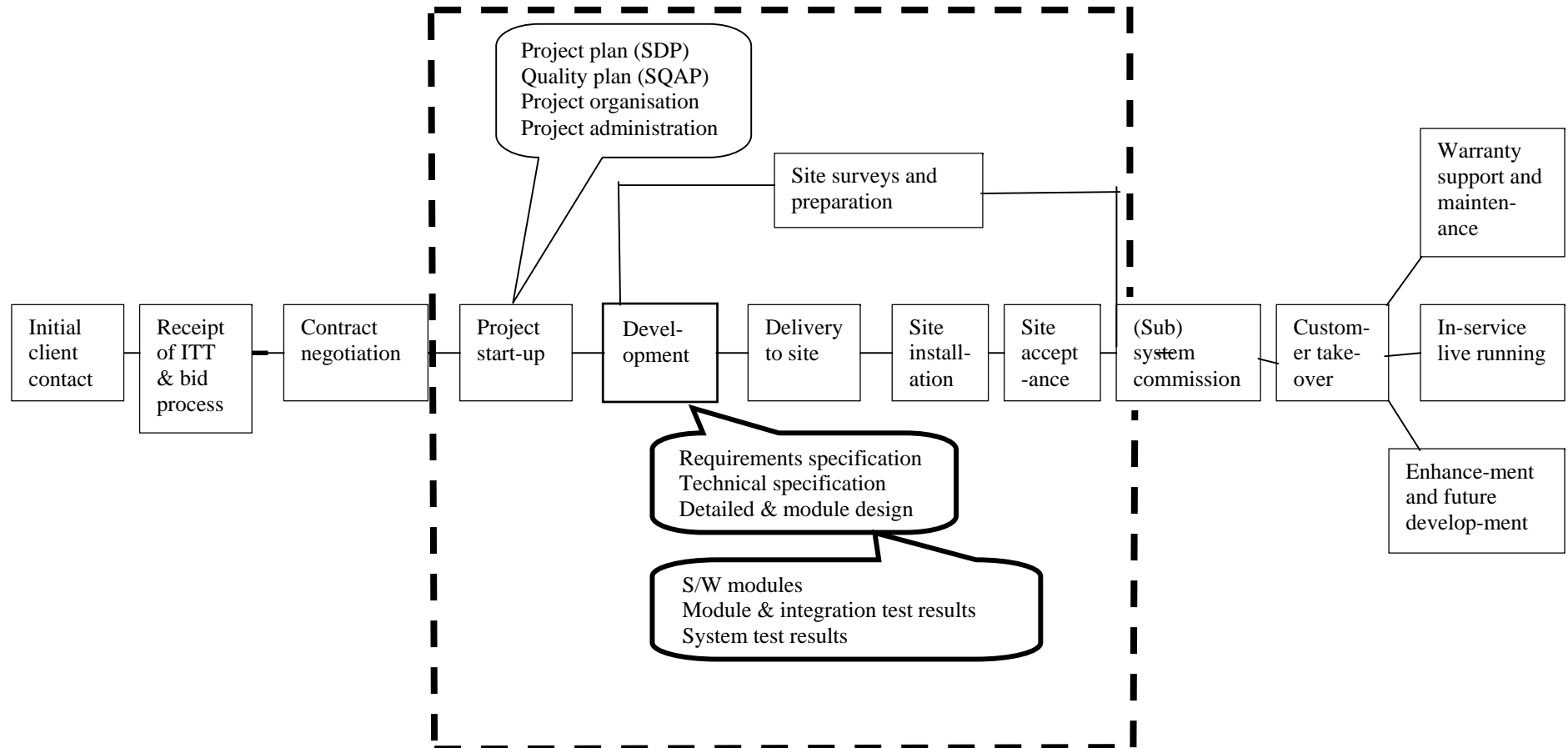


Figure 2.1-1: Overview of project and its context (major deliverables/results shown in “call-outs”)

3. Managerial process

3.1 Management objectives & priorities

The first priority is that a working version of the iteration one delivery (WP2) should be made at the scheduled date, even if it has not been completely verified. This is because the customer is arranging that users will be available at that date to familiarise with the system and to provide feedback.

The priority in iteration two (WP3, WP4) is that the system should be functionally complete and fully verified and validated, even if this means some delay in the delivery schedule (see contract [1]).

3.2 Assumptions, dependencies & constraints

It is assumed that the customer will authorise final agreement to the software requirements by the scheduled end date of WP1 (see section 5 of this plan).

.... maybe one or two others ...

3.3 Risk management

The major risk identified is that a number of other projects in the “Bespoke Projects” section are behind schedule so that personnel may not be available when required for “Project X”. Mitigation measures, particularly hiring of contract software engineers for detailed design, coding and unit testing will be taken should this risk materialise. However, it will be ensured that senior team members will be from the company’s own staff.

3.4 Monitoring and controlling mechanisms

“Earned Value Analysis”, calculated within Microsoft Project, will be used to analyse differences, for both resource and schedule, between planned and actual "expenditure".

As per the contract [1], monthly project status reports will be sent to the customer, summarising progress made, any problems encountered, and planned activities for the next month.

Major project reviews will be prepared for and organised in accordance with [3].

3.5 Staffing plan - more detail than section 2.4

Refer to Figure 2.2-1. More specific detail will be provided in the next issue of this plan (end of WP1).

4. Software engineering process

4.1 Standards, methods, tools and techniques

The company's standard work practices for software development [3] will be applied. In particular, the company's guidelines for the use of UML, its Java coding standards and its guidelines for interfacing to Oracle will be followed.

Details of the verification approach, including testing but also peer reviews and code inspections, will be specified separately in the Software Verification & Test Plan.

4.2 Project support functions

Separate Software Configuration Management and Software Quality Assurance plans will be issued.

4.3 Management of re-usable software products

It is not intended to incorporate any existing in-house software components. It is also not planned to make use of any non-standard Java components. The parts of the system interfacing with Oracle, as well as any operating system specific elements, will be isolated to facilitate maintenance.

It is not planned to produce any re-usable components in this project, though any promising re-usable opportunities will be documented and reported to the "Bespoke Projects" Section Manager.

4.4 Handling of critical requirements

While there are no major critical requirements for the system, appropriate technical measures will be taken to protect the privacy of library members' personal information.

5. Work packages (WPs), schedule and budget

5.1 Work breakdown structure (WBS), work packages and tasks

5.1.0 Work breakdown structure (WBS)

The following work breakdown structure (Figure 5.1.0-1) has been established for the project (using Microsoft Project). The planned schedule is also shown in the figure.

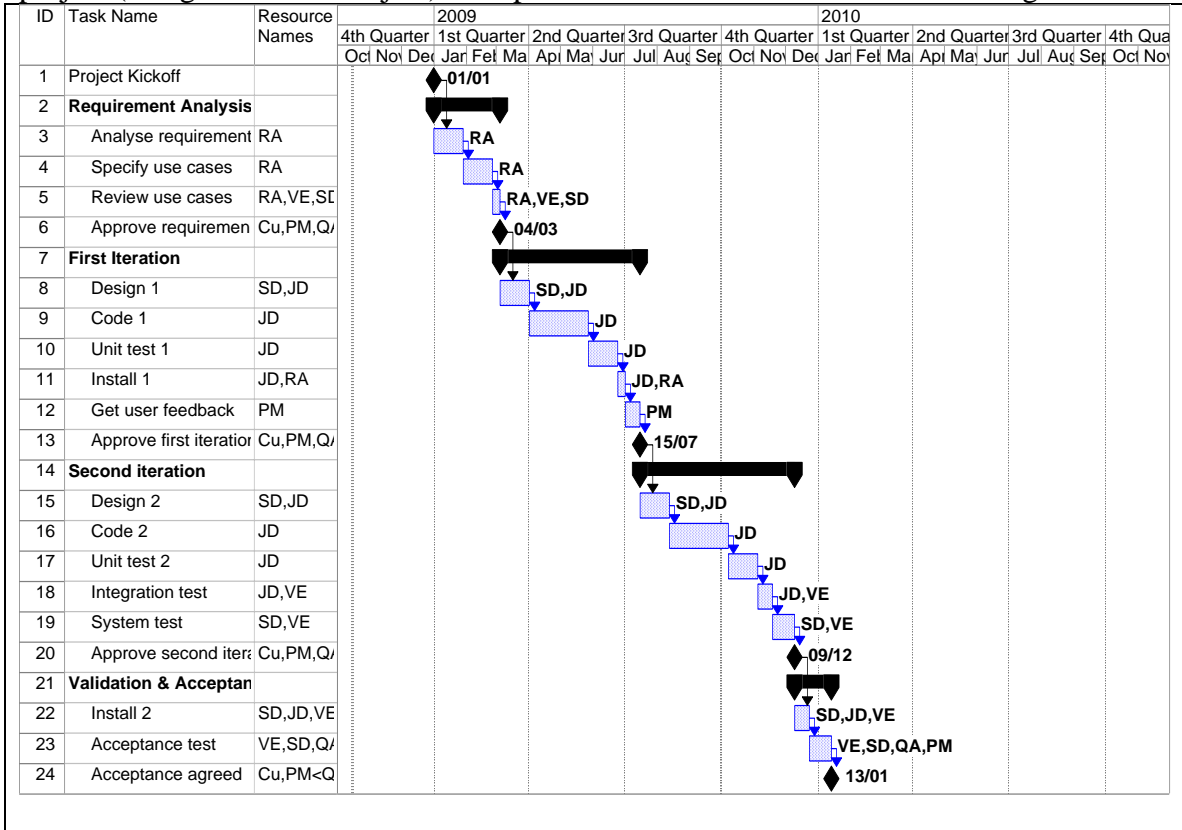


Figure 5.1.0-1: Project Work Breakdown Structure & Schedule

Detailed work package descriptions are provided in Appendix A.

5.2 WP and task interdependencies

Figure 5.10-1 depicts the task interdependencies, a network diagram being unnecessary in this simple case.

Also, each work package description (Appendix A) identifies the activity’s inputs and outputs, and any constraints on its execution.

5.3 Estimated resource requirements

The following table [not realistic!] is intended to give an impression of the capability of Microsoft Project to store and display resource information. Note that hardware costs have been omitted (as yet!).

ID	Resource Name	Work	Detail	2009												2010	
				1st Quarter			2nd Quarter			3rd Quarter			4th Quarter			1st Quarter	
				Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb
	Unassigned	0 h	Wor														
	Project Kick	0 h	Wor														
1	RA	400	Wor	17	16	24			32	8							
	Analyse re	160	Wor	16													
	Specify us	160	Wor	16	14												
	Review us	40	Wor		16	24											
	Install 1	40	Wor						32	8							
2	SD	680	Wor		16	17	8			96	64			64	18	72	
	Review us	40	Wor		16	24											
	Design 1	160	Wor			15	8										
	Design 2	160	Wor							96	64						
	System te	120	Wor											64	56		
	Install 2	80	Wor													80	
	Acceptanc	120	Wor													48	72
3	JD	1,480	Wor			15	17	16	17	10	16	17	17	10	80		
	Design 1	160	Wor			15	8										
	Code 1	320	Wor				16	15									
	Unit test 1	160	Wor					16	14								
	Install 1	40	Wor						32	8							
	Design 2	160	Wor							96	64						
	Code 2	320	Wor								10	17	40				
	Unit test 2	160	Wor										13	24			
	Integration	80	Wor											80			
	Install 2	80	Wor													80	
4	VE	440	Wor		16	24								14	18	72	
	Review us	40	Wor		16	24											

Table 5.3-1: Summary of required resources

Note on estimation:

Basic approach:

- Break work into “small” tasks, then estimate time and cost for each task. Finally estimate whole project by accumulating or synthesing the individual task estimates.
- It is wise to base estimates on valid past experience.
- In addition, other approaches may be sometimes useful such as
 - prototyping
 - cost models (e.g. COCOMO, function points).

As indicated in section 1.3 of this template, refined estimates can be documented in later versions of the SDP corresponding to more detailed versions of the WBS.

5.4 Budget and resource allocation

Table 5.3-1 gives an initial impression of what is involved.

5.5 Schedule

See Figure 5.1.0-1

Appendix A: Work package [and main activity] descriptions

Complete, as necessary, work package descriptions (WPDs) (or similar) for work packages and major activities. As a minimum include a WPD for each work package. A sample blank WPD form is as follows:

WORK PACKAGE DESCRIPTION (WPD) FORM

PROJECT: Library X			
WORK PACKAGE			
Title: Requirements Analysis	Reference:		
Manager:	Version & date:		
Planned	Start date:	End date:	Effort:
INPUTS:			
CONSTITUENT TASKS:			
CONSTRAINTS (e.g. task sequencing, environment, etc):			
OUTPUTS:			

NB: Above is incomplete & need similar for the other 3 work packages.