Seminole Software

Electronic Stamp Project

Software Development Plan

Version: 1.0

Date Created: 2003.09.04

Last Updated: 2003.09.12

Signatures

Date	Revision	Approved By
2003.09.12	1.0	Nikhil Bandodkar
2003.09.12	1.0	Jidong Long
2003.09.12	1.0	Chuck Weddle

List of Contributors

Name	Initials	Organization	E-Mail
Nikhil Bandodkar	NB	Seminole Software	bandodka@cs.fsu.edu
Jidong Long	JL	Seminole Software	jidolong@cs.fsu.edu
Chuck Weddle	CW	Seminole Software	weddle@cs.fsu.edu

Change History

Revision	Date	Description
1.0	2003.09.04	Initial Revision

Preface

This document represents the Software Development Plan for the Electronic Stamp project by Seminole Software. The Overview section describes the project purpose and most importantly the project scope and project objectives. This section also includes assumptions made and constraints the project will be working within.

The Reference section provides a complete list of all documents and other sources of information referenced in this document.

The Project Organization section identifies external and internal interfaces as organizational entities to the project. This section also defines roles and responsibilities for the project.

The Managing Process Plans section describes the project estimation and project staffing plan. Resource acquisition and resource training for the project is also described in this section. This section goes into great detail about the project work plan, including the resource allocation. Finally, the project Control plan is discussed, which includes quality control.

The Technical Process Plan section details the process model employed by the project along with the methods and tools used to help implement the process model. This section also defines the product acceptance plan.

The Supporting Process Plan section goes into detail about processes that assist and accommodate the execution of the project plan including the process for problem resolution.

Table of Contents

List of Figures

List of Tables

1 Overview

1.1 Project Summary

1.1.1 Purpose, Scope, and Objectives

The purpose of the Seminole Software Electronic Stamp project is to provide a solution to the ever increasing problem of unsolicited electronic mail. The solution proposed by Seminole Software is a software solution that will impose an economic reality to sending electronic mail that is similar to sending hard mail via the United States Post Office. The financial burden imposed on a mass mailing by the proposed solution will put financial responsibility on those producing unsolicited mass mailings. This financial responsibility will force reduced unsolicited emails.

It is within the scope of this document to discuss and describe the solution to the unsolicited electronic mail problem that will be implemented by Seminole Software. It is outside of the scope of this document to describe details about the problem of unsolicited electronic mail unless it is specific to the solution described in this document.

It is the objective of this document to describe in detail the Software Development Plan for the Seminole Software Electronic Stamp Project. This includes defining the process organization, the managerial process plan, the technical process plan, and the supporting process plan.

1.1.2 Assumptions

The Electronic Stamp project is working within set of assumptions that do influence the approach that Seminole Software is taking to develop the solution. The following table lists the assumptions made for the project.

Assumption	Detail
Static Requirements	The requirements for the Electronic Stamp project will not change.
Static Deliverable Dates	The deliverable dates, especially the end project date, will not change. Due to the fact that this is to be completed within the semester of coursework, the project deadline is static.
Static Team Members	The people that make up the team Seminole Software will not change.

Table 1: Assumptions

1.1.3 Project Deliverables

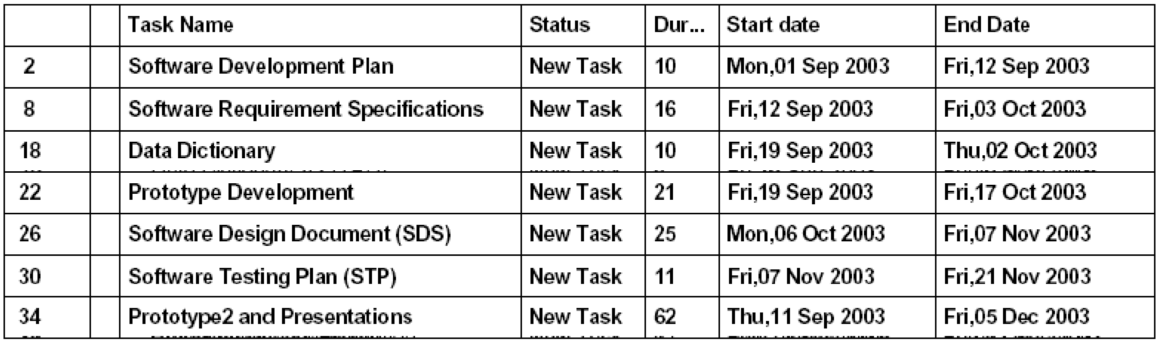
Over the course of the project, several different documents, charts, programs, and other project management and software development tools will be delivered. The following table is a list of deliverables for the project.

Deliverable	Detail
Software Development Plan	The Software Development Plan describes the process that the project is planning to exercise to meet the various project goals.
Risk Assessment Document	This document will describe technical and non-technical risks formally identified by the project.
Prototype	The prototype will help define how the solution will do what it is it will be doing. If a user interface is to be implemented, the prototype will help define the style and usability of the user interface.
Design Plan	This document will describe the proposed technical design of the solution being implemented.
Test Plan	The test plan will define test scenarios and test scripts to appropriately test the implemented solution to meet the desired user acceptance level.
Data Dictionary	The data dictionary will define the data elements used in the implemented solution

Table 2: Deliverables

1.1.4 Schedule and Budget Summary

The table below is a list of major project tasks and the project dates associated with those tasks. This table was generated from the project management software used to manage our project tasks.



		Task Name	Status	Dur...	Start date	End Date
2		Software Development Plan	New Task	10	Mon,01 Sep 2003	Fri,12 Sep 2003
8		Software Requirement Specifications	New Task	16	Fri,12 Sep 2003	Fri,03 Oct 2003
18		Data Dictionary	New Task	10	Fri,19 Sep 2003	Thu,02 Oct 2003
22		Prototype Development	New Task	21	Fri,19 Sep 2003	Fri,17 Oct 2003
26		Software Design Document (SDS)	New Task	25	Mon,06 Oct 2003	Fri,07 Nov 2003
30		Software Testing Plan (STP)	New Task	11	Fri,07 Nov 2003	Fri,21 Nov 2003
34		Prototype2 and Presentations	New Task	62	Thu,11 Sep 2003	Fri,05 Dec 2003

Table 3: Major Project Tasks

1.2 Evolution of the Plan

Updates to the Software Development Plan shall be communicated at the regularly scheduled team meetings. The updates identified at the team meetings will be committed to the project plan by the owner of the update.

In certain situations, committing the update to the project plan may be the responsibility of someone other than the person who identified the update. For example, the project manager is responsible for creating and managing the various project management charts used to facilitate the management of the project. If an update to the project plan is communicated that impacts the various management charts, it is the responsibility of the project manager to reflect the update to the project plan on the various management charts.

Any update made to the project plan that was not formally communicated at the team meetings is considered unscheduled. Any unscheduled update will be expected to be communicated in an adequate fashion so that all team members are aware of the unscheduled update made to the project plan.

It is the responsibility of the project leader to place the Software Development Plan under configuration management and to control any subsequent major revisions to the Software Development Plan.

2 References

Reference	Detail
http://suberic.net/pooka/	This is a reference for the Pooka Email Client/Server software that is being used to develop the electronic stamp software.

Table 4: References

3 Project Organization

3.1 External Interfaces

The clients to the Electronic Stamp project are the Taltech Alliance and the FSU Technology Transfer Office. These organizations will be reviewing the electronic stamp solution provided by Seminole Software. These organizations will not be made available to the project team.

The requirements expert for the electronic stamp project is Dr. Baker at Florida State University. Dr. Baker has identified the initial set of requirements for the electronic stamp project and is fielding any questions regarding the electronic stamp requirements.

3.2 Internal Structure

The Seminole Software team is made up of a project manager, project leader, and a technical leader. Figure 1 shows the internal organization and how communication is performed between the different team members. The team takes the formation shown in the External Team Roles when dealing with any external interface. For example, when Seminole Software will present the solution to the review committee, the team will be working within the External Team Roles. For all internal work, the work effort flows evenly between us, which is depicted by the Internal Work Collaboration in Figure 1. The team has chosen this type of internal structure due to the small team size. It is simply not feasible for us to stay exactly in our team role considering all of the work to be done.

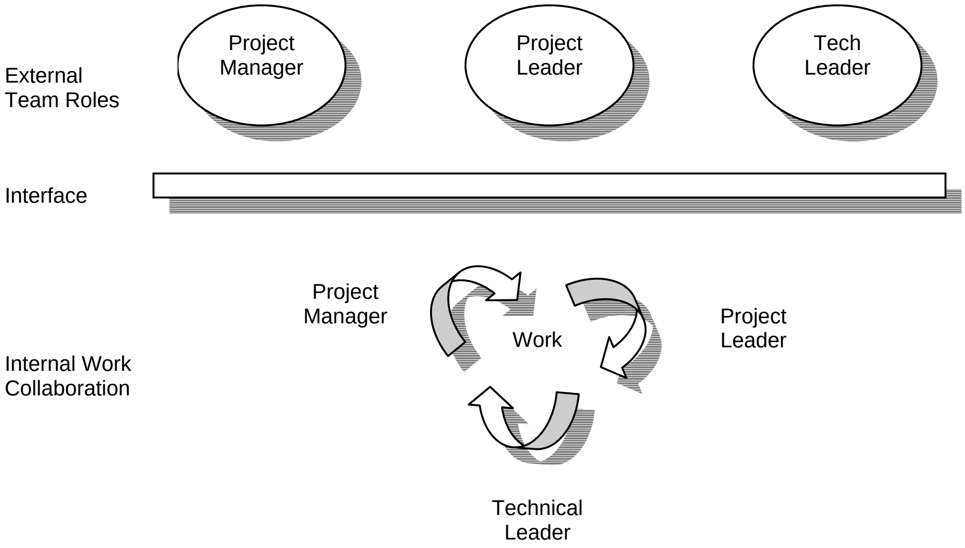


Figure 1: Internal Team Organization

The project manager is responsible for scheduling and holding team meetings and the minutes of those meetings. The project manager is also responsible for the organization of the team and communicating any issues to the client and requirements expert. Lastly, the project manager is responsible for creating and managing the actual project management plan and the various project management charts that are used to facilitate the management of the project.

The project leader is responsible for the compiling original project deliverables accomplished by all team members. The project leader is also responsible for directing the creation and completion of the project deliverables.

The technical leader is responsible for leading the technical effort of the project including the software development team. The technical team leader is also responsible for expert knowledge of the technologies used to construct and deploy the software solution.

3.3 Roles and Responsibilities

The following table lists the major project tasks and the team roles that are responsible for that task.

Project Task	Responsible Role	Detail
Project Plan	Project Manager	The creation and management of the project plan. This includes the various project charts created from the project plan.
Software Development Plan	Project Leader, Entire Team	The creation and management of the project plan documentation. The project leader is ultimately responsible for the project plan but the entire team contributes to this effort.
Risk Assessment	Project Manager	The identification, compilation, and reporting project risks.
Prototype	Technical Leader	Design and construction of the prototype.
Design Document	Project Leader, Technical Leader	The creation of the design and documentation of the design that will be used to build the software solution.
Test Plan	Project Leader, Entire Team	The creation of the test scenarios that will be used to test the performance and acceptability of the software solution.

Table 5: Roles and Responsibilities

4 Managerial Process Plans

4.1 Start-Up Plan

4.1.1 Estimation Plan

The Electronic Stamp Project is being completed within the timeframe of a semester at Florida State University. Because of this, the ultimate project deliverable deadline is hard and fast at the end of the semester. The intermediary project deadlines have been set to reflect this end of semester project deadline.

4.1.2 Staffing Plan

The project is staffed by a team assembled almost randomly at the start of the semester. Our staff for the project is limited to three people. Together, the project team has identified a Project Manager, a Project Leader, and a Technical Leader. All other roles that will need to be present at different times over the course of the project will be filled appropriately by one of the three team members. The team resources are static; they will not change over the course of the project.

4.1.3 Resource Acquisition Plan

The following table lists the resources that need to be acquired and by what project milestone they need to be acquired by to be able to complete the project.

Resource	Needed by Milestone	Detail
Project Management Software	Pre-Software Development Plan Milestone	Project management software will be needed to help manage the resources and tasks involved with the project.
Email Client/Server Software	Pre-Prototype Milestone	The email client and server software that will be used for the development of the electronic stamp.

Table 6: Resource Acquisition Plan

4.1.4 Project Staff Training Plan

The following table lists the staff training that will need to take place and by what project milestone the training needs to be completed by to be able to complete the project.

Training	Needed by Milestone	Detail
Project Management Software	Pre-Software Development Plan Milestone	The Project Manager needs to know how to use the project management software to be able to produce the project plan and various project management charts.

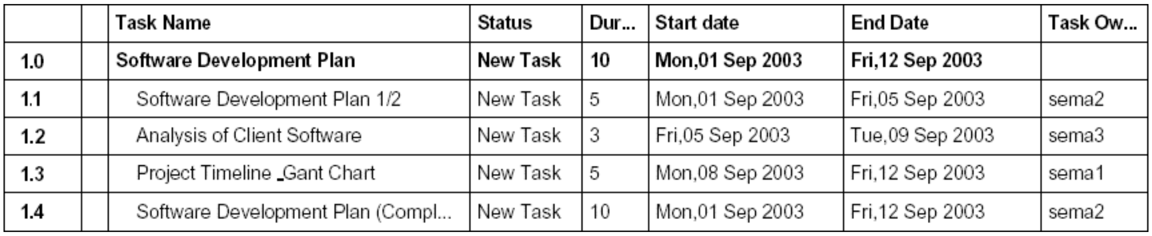
Training	Needed by Milestone	Detail
Email Client/Server Software	Pre-Prototype Milestone for Technical Leader. During Prototype Milestone for rest of team.	The Technical Leader needs to develop an intimate knowledge with the email software that will be used to develop the electronic stamp solution. The entire team needs to develop at least a working knowledge with the email software.
Java Software Libraries	Pre-Prototype Milestone for Technical Leader.	The development team needs to become familiar with the components of the Java libraries and any other software libraries needed to develop the electronic stamp solution.
SMTP	Pre-Prototype Milestone for Technical Leader.	The Technical Leader needs to develop an intimate knowledge with SMTP to be able to develop the electronic stamp solution.

Table 7: Staff Training Plan

4.2 Work Plan

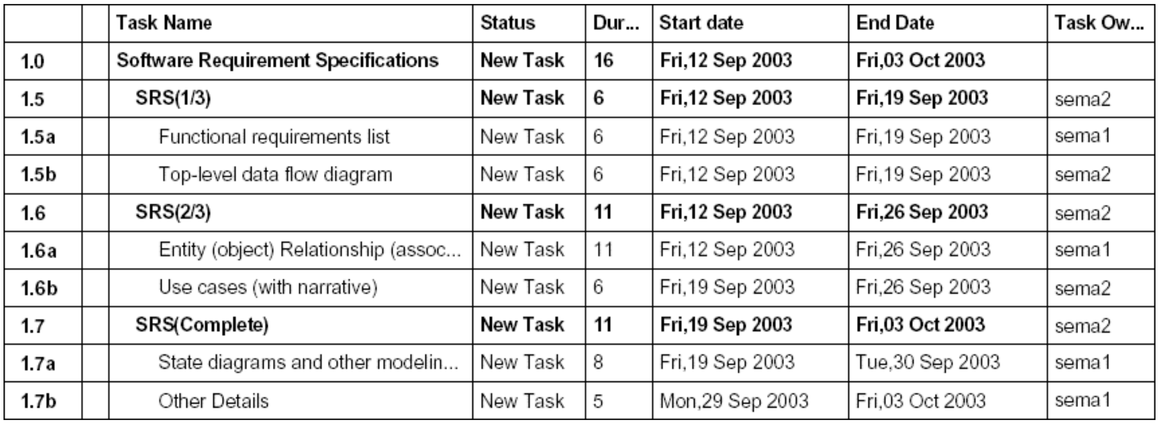
For the following Work Plan subsections please refer to the tables below. The tables are taken from the project management tool the project team used to develop the project plan. The task numbers have been updated to reflect the Iterative Prototyping Model described in Section 5.1, which has been employed as the process model for this project. Please refer to that section to better understand the task resource planning tables below.

The entire project plan, including a Gantt chart that graphically depicts our project plan, can be found in Appendix A.



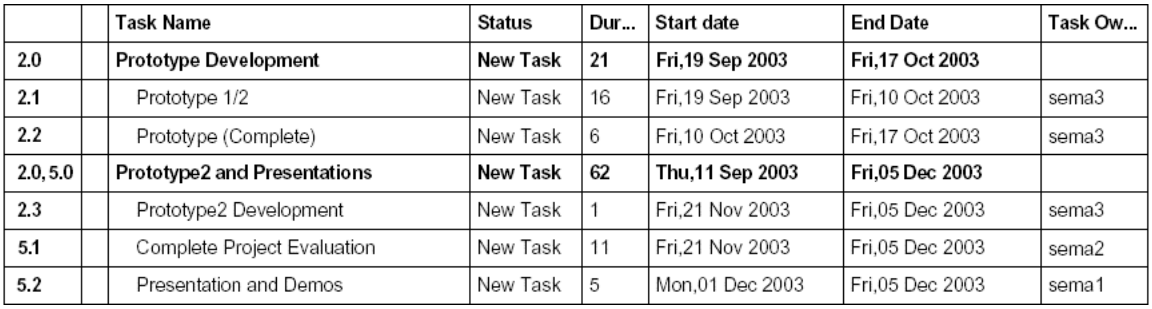
		Task Name	Status	Dur...	Start date	End Date	Task Ow...
1.0		**Software Development Plan**	**New Task**	**10**	**Mon,01 Sep 2003**	**Fri,12 Sep 2003**	
1.1		Software Development Plan 1/2	New Task	5	Mon,01 Sep 2003	Fri,05 Sep 2003	sema2
1.2		Analysis of Client Software	New Task	3	Fri,05 Sep 2003	Tue,09 Sep 2003	sema3
1.3		Project Timeline _Gant Chart	New Task	5	Mon,08 Sep 2003	Fri,12 Sep 2003	sema1
1.4		Software Development Plan (Compl...	New Task	10	Mon,01 Sep 2003	Fri,12 Sep 2003	sema2

Table 8: Software Development Plan Project Plan Table



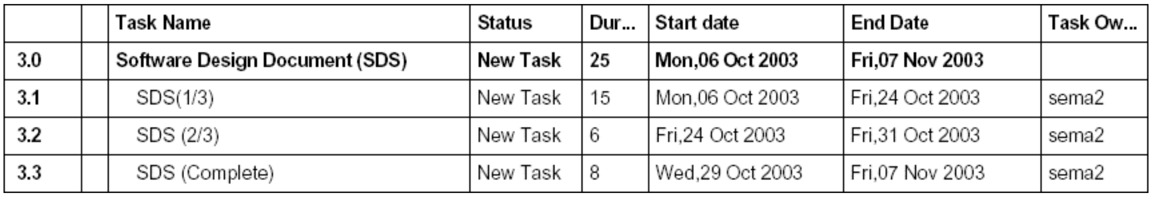
		Task Name	Status	Dur...	Start date	End Date	Task Ow...
1.0		**Software Requirement Specifications**	**New Task**	**16**	**Fri,12 Sep 2003**	**Fri,03 Oct 2003**	
1.5		**SRS(1/3)**	**New Task**	**6**	**Fri,12 Sep 2003**	**Fri,19 Sep 2003**	sema2
1.5a		Functional requirements list	New Task	6	Fri,12 Sep 2003	Fri,19 Sep 2003	sema1
1.5b		Top-level data flow diagram	New Task	6	Fri,12 Sep 2003	Fri,19 Sep 2003	sema2
1.6		**SRS(2/3)**	**New Task**	**11**	**Fri,12 Sep 2003**	**Fri,26 Sep 2003**	sema2
1.6a		Entity (object) Relationship (assoc...	New Task	11	Fri,12 Sep 2003	Fri,26 Sep 2003	sema1
1.6b		Use cases (with narrative)	New Task	6	Fri,19 Sep 2003	Fri,26 Sep 2003	sema2
1.7		**SRS(Complete)**	**New Task**	**11**	**Fri,19 Sep 2003**	**Fri,03 Oct 2003**	sema2
1.7a		State diagrams and other modelin...	New Task	8	Fri,19 Sep 2003	Tue,30 Sep 2003	sema1
1.7b		Other Details	New Task	5	Mon,29 Sep 2003	Fri,03 Oct 2003	sema1

Table 9: Software Requirements Specifications Project Plan Table



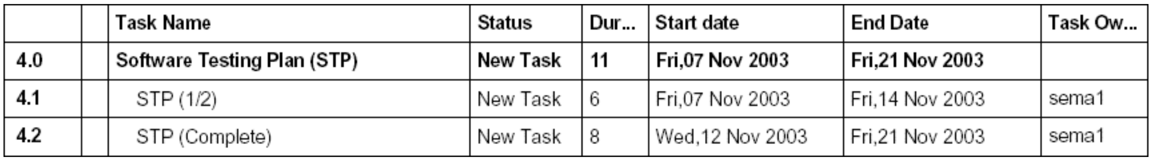
		Task Name	Status	Dur...	Start date	End Date	Task Ow...
2.0		**Prototype Development**	**New Task**	**21**	**Fri,19 Sep 2003**	**Fri,17 Oct 2003**	
2.1		Prototype 1/2	New Task	16	Fri,19 Sep 2003	Fri,10 Oct 2003	sema3
2.2		Prototype (Complete)	New Task	6	Fri,10 Oct 2003	Fri,17 Oct 2003	sema3
2.0, 5.0		**Prototype2 and Presentations**	**New Task**	**62**	**Thu,11 Sep 2003**	**Fri,05 Dec 2003**	
2.3		Prototype2 Development	New Task	1	Fri,21 Nov 2003	Fri,05 Dec 2003	sema3
5.1		Complete Project Evaluation	New Task	11	Fri,21 Nov 2003	Fri,05 Dec 2003	sema2
5.2		Presentation and Demos	New Task	5	Mon,01 Dec 2003	Fri,05 Dec 2003	sema1

Table 10: Prototype Project Plan Table



		Task Name	Status	Dur...	Start date	End Date	Task Ow...
3.0		**Software Design Document (SDS)**	**New Task**	**25**	**Mon,06 Oct 2003**	**Fri,07 Nov 2003**	
3.1		SDS(1/3)	New Task	15	Mon,06 Oct 2003	Fri,24 Oct 2003	sema2
3.2		SDS (2/3)	New Task	6	Fri,24 Oct 2003	Fri,31 Oct 2003	sema2
3.3		SDS (Complete)	New Task	8	Wed,29 Oct 2003	Fri,07 Nov 2003	sema2

Table 11: Software Design Document Project Plan Table



		Task Name	Status	Dur...	Start date	End Date	Task Ow...
4.0		**Software Testing Plan (STP)**	**New Task**	**11**	**Fri,07 Nov 2003**	**Fri,21 Nov 2003**	
4.1		STP (1/2)	New Task	6	Fri,07 Nov 2003	Fri,14 Nov 2003	sema1
4.2		STP (Complete)	New Task	8	Wed,12 Nov 2003	Fri,21 Nov 2003	sema1

Table 12: Software Testing Plan Project Plan Table

Table 8 shows the start and end dates along with the resource task owners for the major task of Software Development Plan.

Table 9 shows the start and end dates along with the resource task owners for the major task of Software Requirement Specification.

Table 10 shows the start and end dates along with the resource task owners for the major task of Building the Prototype. The Project Evaluation and Client Presentations have also been included in this table due to the proximity of deliverable dates to the completion of the prototype.

Table 11 shows the start and end dates along with the resource task owners for the major task of Software Design Document.

Table 12 shows the start and end dates along with the resource task owners for the major task of Software Test Plan.

4.2.1 Work Activities

The work activities for the project are shown in the project management tables in the second column labeled “Task Name”. The tables also show via the task id number what part of the Iterative Prototyping Model the work tasks are a part of. The next section shows in more detail where each task fits into this model. Refer to section 5.1 for an explanation of the Iterative Prototyping Model

4.2.2 Schedule Allocation

The following task network charts show the dependencies between work tasks. Figures 2 through 7 show the detail task network diagrams for each of the major tasks found in the Iterative Prototyping Model as it pertains to the Electronic Stamp Project.

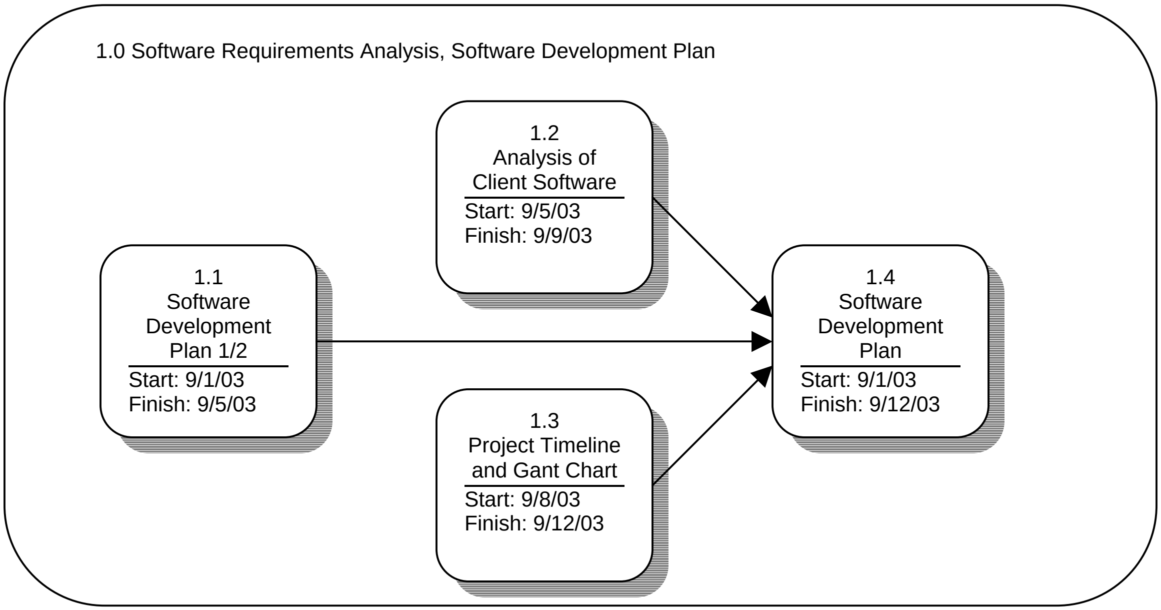


Figure 2: Software Development Plan Task Network

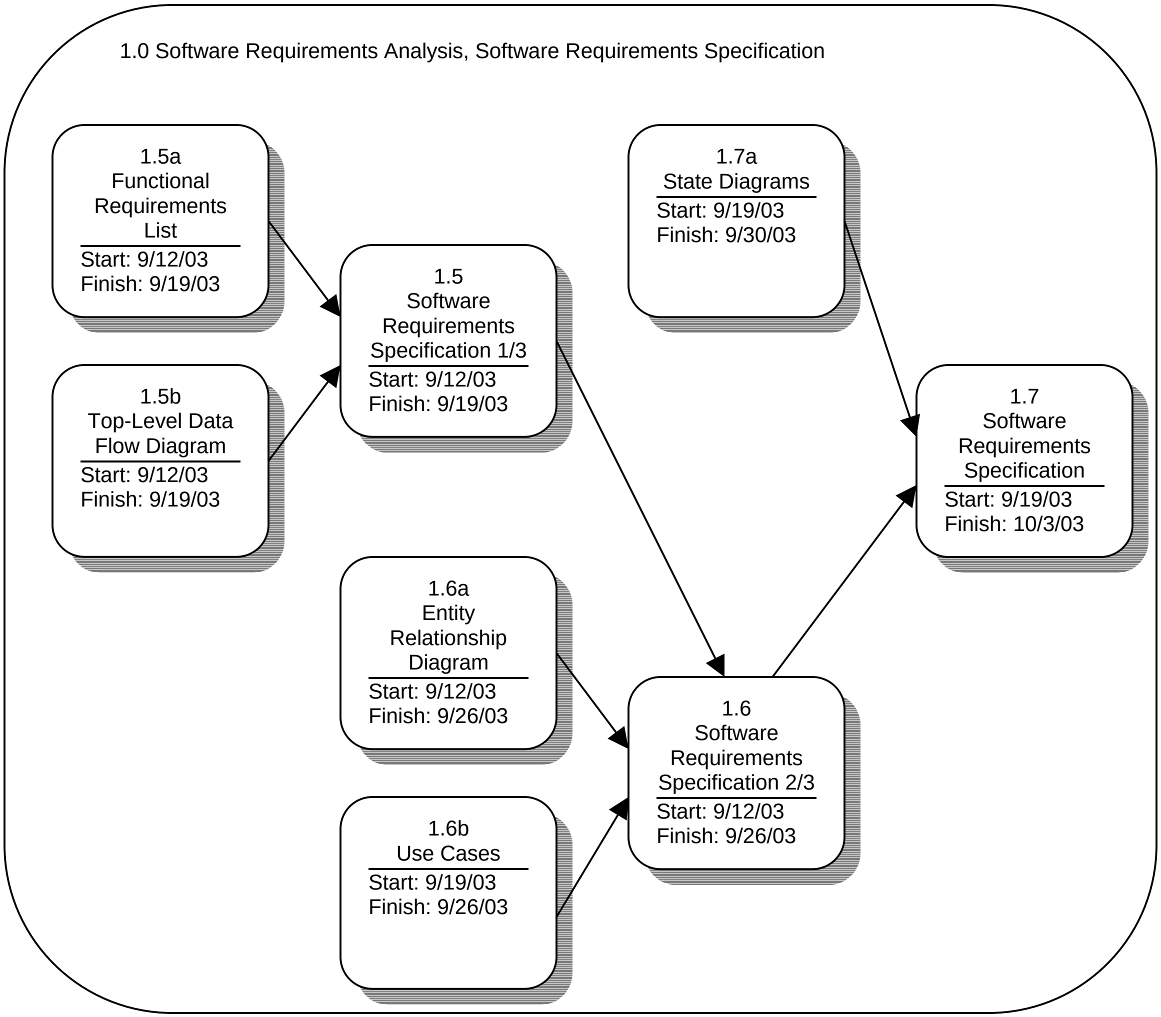

1.0 Software Requirements Analysis, Software Requirements Specification
1.5a Functional Requirements List
Start: 9/12/03
Finish: 9/19/03
1.5b Top-Level Data Flow Diagram
Start: 9/12/03
Finish: 9/19/03
1.5 Software Requirements Specification 1/3
Start: 9/12/03
Finish: 9/19/03
1.7a State Diagrams
Start: 9/19/03
Finish: 9/30/03
1.7 Software Requirements Specification
Start: 9/19/03
Finish: 10/3/03
1.6a Entity Relationship Diagram
Start: 9/12/03
Finish: 9/26/03
1.6b Use Cases
Start: 9/19/03
Finish: 9/26/03
1.6 Software Requirements Specification 2/3
Start: 9/12/03
Finish: 9/26/03

Figure 3: Software Requirements Specification Task Network

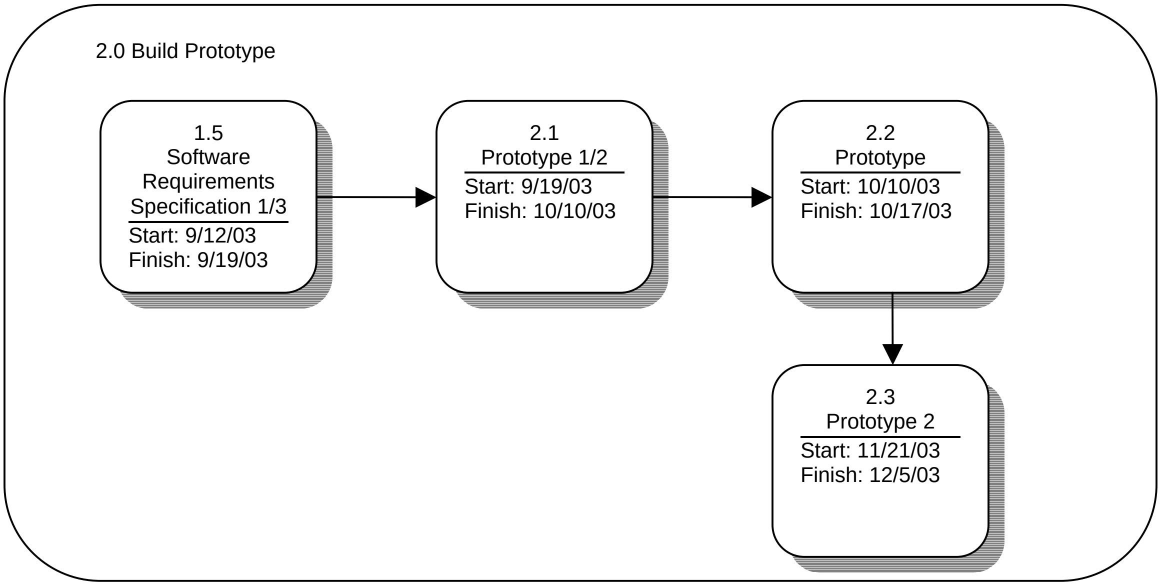


Figure 4: Build Prototype Task Network

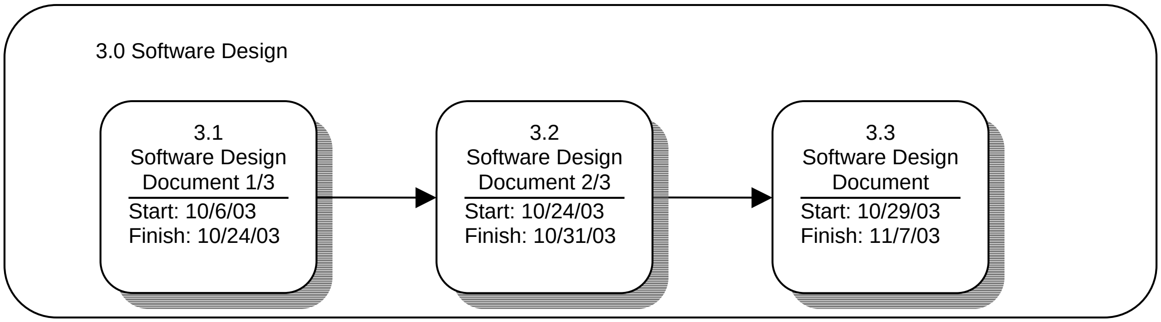


Figure 5: Software Design Task Network

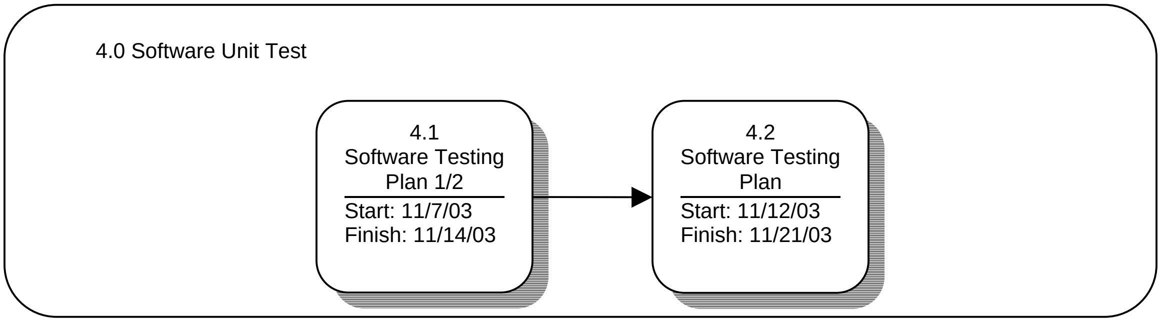


Figure 6: Software Unit Test Task Network

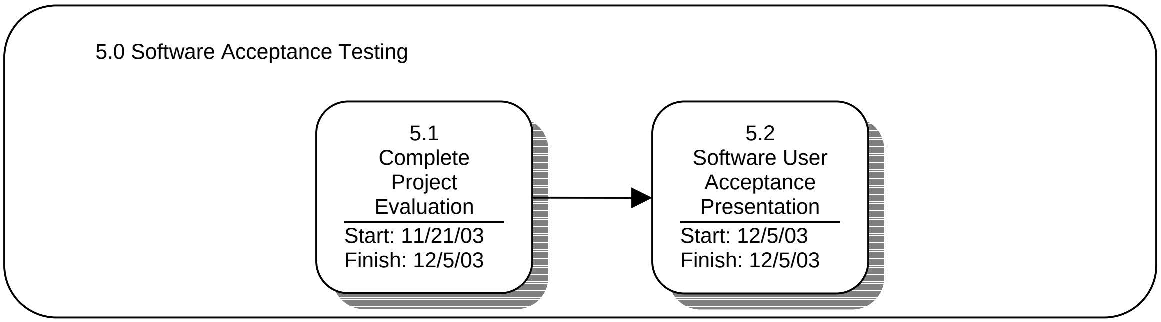

5.0 Software Acceptance Testing
5.1
Complete Project Evaluation
Start: 11/21/03
Finish: 12/5/03
5.2
Software User Acceptance Presentation
Start: 12/5/03
Finish: 12/5/03

Figure 7: Software Acceptance Testing Task Network

4.2.3 Resource Allocation

The project management tables show the resources identified to own each task in the last column labeled “Task Ow…” Within the project management software, each role has been given a token identifier. The sema1 resource identifier is the Project Manager, the sema2 resource identifier is the Project Leader, and lastly the sema3 resource identifier is the Technical Leader. The project resource allocation reflects each person on the project team committing at least 2 hours per day to the project.

4.3 Control Plan

For many of the control plans identified in the follow Control Plan subsections, the project team feels that the regularly scheduled team meetings are an adequate vehicle for communication and control. The project team has drawn this conclusion because of the small size of our project team. In fact, it could potentially be anti-productive to implement sophisticated Requirements, Schedule, and Quality control plans.

4.3.1 Requirements Control Plan

The requirements for the project will be identified and documented in the Software Requirements Specification. Any changes to requirements that need to be reflected in the Software Requirements Specification should be communicated at the regularly scheduled team meetings. It should be noted that it has been communicated to the project team that the requirements will not be changing.

4.3.2 Scheduled Control Plan

It is the responsibility of the Project Manager to insure that the schedule is under control. Any problems with project progress will be managed by the Project Manager. Any project schedule issue will be communicated at the regularly scheduled team meetings.

4.3.3 Quality Control Plan

Most, if not all, of the quality control will be done by successfully meeting the criteria identified in Section 5.3 Product Acceptance Plan. During product development, quality control will be done through properly unit and regression testing all functional areas of code. Also, milestone review meetings will allow the project team to analyze and possibly improve project deliverables.

4.4 Risk Management Plan

The identification of risk to the project is done through the Risk Assessments performed throughout the course of the project. The project risks are identified at the project team meetings and captured on the Risk Assessment documents by the project manager. The Risk Assessments are communicated to the requirements expert by the team project manager.

5 Technical Process Plan

5.1 Process Model

An Iterative Prototyping Model is being used by the Seminole Software team to develop the Electronic Stamp software. Figure 8 shows the major tasks and task dependencies associated with an Iterative Prototyping Model. Figure 8 also shows the deliverables produce by each major task.

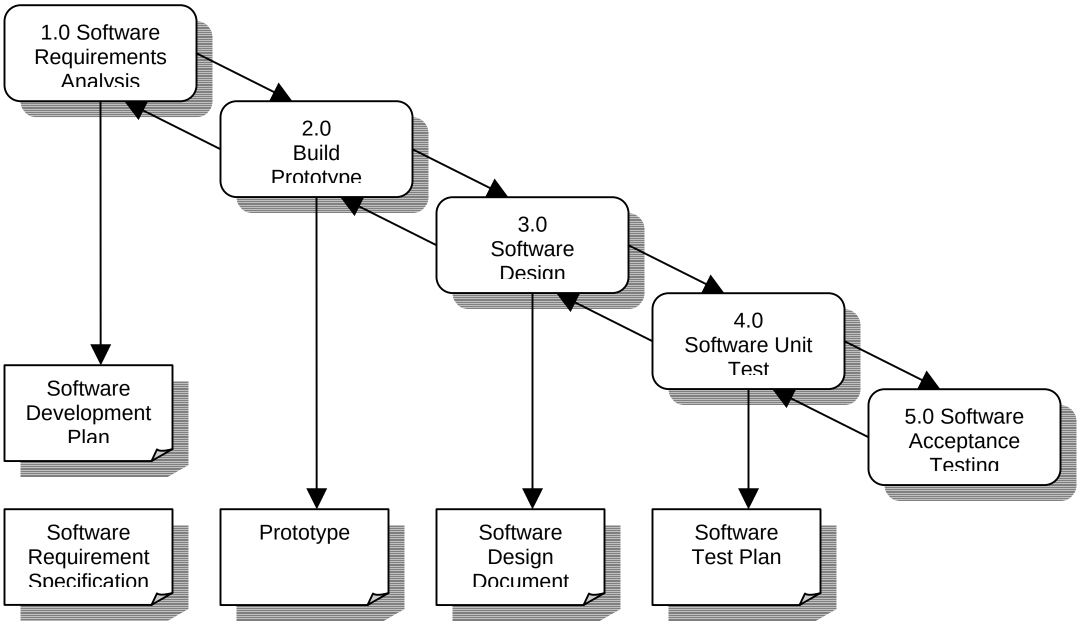

1.0 Software Requirements Analysis
2.0 Build Prototype
3.0 Software Design
4.0 Software Unit Test
5.0 Software Acceptance Testing
Software Development Plan
Software Requirement Specification
Prototype
Software Design Document
Software Test Plan

Figure 8: Iterative Prototyping Model

5.2 Methods, Tools, and Techniques

The project team will be using an object oriented design and object oriented programming model for implementing the electronic stamp project. This is in part due to the object oriented nature of the Java programming language, which the chosen email server and email client are implemented in.

The development effort will be using the Pooka email client/server software to implement the electronic stamp software. Where possible, the project team will be following the standard of development implemented in the Pooka software.

5.3 Product Acceptance Plan

The acceptance criteria for the Electronic Stamp software are listed in the following tables. The first table lists the generic software acceptance criteria and the second table lists the specific software acceptance criteria.

Acceptance Criterion	Acceptance Test	Detail
Software Usability	The customer acceptance testing.	Has the project team delivered a user interface that meets the design software design requirements?
Software Performance	The customer acceptance testing.	Does the software perform with acceptable performance? For example, does the software have long processing delays? Does the software accurately meet the defined design requirements?
Software Platform	The customer acceptance testing.	Does the software execute on the hardware that was specified in the software requirements?
Software Maintenance	The customer acceptance testing.	Does the software execute at an acceptable length of time without needing maintenance?

Table 13: Generic Software Acceptance Criteria

Acceptance Criterion	Acceptance Test	Detail
Does the electronic stamp vendor provide a unique stamp for every request?	Unit testing and customer acceptance testing.	The electronic stamp vendor should provide a unique stamp for every request.
Does the electronic stamp inhibit unsolicited email that does not have a properly addressed electronic stamp?	Unit testing and customer acceptance testing.	Unsolicited email without a properly purchased stamp that is properly addressed should not be received.
Does the client user have the ability to block unwanted email with an electronic stamp?	Unit testing and customer acceptance testing.	The client needs to be able to block email from a known source even if they have an electronic stamp.
Does the sender have the ability to log the electronic stamp as “Use for Return”?	Unit testing and customer acceptance testing.	The sender may want to allow the receiver to send a response message on the same stamp.

Acceptance Criterion	Acceptance Test	Detail
Does the receiver have the ability to log the electronic stamp as “Multi Use”?	Unit testing and customer acceptance testing.	The receiver may want to allow the sender to send additional responses to the original email.
Does the client user have the ability to send a reject return message to email that the client user has not accepted?	Unit testing and customer acceptance testing.	The client should be able to send a message to a sender when that client user rejects or does not accept a message.

Table 14: Specific Software Acceptance Criteria

6 Supporting Process Plans

6.1 Configuration Management Plan

The project team will be using CVS for source configuration management. The audited change control schedule identified in the table below will be put into practice during prototype development.

Development Milestone	Change Control	Detail
Pre-Prototype Release 1	No Audited Change Control	The development team may check code into source configuration management without audited change control. Additional changes can be made to those files without audited change control.
Prototype Release 1	Loose Audited Change Control	After the first release of the prototype, the development team will use a loose change control method.
Functional Code Freeze	Strict Audited Change Control	Once all of the functionality has been implemented in the prototype that needs to be delivered to the customer, any changes made to the source code must be reviewed by the project team before the changes can be checked into the base source code.

Table 15: Change Control Schedule

No Audited Change Control allows the development team to work freely without delay of project team approval. This is most important during the initial development stages of the prototype.

Loose Audited Change Control will require the development team to identify the changes being made to the prototype source code to the project team but do not need approval.

Strict Audited Change Control will require the development team to identify the changes being made to the prototype source code to the project team along with project team approval.

Upon every major release of the prototype, the source code will be branched in source configuration management so that the state of the successfully united test release can be preserved. Any major bugs that are identified in subsequent releases will be merged back into the previous source branches upon project team approval.

6.2 Verification and Validation Plan

Most of the verification of work activities will take place at the regularly scheduled team meetings. Special milestone verification reviews will take place at additionally scheduled milestone meetings in which the milestone deliverables will be verified for completeness. It may be the case that the milestone meetings take place as a special agenda item at the regularly scheduled team meetings.

There is not much of a degree of independence between development activities and the verification activities to verify the development effort. This is due to the limited number of resources available to the project. Mainly, the three members of the project team will be building the software and verifying its correctness.

The validation of the software will be done through successful completion of unit tests for each new identifiable function of code introduced into the email server and client software. Regression testing will take place on previously implemented functions of code for every major software milestone.

6.3 Documentation Plan

The following table identifies the non-deliverable and deliverable documents to be produce by the project team.

Document	Type	Detail
Software Development Plan	Non-deliverable	This document describes the general management plan and project process plans.
Software Design Document	Non-deliverable	This document describes the software design for the electronic stamp software.
Software Test Plan	Non-Deliverable	This document describes the test plan that will be used to test the electronic stamp software.
Meeting Minutes	Non-Deliverable	The meeting minutes account the details of the team meetings that take place over the course of the project.
Risk Assessments	Non-Deliverable	The risk assessments will be used to identify areas of concern that need to be addressed over the course of the project.

Table 16: Project Documents

6.4 Quality Assurance Plan

Quality assurance will be provided throughout the project by software development policy and product software milestone testing. It is the project team's software development policy that developed functional code will be thoroughly unit tested at the individual function level. Along with unit testing, the software will be holistically tested from end to end for every product milestone, verifying correctness according to the software requirements and software design.

6.5 Reviews and Audits

Intermediary project reviews to assess quality and audit completion toward project deliverables will take place at the specially schedule milestone review meetings. As identified in section 6.2 Verification and Validation, the milestone review meetings will coincide with major milestones, before and after major milestone completion. The milestone review meetings will verify and validate project correctness, review project quality, and audit completion toward deliverables.

A final review and audit of project deliverables will be conducted by the customer and requirements expert at the end of the project upon delivery of the completed software product.

6.6 Problem Resolution Plan

All problems should be made known at the regularly scheduled team meetings by the person who has identified the potential problem. The project manager is responsible for documenting the problem and managing the problem to resolution. This means that the project manager will need to get the proper people involved to resolve the problem; this includes external resources as well as internal resources.

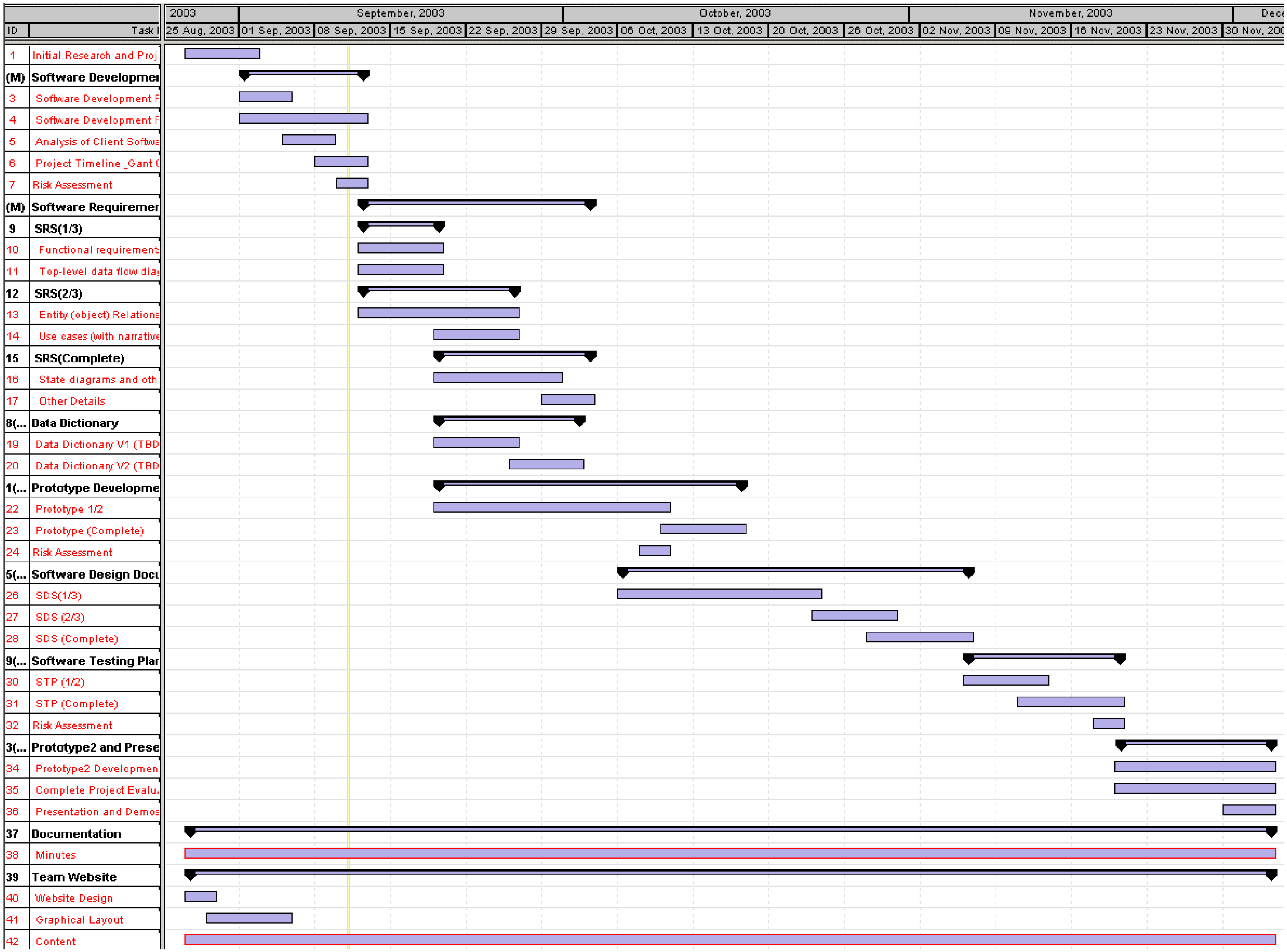
In the event that a problem has been identified that is an emergency and cannot wait to be resolved until the next scheduled meeting, the problem identifier will contact the project manager directly via phone or email and make the emergency problem known. It is then the responsibility of the project manager to manage the emergency problem to resolution. This may require that an unscheduled emergency meeting be called to facilitate the effort to resolve the emergency problem.

Specific problems found in the product software during software testing, also known as "bugs", will be documented and managed by the technical leader. All identified software bugs will be entered into a spreadsheet that will be kept in source configuration management.

The spreadsheet of bugs will identify when the bug was identified, who identified it, how to recreate the bug, what version of the software the bug was found in, the status of being fixed, and who is assigned to fix the bug. It is the responsibility of the technical leader to make sure that all bugs are fixed by the appropriate software milestone. The person who identified the bug can either communicate the found bug to the technical leader or enter the bug directly into the spreadsheet.

The technical leader will be responsible for giving a bug status report at every regularly scheduled team meeting.

Appendix A – Full Project Plan



ID	Task I
1	Initial Research and Proj
(M)	**Software Developmer**
3	Software Development F
4	Software Development F
5	Analysis of Client Softwa
6	Project Timeline _Gant (
7	Risk Assessment
(M)	**Software Requiremer**
9	**SRS(1/3)**
10	Functional requirement
11	Top-level data flow dia
12	**SRS(2/3)**
13	Entity (object) Relations
14	Use cases (with narrative
15	**SRS(Complete)**
16	State diagrams and oth
17	Other Details
8(...	**Data Dictionary**
19	Data Dictionary V1 (TBD
20	Data Dictionary V2 (TBD
1(...	**Prototype Developme**
22	Prototype 1/2
23	Prototype (Complete)
24	Risk Assessment
5(...	**Software Design Docu**
26	SDS(1/3)
27	SDS (2/3)
28	SDS (Complete)
9(...	**Software Testing Plar**
30	STP (1/2)
31	STP (Complete)
32	Risk Assessment
3(...	**Prototype2 and Prese**
34	Prototype2 Developmen
35	Complete Project Evalu.
36	Presentation and Demos
37	**Documentation**
38	Minutes
39	**Team Website**
40	Website Design
41	Graphical Layout
42	Content