

Software Project Management Plan

For the Real Estate Association of Cascadia

Kal Toth

V1.0, April 23, 2009

[This sample SPMP is from a past case study that is similar to your current case. Not that this customer is in the real estate industry. Kal Toth]

Description	Author	Version	Date
Initial Draft (sections 1-4 and 6 only)	Kal Toth	V1.0	04/23/09

Table of Contents

1	<i>Project Overview and Summary</i>	3
1.1	Purpose	3
1.1.1	Context	3
1.1.2	Scope of Requirements	3
1.1.3	Project Objectives and Priorities	4
1.1.4	Flexibility Matrix	4
1.2	Assumptions, Dependencies and Constraints	4
1.2.1	Assumptions	4
1.2.2	Dependencies	5
1.2.3	Constraints	5
1.3	Project Deliverables	5
1.4	Summary of Schedule and Budget	5
1.5	Evolution of the SPMP	5
2	<i>References</i>	6
3	<i>Definitions and Acronyms</i>	6
4	<i>Project Team Organization</i>	7
4.1	Organization Chart: Internal and External Interfaces	7
4.2	Project Roles and Responsibilities	8
5	<i>Managerial Process</i>	8
6	<i>Technical Processes</i>	9
6.1	Process Model	9
6.1.1	Development Iterations (a.k.a. "cycles")	9
6.1.2	Milestones and Delivery Sequence	10
6.2	Methods, Tools and Techniques	12
6.2.1	Software Development Methods, Tools and Techniques	12
6.2.2	Technical Specifications	12
6.2.3	User Documentation	12
6.3	Development Environment and Infrastructure	12
6.4	Acceptance, Training, and Deployment Processes	13
7	<i>Support Processes</i>	13
8	<i>Additional Supporting Plans</i>	13

1 Project Overview and Summary

1.1 Purpose

JDI has entered into a contract to provide Cascadia a single integrated system enabling Cascadia employees to quickly and easily access current and detailed real estate information in support of their business. A second aspect of this project is to establish a software engineering competency within JDI in the mobile applications field.

1.1.1 Context

Cascadia is an association of municipal real estate boards, agencies and their member real estate agents across the states of Oregon, Idaho, Washington and the Province of British Columbia. There are about 16,000 real estate agents, 50 real estate companies and about 150 agencies (real estate offices) across Cascadia.

Cascadia employees currently rely upon several localized groups within Cascadia that have independently developed their own systems, however these system where not made with a common architecture and do not share information.

This project will build and deploy a common system for all Cascadia operations and track use for appropriate billing. The system will be web enabled and support cross-platform messaging, calendaring and forms for highly mobile real estate agents using PCs, cell phones and PDAs (Personal Digital Assistants).

The system is to provide a central cluster process in Portland, area processors in each state (Oregon, Washington, Idaho and British Columbia), and 25 district processors.

The contact with Cascadia is fixed price with a duration of 18 months. Work began January 2nd and is to be completed by July 1st of the following year.

1.1.2 Scope of Requirements

The customer will ensure that user representatives are available to participate in the requirements elicitation and specification activities called up in the SPMP. Further, the customer and user representative will have the opportunity to review prototypes of the user interfaces (UIs) to arrive at suitable UIs early in the development process.

The functional capabilities to be satisfied by the system and its software are taken directly from the project terms of reference and summarized as follows:

- Authentication
 - Accounts and Logon
 - Personal Profiles and Preferences
- Primary Application: Real Estate
 - Database / Repository
 - Posting New Listings
 - On-Demand Real Estate Searches
 - Real Time Notifications (matching listings)
- Online Internet Access
- Cross-Platform Messaging
- Calendaring
- Scheduling
- Appointment Reminders
- Forms Management

Performance, reliability, availability, and security requirements are also to be satisfied.

1.1.3 Project Objectives and Priorities

JDI shall provide Cascadia a system that meets their requirements while concurrently developing a core multi-platform capability that may be applied to future JDI projects. It is JDI's intent to establish a good working relationship with the customer to ensure effective communication, which is fundamental to the requirements gathering and project control aspects of this project.

To ensure success the development staff will be provided the necessary training, tools and environment to effectively meet their development deadlines without unnecessary overtime.

Different aspects of the system are subject to varying degrees of flexibility or constraints. The initial delivery date and the project cost are fixed. The calendaring capabilities will have some constraints depending upon the COTS product implemented, however these constraints must not prevent adherence to customer requirements. The remaining areas of requirements can be categorized as either fixed or flexible depending on how these requirements support Cascadia's business model and work-flow. For example, the degree to which cross-platform capabilities will be implemented has some flexibility. The following Flexibility Matrix will be reviewed, updated and agreed to during project mobilization.

1.1.4 Flexibility Matrix

Dimension	Fixed	Constrained	Flexible
Delivery Schedule	X		
Project Cost	X		
Requirements: Authentication	X		
Requirements: Real Estate Domain		X	
Requirements: Internet Access	X		
Requirements: Calendaring		X	
Requirements: Cross-Platform			X
Requirements: Forms			X

1.2 Assumptions, Dependencies and Constraints

Fundamental to the project is the understanding that Cascadia will not accept delivery of a system that does not meet their nominal requirements. These nominal requirements include following:

- Identity
- Internet Access
- Core: MLS DB and New Posting Entry
- Calendaring: Scheduling and schedule based notifications
- Cross Platform: Cell phone based notification
- Forms: Sales Agreement, Earnest Agreement

Additionally, the project will proceed with the following assumptions, dependencies and constraints.

1.2.1 Assumptions

- JDI can provide funding for any required training. This funding is not part of the budgeted contract.
- Contract with Cascadia is agreed upon and payments made per schedule.
- JDI can provide temporary assistance, as needed, in the form of use case experts to facilitate the requirements process.

1.2.2 Dependencies

- Cascadia must provide sufficient internal resources to clearly define system requirements, evaluate prototypes, and participate in validation testing.

1.2.3 Constraints

- Programming staff will be comprised entirely of JDI internal employees with the exception of 4 programmers provided by Cascadia.
- Multi-platform capability to be limited to Palm Pilot, WinTel based laptops, text pagers and cell phones supporting JVM 1.4.2 or later.

1.3 Project Deliverables

This section describes deliverables expected by the customer in support of the system. Each of these deliverables will be provided to the customer upon completion. Where appropriate a draft will be provided for customer review prior to completion of the final version. Deliverables will be aligned with milestones as outlined in 2.1.2 below.

- Plans
 - Project Management Plan (SPMP)
 - Requirements Management Plan (RMP)
 - Software Development Plan (SDP)
 - Software Test Plan (STP)
 - Software Quality Assurance Plan (SQAP)
 - Software Configuration Management Plan (SCMP)
- Technical Documents
 - Software Requirements Specification (SRS)
 - Software Architecture Document (SAD)
 - Software Design Description(s) (SDD)
 - Software Test Description(s) (STD)
 - Software Test Report(s) (STR)
 - Software User Manual(s) (SUM)
 - Source Code (SC)
- Licenses for COTS
 - Licenses for any COTS products included in the system delivery
- Maintenance Contracts
 - For JDI's on-going support and enhancements of the system
 - For COTS products included in the system delivery

1.4 Summary of Schedule and Budget

The project is to be conducted over an 18 month period starting January 1, 2008 and finishing June 30, 2009. The Cascadia-JDI contract is fixed price totaling \$3.6M.

1.5 Evolution of the SPMP

This document will be subject to versioning as the project progresses. As additional detail is learned throughout the course of the project requirements, designs and estimates will be revised appropriately. JDI internal modifications will be represented by minor version number changes. A major version will be prepared for the customer if there is any material change to a deliverable.

2 References

1. Cascadia's Terms of Reference [details to be added]
2. Cascadia-JDI Contract [details to be added]
3. JDI's contracting procedures [details to be listed]
4. JDI's system and software procedures, guidelines and templates

Note: These will be customized to meet the needs of the Cascadia project.

3 Definitions and Acronyms

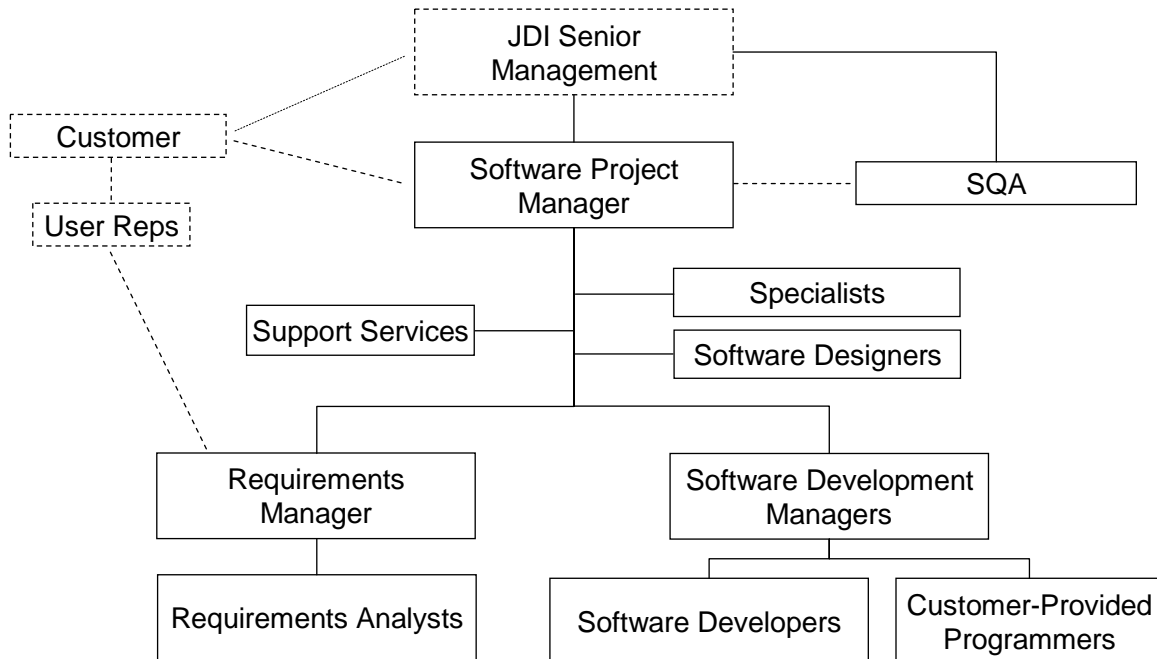
Term	Definition
UML	Unified Modeling Language – an object oriented design language.
SDLC	System Development Life Cycle
COTS	Commercial-Off-The-Shelf (typically software)
[other]	[Other definitions and acronyms required to understand this SPMP will be added as required and published as future SPMP revisions are issued.

4 Project Team Organization

4.1 Organization Chart: Internal and External Interfaces

Team roles and responsibilities will be reviewed and modified as required during project mobilization and will be updated as required. Several members will participate in multiple teams to facilitate coordination of critical tasks including project management and QA. The preliminary team organization is depicted below.

Project Organization Chart



Organization	Liaison	Contact
Cascadia Project Office	Project Director	Will Smith
Cascadia Users Group	User Representatives	Emily Post
CGI Consulting	Contracted designers and specialists	Serge Godin

4.2 Project Roles and Responsibilities

Following are the roles and responsibilities for members of the project team and supporting resources. Those listed in *italics* are external to JDI.

Position or Role	Responsibilities
Project Director	Ensure adequate response is given to JDI requests. Provide resources to JDI as agreed. Assemble and assign customer representatives (e.g. users and customer-provided programmers) to participate in milestone reviews. Sign off on requirements, scope, or budgetary changes throughout project.
SW Project Manager	Manage overall execution of the project including risk mitigation, budget and scope, and to ensure features and quality are delivered per the contract. Coordinates with other JDI resources to ensure project requirements are met.
SW Development Manager	Lead and mentor SW development resources. Foster a jelled team and an energized environment. Manage the career growth of team members. Handle or mitigate any interpersonal issues that arise among team members.
SW Designers	Responsibilities of Lead SW Architect have been migrated instead to a team of SW designers who will contract to JDI for a shorter duration than the Architect was planned for.
Specialist	Provide technical or domain expertise to the team. Coordinates with Requirements Engineers to ensure specification of system is correct and complete. Researches issues as directed by the SPM.
Sr. SW Engineer	Design, develop, and test sub-systems of the overall solution as the senior member of a small team. Ensures sub-systems meet the detailed requirements. Directly daily development tasks of team members as needed. Works with SQA Manager to ensure quality process is being followed correctly. Leads design and code inspections. Acts as resource to team members as needed. Works with the Lead SW Architect on broad design issues.
SW Engineer	Participate in design of sub-systems. Codes and tests individual modules as directed by the Sr. SW Engineer or the Architect. Ensures delivered items meet the detailed requirements and quality standards specified.
SQA Manager	Oversees SW development <i>process</i> and <i>quality</i> concerns, including managing the Requirements process. Assigns resources to work on projects as needed and works with them to ensure projects meet the JDI quality standards and the defined processes for development. Ensures final integration and system testing is completed as needed.
SQA Engineer	Develops and implements test plans and other V&V activities throughout project lifecycles as directed by the SQA Manager. Works with SW Development to ensure Unit Testing is completed with sufficient coverage and quality. Works with other SQA staff to promote JDI guidelines and quality practices.
SW Requirements Engineer	Specializes in eliciting (gathering) and specifying the correct requirements to ensure the built product satisfies the needs of the stakeholders. Identifies gaps in requirements and solicits information to remedy. Oversees requirements change process. Works with other SQA staff to ensure overall project process is followed.
SCM Engineer	Creates, manages, and oversees configuration management, build, and defect tracking systems. Works with SW Developers and managers as needed to ensure systems meet requirements and are being used properly in line with JDI's defined processes.

5 Managerial Process

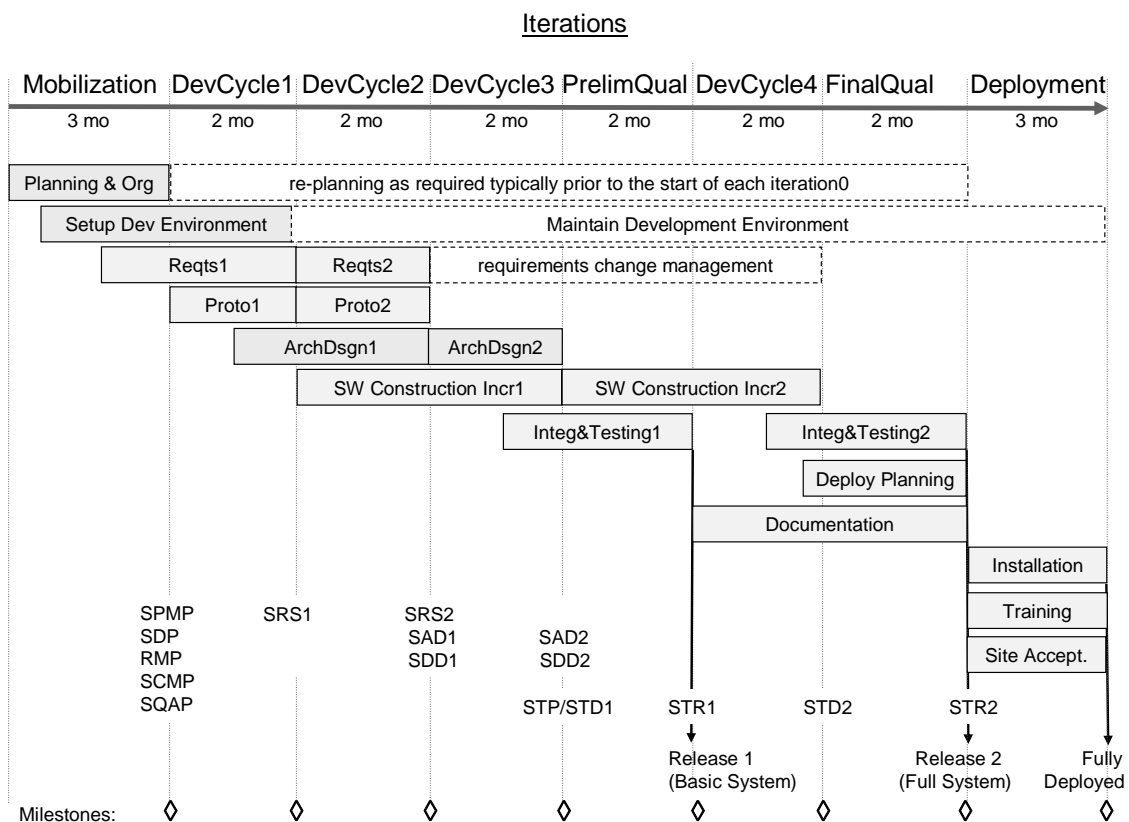
6 Technical Processes

6.1 Process Model

This project is organized around an iterative software development process with two major increments of development and product release. Aspects of the spiral model are being incorporated to address ongoing risk assessment, risk mitigation, planning and estimating throughout the project. The plan will be structured around several iterations as illustrated below over the project's anticipated schedule. Evolutionary prototyping iterations are planned that will focus on discovering and exploring the mobile technology options.

6.1.1 Development Iterations (a.k.a. "cycles")

Each iteration will have specific goals and deliverables arrived at with close customer and user involvement in activities. This approach will facilitate progress visibility and will ensure effective validation of requirements, the production of a stable design, and the delivery of a maintainable solution. This process, which is sketched below, is the foundation process template for the detailed description of activities, tasks, milestones, schedules and staffing described in later sections of this SPMP.



- Mobilization: Will focus on planning, organizing the team, start setting up the development environment, and start requirements development. Re-planning will occur as required throughout the project.
- Dev Cycle 1: Will complete the first requirements baseline, prototype the most uncertain application and technology areas, and start the first architectural design baseline. The development environment will be put in place by the end of this iteration.
- Dev Cycle 2: Will complete the second requirements baseline in parallel with a second iteration of prototyping (UIs and unresolved technology areas), and the completion of the first architectural

design baseline. The first increment of software construction will be started at the beginning of this iteration.

- Dev Cycle 3: Will complete the first increment of software construction and will also complete the second architectural design baseline. Given requirements at this stage are all baselined (approved by the customer), ongoing requirements management subject to agreed change control procedures will start at the beginning of this cycle. Further, integration and testing of the first software increment will begin during this iteration.
- Preliminary Qualification: Will complete integration and testing of the first software increment and deliver the first software release to the customer. This first release will not be deployed – rather, a select group of users will beta test this release during this iteration. Construction of the second software increment will start during this iteration.
- Dev Cycle 4: Will complete construction of the second software increment. Integration testing of this increment will start during this iteration. Deployment planning will be launched during this iteration. User, training, maintenance documentation will be started during this iteration.
- Final Qualification: Will complete integration and testing of the second software increment. This iteration will complete user, training and maintenance documentation. Formal (in-plant) customer and user acceptance of the software and system will be conducted during this iteration. Deployment planning will be completed during this iteration.
- Deployment: The deployment plan will be executed including software delivery and installation, transition into operation of all technical components, user and maintenance training, and delivery of updated documentation deliverables. Deployment will also complete customer and user site acceptance.

6.1.2 Milestones and Delivery Sequence

The completion of each iteration denotes a major project milestone (e.g. “Dev Cycle 3 Complete”). Formal deliverables called up in the contract will generally align with each major milestone. However, developmental work products and demonstrations will be scheduled as required to elicit customer and user feedback and provide visibility into progress.

Mobilization - This iteration of the project will deliver the following:

- Software Project Management Plan (SPMP): to be updated as required.
- Requirements Management Plan (RMP): calls for two SRS baselines.
- Define the organizational structure (teams) that will most effectively meet project goals including interfaces between the teams and the responsibilities of the teams. These teams may combine forces at various stages of the project.
- Identify and establish initial software engineering processes and products in the Software Development Plan (SDP): agree upon fundamental concepts and processes (code review, testing, etc); agree upon design nomenclature (UML) and ensure team is trained;
- Software Configuration Management Plan (SCMP): Select Fundamental Tools (e.g. code mgt)
- Create Software Quality Assurance Plan (SQAP)

Development Cycle 1 (Proto Alpha) – This iteration of the project will have a different focus for the various teams involved:

- Requirements refinement – complete initial baseline of software requirements specification (SRS): Prototype UI to help elicit customer needs; Customer initial validation of UI; List summarizing customer acceptance and desired changes to UI with customer sign-off
- Develop mobile technology platform expertise: Research and acquire training as required in mobile and handheld technologies to establish common level of understanding. Subsequent research into COTS products and piloting of cross platform tools and techniques. Identify COTS product candidates (excluding cross platform tools): Research calendaring and scheduling products; Identify criteria for selection

- Start developing initial Software Architecture Document (SAD) and Software Design Descriptions (SDD) baselines. Must provide sufficient detail for all development teams to begin proto development cycle 2

Development Cycle 2 (Proto Beta) – This iteration aggressively pursues the requirements and begins analysis and initial development work across all teams.

- Complete second Software Requirements Specification (SRS) baseline: Additional prototyping of UI to help elicit customer needs; List summarizing customer acceptance and mutually agreed upon changes to UI with customer sign-off; Any further changes after this cycle is completed will be in the form of Customer Change Orders (COs); COs are reviewed to determine if they are within scope of project and what their priority is
- Prototype Cross Platform Systems: Develop prototypes based upon what was learned in training; Define and develop a proof of concept system.
- Identify COTS product candidates (excluding cross platform tools): Develop prototypes based upon candidates identified; Define and develop a proof of concept calendaring and scheduling system.
- Complete first baselines of Software Architecture Document (SAD) and Software Design Descriptions (SDD) baselines. Provide sufficient detail for all development teams to begin development.; Any further changes should be in the form of Engineering Change Orders (ECOs); ECOs are reviewed to determine if they are within scope of project and priorities.

Development Cycle 3 (End to End) – Complete nominal implementation of all parts of system enabling complete 'end to end' software test plan (STP) for deployment.

- All teams to drive to completing of a nominal system. All core functionality in place.
- Verification and internal Validation performed against all core functionality
- Create Testing Plan (validation cycle) for customer review:
- Develop Software Test Descriptions (STDs)

Preliminary Qualification – Customer validation of end to end system and critical documentation.

- Review with customer capabilities of system. Identify where application is implemented as it will be delivered and where functionality is still to be added.
- Conduct software testing according to STP.
- Produce Software Test Reports(s) (STR) during testing.

Development Cycle 4 (Code Complete) – Complete system development and all documentation in preparation for final qualification / testing.

- Start drafting Software User Manual(s) (SUM) and training documentation
- Start deployment plan
- Complete final Training Plan and obtain customer sign-off

Final Qualification – Final testing with time budgeted for very short rework cycle to accommodate minor 'tweaks'. All documentation for review and sign-off with emphasis on the following: SUMs; STDs and accompanying STRs; Complete final Deployment plan and obtain sign-off; and deliver the source code (SC).

Deployment

- Enhancement proposal for COs that were out of scope
- All deliverables to be turned over to customer upon final payment
- Maintenance contracts
- Licenses for COTs products
- Pizza for everybody!

6.2 Methods, Tools and Techniques

JDI will adopt and tailor its standard software processes to meet the needs of the project. This tailoring effort will be conducted during the initial project mobilization iteration. These standard processes will yield requirements specifications, architectural design, detailed design, code reviews, testing and user documentation. At the time of writing it is anticipated that the following plans and specifications will be required to support this project. Note that some of these plans may be merged with this SPMP or grouped by related areas.

6.2.1 Software Development Methods, Tools and Techniques

Requirements Management Plan (RMP): will define the specification standards and methods by which requirements are to be documented, versioned and validated. This plan will also identify the requirements engineering tools to be used throughout the project.

Software Development Plan (SDP): this plan will describe and guide architectural design, unit development, software integration, and test processes, methods and tools, and design techniques used throughout the project. This plan will leverage JDI's standard software development procedures, templates and guidance. This plan will also identify the tools and environment required to support all software construction activities.

Software Test Plan (STP): This plan will define the test methods, procedures, and tools to be used to qualify software releases before they are released to Cascadia.

Software Test Descriptions (STD): Descriptions will be provided for all tests that will relate directly to the use cases and scenarios defined in the requirements specification. These will be in the form of javadocs. This will be auto-generated as part of the build process.

Software Test Reports (STR): Test results will be provided in the form of HTML junit reports. This will list the results to all tests and will be auto-generated as part of the build process.

Software Configuration Management Plan (SCMP): The SCMP will define how all software is to be shared, versioned and the means for auditing changes in the system (see below for details).

Software Quality Assurance Plan (SQAP): The SQAP will define the processes in place to ensure the system is developed, validated and verified in a manner to ensure compliance with all documented requirements. (see below for details).

6.2.2 Technical Specifications

Software Requirements Specification (SRS): Will be developed by the requirements team. This will provide the detailed description of how the system will work and will include all user interfaces. The requirements will be defined in a specific manner so they be verified and validated.

Software Architecture Description (SAD): Will be developed by the architecture team. This document will define the technologies to be applied, and fundamental design patterns employed by the system. Additionally it will detail the interfaces between the fundamental modules (UI, middleware and scheduling).

Software Design Description (SDD): Will be developed by the team leads and architecture group. This will include the interfaces between the system components, descriptions of the methods, and the data base schema.

6.2.3 User Documentation

Software User Manual(s) (SUM): The software user manuals will clearly define how the system works. It will provide examples that correspond directly to the use cases and scenarios defined in the requirements specification.

6.3 Development Environment and Infrastructure

- Software requirements analysts and software design engineers will use OOA/OOD processes to determine the system requirements and design. The SRS, SAD, and SDD documents will be specified in UML using the Borland TogetherJ application.
- Application Source Code will be developed in Java in accordance with JDI's coding standards defined.

- The Eclipse IDE will be used by developers to create the source code for Java programs.
- During component construction, code reviews will be performed at the completion of each module to ensure quality is built-in to the final product.
- Product ABC will be used to perform regression testing on the product following each major Integration task.
- Performance testing will be outsourced to PC Test Labs.
- Versions of project artifacts will be stored in a CVS repository (alternately Subversion). Application programmers use either WinCVS or the Tortoise Shell to store and retrieve project artifacts.
- Bugzilla will be used to track software defects
- A Change Control Board will manage and evaluate product requirement change requests.
- A project Wiki will be established for project documentation sharing.
- Microsoft Project will be used to manage information about the project schedule, tasks, and progress.
- Earned value analysis will be used to determine overall cost, schedule, and progress for this project. Earned value charts will be created using Microsoft Excel 2003.
- The JDI cost account software will be used to track labor costs against work packages.
- Software documentation will be produced using Microsoft Word 2003 and published as PDFs.

6.4 Acceptance, Training, and Deployment Processes

- An acceptance process identifying the documents, software, media, and procedures required to achieve acceptance and sign-off from will be developed jointly with Cascadia.
- A plan will developed to address all user and maintenance training to be delivered to the customer.
- A deployment plan will also be developed to guide the activities and tasks required to deploy the system release(s) to the customer.

7 Support Processes

8 Additional Supporting Plans