



Software Development Plan

Revision History

Date	Version	Description	Author
23/Oct/2002	1.0	Initial draft for feedback	Brian Sidharta
08/Dec/2002	1.1	Updated with feedback at end of E1	Brian Sidharta
14/Dec/2002	1.2	Updated RUP roles	Aleksandra Faust
15/Dec/2002	1.3	Updated with feedback during E2	Brian Sidharta
09/Feb/2002	1.4	Updated with new team members	Aleksandra Faust

Table of Contents

1.	Introduction	5
1.1	Purpose	5
1.2	Scope	5
1.3	Definitions, Acronyms, and Abbreviations	5
1.4	References	5
1.5	Overview	6
2.	Project Overview	6
2.1	Project Purpose, Scope, and Objectives	6
2.2	Assumptions and Constraints	6
2.3	Project Deliverables	7
3.	Project Organization	7
3.1	Organizational Structure	7
3.2	Roles and Responsibilities	9
4.	Management Process	12
4.1	Project Plan	12
4.1.1	Phase Plan	12
4.1.2	Iteration Objectives	15
4.1.3	Releases	16
4.2	Iteration Plans	17
4.3	Project Monitoring and Control	17
4.3.1	Requirements Management Plan	17
4.3.2	Schedule Control Plan	17
4.3.3	Quality Control Plan	18

4.4	Risk Management Plan	18
4.5	Close-out Plan	18
5.	Technical Process Plans	18
5.1	Development Case	18
6.	Supporting Process Plans	18
6.1	Configuration Management Plan	18

Software Development Plan

1. Introduction

1.1 Purpose

This Software Development Plan will define the development activities for developing the dbViZ system in terms of phases and iterations.

1.2 Scope

This Software Development Plan describes the plan for developing the dbViZ database schema visualization tool as a CS327/329 class project.

This plan is influenced by the dbViZ Vision Statement [1].

1.3 Definitions, Acronyms, and Abbreviations

Please refer to the dbViZ Glossary [2].

1.4 References

1. dbViZ Vision Statement
2. dbViZ Glossary
3. Rational Unified Process
4. dbViZ Development Case
5. dbViZ Iteration Plans

1.5 Overview

This document contains the following information:

Project Overview - provides a description of the project's purpose, scope and objectives. It also defines the artifacts that the project is expected to produce.

Project Organization - describes the organizational structure of the project team.

Management Process - defines the major phases and milestones for the project, and describes how the project will be monitored.

Technical Process Plans - provides an overview of the software development process, including methods, tools and techniques to be followed.

Supporting Process Plans - this includes the configuration management plan.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

The primary goal of the dbViZ project is to allow team members to learn how to follow a software development process to construct software. A secondary, but still important, goal is to construct an Alpha-quality version of an application that can be transitioned for future development.

2.2 Assumptions and Constraints

The end of the spring semester imposes a hard deadline for completing the project.

Because of this, emphasis will be placed on constructing a system that includes a large, but

not necessarily fully detailed feature set (breadth instead of depth).

Additionally, our staffing is not negotiable, limiting the flexibility of the team skill set. Midway through the project, the team may lose up to half of its members, forewarning us of the necessity to produce quality documentation. It is assumed that if more than half of the members leave that that time, the project will be cancelled.

2.3 Project Deliverables

The following deliverables will be produced during the project:

- Software Development Plan (this document)
- Vision Statement
- Use Case Model
- Use Case Specifications
- Use Case Realizations
- Development Case
- Glossary
- Software Architecture Document
- SQL '92 Specifications Document
- Iteration Plans
- Iteration Assessments
- Build

3. Project Organization

3.1 Organizational Structure

Professor Johnson and the CS327 TAs will evaluate the project at the end of the semester.

Their roles as Stakeholders are not clearly defined to the project team. The team generally

has no hierarchy, with individual members taking on management and review roles voluntarily. Below are the roles for Fall and Spring semesters.

Role	Names
Project Manager	Brian Sidharta, Ross Paul
System Architect	Ross Paul, Sonia Kaura, Jianmei Fan
System Analyst	Abhay Sathe, Brian Schoudel
Requirements Specifier	Abhay, Sathe, David Hampshire, Sonia Kaura, Sandra Faust, Jianmei Fan, Brian Schoudel
Requirements Reviewer	David Hampshire, Jianmei Fan, Brian Sidharta, Ross Paul
Architecture Reviewer	Brian Sidharta
Designer	Sandra Faust, Jianmei Fan
Implementor-Integrator	Brian Schoudel, Abhay Sathe, David Hampshire, Brian Sidharta, Ross Paul
Code Reviewer	Ross Paul
Tester	David Hampshire, Sandra Faust
Configuration Management Manager	Ross Paul
User Interface Designer	Brian Sidharta
Tool Specialist	Brian Sidharta, Ross Paul
Web Site Administrator	Brian Schoudel, Sandra Faust
Recorder	David Hampshire, Sonia Kaura

Team Roles for Fall semester (CD327)

Role	Names
-------------	--------------

Role	Names
Project Manager	Brian Sidharta
System Architect	Ross Paul
System Analyst	Brian Schoudel
Requirements Specifier	Sandra Faust, Brian Schoudel
Requirements Reviewer	Brian Sidharta, Ross Paul
Architecture Reviewer	Brian Sidharta
Designer	Larry Knox, Uday Kale, Brian Schoudel, Brian Sidharta, Ross Paul, Sandra Faust
Implementor-Integrator	Brian Schoudel, Brian Sidharta, Ross Paul, Sandra Faust, Larry Knox, Uday Kale
Code Reviewer	Ross Paul, Brian Sidharta
Tester	Jim Rarick, Sobby Gandotra
Configuration Management Manager	Ross Paul
User Interface Designer	Brian Sidharta
Tool Specialist	Brian Sidharta, Ross Paul
Web Site Administrator	Brian Schoudel, Uday Kale, Sandra Faust (back up)
Recorder	Sobby Gandotra, Larry Knox

Team Roles for Spring semester (CD329)

3.2 Roles and Responsibilities

Team members have volunteered for the following roles as defined by the Rational Unified Process [3] with the exception of the Implementor-Integrator and Recorder. At this time, we are only selecting roles needed to complete elaboration.

Role	Description
-------------	--------------------

Role	Description
Project Manager	Allocates resources, shapes priorities, coordinates interactions with the customers and users and generally tries to keep the project team focused on the right goal. The project manager establishes a set of practices to ensure the integrity and quality of project artifacts.
System Architect	Leads and coordinates technical activities and artifacts throughout the project. The architect establishes the overall structure for each architectural view: the decomposition of the view, the grouping of elements and the interfaces between these major groupings.
System Analyst	Leads and coordinates requirements elicitation and use-case modeling by outlining the system's functionality and delimiting the system.
Requirements Specifier	Details the specification of a part of the system's functionality by describing the Requirements aspect of one or several use cases and other supporting software requirements. The requirements specifier may also be responsible for a use-case package, and maintains the integrity of that package.
Requirements Reviewer	The requirements reviewer plans and conducts the formal review of the use-case model.
Architecture Reviewer	The architecture reviewer role plans and conducts the formal reviews of the software architecture in general.
Designer	Defines the responsibilities, operations, attributes, and relationships of one or several classes, and determines

Role	Description
	<p>how they will be adjusted to the implementation environment. In addition, the designer role may have responsibility for one or more design packages, or design subsystems, including any classes owned by the packages or subsystems.</p>
Implementor-Integrator	<p>Responsible for developing and testing components, in accordance with the project's adopted standards. Additionally, the Implementor-Integrator integrates components into the system.</p>
Code Reviewer	<p>Ensures the quality of the source code, and plans and conducts source code reviews. The code reviewer is responsible for any review feedback that recommends necessary rework.</p>
Tester	<p>Responsible for the core activities of the test effort, which involves conducting the necessary tests and logging the outcomes of that testing.</p>
Configuration Management Manager	<p>Provides the overall Configuration Management (CM) infrastructure and environment to the product development team. The CM function supports the product development activity so that developers and integrators have appropriate workspaces to build and test their work, and so that all artifacts are available for inclusion in the deployment unit as required. The configuration manager also has to ensure that the CM environment facilitates product review, and change and</p>

Role	Description
	defect tracking activities. The configuration manager is also responsible for writing the CM Plan and reporting progress statistics based on change requests.
User Interface Designer	Leads and coordinates the prototyping and design of the user interface.
Tool Specialist	Responsible for the supporting tools on the project. This includes selecting and acquiring tools. The tool specialist also configures and sets up the tools, and verifies that the tools work.
Web Site Administrator	Responsible for maintaining the project web site, which contains project news, general project information and project documentation.
Recorder	Responsible for writing a "Meeting Minutes" document after each team-wide meeting and making it available to all team members.

4. Management Process

4.1 Project Plan

4.1.1 Phase Plan

A Work Breakdown Structure is being prepared and will be provided in the next version of this document (TBD).

The development of the dbViZ system will be conducted using a phased approach where multiple iterations occur within a phase. The phases and the relative timeline is shown in the table below:

Phase	# of Iterations	Start	End
Inception Phase	2	24/Sep/02	04/Nov/02
Elaboration Phase	2	28/Oct/02	16/Dec/02
Construction Phase	3	03/Feb/03	28/Apr/03
Transition Phase	1	21/Apr/03	05/May/03

The table below describes each phase and the major milestone that marks the completion of the phase.

Phase	Description	Milestone
Inception Phase	The Inception Phase will develop the product requirements and establish the business case for the dbViZ. The major use cases will be developed as well as the high level Software Development Plan.	The Lifecycle Objectives Milestone at the end of the phase marks the completion of Requirements Definition and System Function Scoping.

Phase	Description	Milestone
Elaboration Phase	The Elaboration Phase will analyze the requirements and will develop the architectural prototype. At the completion of the Elaboration Phase, all use cases selected for Release 1.0 will have completed analysis and design. The architectural skeleton will test the adequacy of the architecture for Release 1.0.	The Lifecycle Architecture Milestone at the end of the phase marks the completion of the architecture design and skeleton implementation, and definition of all use cases.
Construction Phase	During the Construction Phase, remaining use cases will be analyzed and designed. The implementation and test activities to support the R1.0a release will be completed.	The Initial Operational Capability Milestone at the end of the phase marks the release of an Alpha version of the system.
Transition Phase	The Transition Phase will prepare the R1.0a release for distribution to the CS327 Staff.	The 1.0a Release Milestone at the end of the phase marks the release of a packaged Alpha version of the system.

4.1.2 Iteration Objectives

Phase	Iteration	Description	Associated Milestones	Risks Addressed
Inception Phase	I1	Defines initial product requirements and Software Development Plan.	none	Develops initial requirements documents for review.
	I2	Defines product requirements and Software Development Plan.	Lifecycle Objectives Milestone	Develops realistic Software Development Plans and scope.
Elaboration Phase	E1	Complete analysis and design for major use cases. Complete initial design of architecture.		Architecture can be reviewed. High-risk use cases can be reviewed.

Phase	Iteration	Description	Associated Milestones	Risks Addressed
	E2	Complete analysis and design for all use cases. Complete prototype of architecture.	Architectural Prototype	Architectural issues clarified. Technical risks mitigated.
Construction Phase	C1	Implement skeleton of architecture.	R0.1 Software	Architecture available for implementors.
	C2	Implement and test high-risk use cases	R0.5 Software	High-risk use cases are implemented.
	C3 – Develop Alpha release	Implement and test low-risk use cases. Complete alpha testing.	R1.0a Software	Defects minimized.
Transition Phase	T1	Package R1.0a Software for distribution.	R1.0a Software	Usable system released for CS327 Staff.

4.1.3 Releases

This Software Development Plan addresses the development releases of the dbViZ system.

Key features as defined in the Vision Document [1] are targeted for the first Alpha release.

Release 0.1 (internal release) must include at a minimum the general skeleton architecture of the system. It must be able to be started and stopped in a user-friendly manner.

Release 0.5 (internal release) must include at a minimum:

- Database schema importation for one database type.
- Diagram creation and editing.

Release 0.9a (Alpha) must include at a minimum:

- SQL query construction using a diagram.

4.2 Iteration Plans

Please refer to dbViZ Iteration Plans.

4.3 Project Monitoring and Control

4.3.1 Requirements Management Plan

4.3.2 Schedule Control Plan

The project manager will maintain a summary schedule showing the expected date of each milestone. Every week, using the weekly team meeting, the project manager will reevaluate the progress of the project, to determine whether the project is on schedule.

If the project is not on schedule, the project manager will consult with team members to determine corrective action, which may result in updating the schedule and/or reducing the number of optional functions that the system will perform.

4.3.3 *Quality Control Plan*

All deliverables are required to go through the appropriate review process. The review is required to ensure that each deliverable is of acceptable quality, using guidelines described in the Rational Unified Process [3] review guidelines and checklists.

4.4 **Risk Management Plan**

4.5 **Close-out Plan**

The Transition iteration plan will define the schedule for terminating the project, which will include making all deliverables available on the project web site, in addition to being sent directly to the CS327/329 Staff.

5. **Technical Process Plans**

5.1 **Development Case**

Refer to the dbViZ Development Case [4].

6. **Supporting Process Plans**

6.1 **Configuration Management Plan**

Configuration Management for software artifacts will be handled using CVS on SourceForge. Instructions on using CVS are distributed by the Configuration Management Manager. This information is archived on the project FAQ.

Documentation will be maintained on the project web site at <http://jdbv.sourceforge.net> by the Web Site Administrators.