**Team Assignment**

**Distributed Team Collaboration**

# Aim

To develop a project plan to establish an open source collaboration environment for distributed software engineering teams.

# Context

Your company, UBC Inc., is a systems integration company that develops custom web-centric e-business solutions in the healthcare industry. The company’s legacy products and services have been built from proprietary solutions. The company’s software development environments have been from vendors such as Oracle, Microsoft, and IBM (Rational).

The company is familiar with waterfall, evolutionary, and Agile development processes. Sound project management, configuration management, and quality assurance processes and tools are followed. Core software development competencies of the company include facility with specification-based requirements and use cases, UML-based architectural design, common programming and scripting languages (e.g. Java, Javascript,C, C++, Perl, …), continuous integration and automated testing techniques and tools.

Notably, UBC has begun to enjoy some success and cost savings using various open source SE products including CVS, Subversion, Eclipse and CruiseControl. The company’s delivered solutions are increasingly exploiting open source operating environments (Linux, Apache, Tomcat, EJBCA, MySQL, etc.).

UBC’s core systems and software development teams are located in Portland Oregon and Vancouver BC. Business activities are currently focused in the Pacific Northwest; however, the company’s target healthcare market spans all of North America. Key customers are currently located in Seattle, Portland OR, and San Francisco.

UBC is increasingly outsourcing software development. Much of the outsourcing is on-shore, but an increasing proportion is outsourced to companies in India and China. Mostly software development and system testing is being outsourced, often to separate companies addressing each of these aspects.

UBC is very concerned about maintaining management and technical control over projects. This is achieved by centering project management, requirements management, architectural design, and quality assurance in Portland and Vancouver. Process standards, including checklists, templates, design, coding, and documentation, are also managed out of these offices – including related training of company employees and contractors.

Although UBC requirements analysts are stationed in Portland and Vancouver, they are sometimes co-located with customer representatives and users for periods of time to achieve “face-time” and facilitate accurate elicitation and specification of requirements. On some occasions, customer representatives will attend facilitation meetings in Portland and Vancouver.

# Project Objectives

UBC’s primary objective is to enable and enhance effective collaboration among any given project’s stakeholders, including UBC, contractor and customer staff. To contain costs, UBC wants to make maximum use of open source collaboration and software development tools. The team assigned to meet these two objectives is tasked to specify and plan the establishment of a collaboration environment that can be effectively used by UBC staff, contractors and users to specify, develop, deploy, and maintain solutions delivered for the use of its customers.

# Assumptions and Limitations

This project is required to specify and plan out the development and integration of an initial system which will be evaluated thereafter by using it to implement a non-critical solution for a customer. This implies that the resulting specification and plan may assume a single server, or cluster of servers, supporting customer, user, contractor and developer workstations over the Internet – independent of location. It should be assumed that workstations are configured with open source browsers, and that other client components are also open source and capable of running on most common workstations.

For the purposes of the project, general-purpose collaboration tools should be distinguished from tools specific to various software engineering processes. For example, collaboration tools used to share arbitrary documents (e.g. plans, reports, spreadsheets, graphs, and other such documents) should be discussed separately from SE tools used to manipulate integral project data (e.g. tasks, defect records, and code).

If open source software cannot be found to support a given required element (below), the specification and plan should propose either:

1. A suitable free-ware solution for the required element characterizing the rationale and implications of the choice
2. A custom software solution for the required element, including high level functional description, also characterizing the rationale and implications for this choice
3. If no reasonably cost effective free-ware or custom solution can be identified, then a proprietary solution for the required element may be proposed.

# Requirements

## General Purpose Collaboration Tools

General purpose collaboration tools will be needed to support storing, sharing, and exchanging of arbitrary documents among all project stakeholders. Although other types of tools may be identified and proposed, the following essential capabilities must be supported by the specified solution:

1. **Repository:** An easy to use and administer open source repository for project documents shall be provided. The repository is to be capable of supporting multiple projects with separate work areas, possibly separated from each other by access control mechanisms.
2. **Messaging:** At least an open source email solution shall be provided. It would be desirable to support text messaging, instant messaging, and real-time alert support as well.
3. **Teleconferencing:** Tools to support video and audio conferencing capable of supporting meetings among at least 5 end-points must be supported. Support for a larger number of end-points would be desirable and should be highlighted.
4. **Discussion Forums:** The solution shall provide open source tools that support multiple discussion threads among stakeholders of a given project. Access controls separating the stakeholders is to be an essential property of this requirement.
5. **Shared Space:** The solution shall support the sharing of documents, spreadsheets, and other “office documents” among stakeholders of a given project with appropriate project and user access controls.

## Software Engineering Processes Supported

Special purpose tools supporting specific software engineering and project management tools should be proposed. Like the above collaboration tools, they should support distributed software engineering teams working from their developer workstations. To the extent possible, these should be open source or freeware tools that are configurable for multiple projects with prerequisite access controls. The following list covers the essential and desirable software engineering processes to be supported:

1. **Project Management**: This tool shall support the specification of tasks and action items allocated to distributed team members. Project managers, team leads and team members should be able to update tasks and report progress against these tasks in a controlled manner. Desirable features include the ability to produce graphical schedules (e.g. Gantt Charts) and earned value summaries.
2. **Configuration Management:** The proposed solution shall incorporate a source code control system that can be used across the project team supporting multiple projects.
3. **Defect Tracking:** A defect/issue/bug tracking system that can be easily deployed and administered across the team shall be included.
4. **Requirements Management:** It is desirable to include an open source tool to support the specification and management of requirements including the capability of version control of individual requirements.
5. **Architectural Design Specification:** It is also desirable to include an open source tool to support the specification and management of designs including versioning of designs. The tool should support UML diagrams and related design specification.
6. **Development Environment:** An open source software development environment shall be part of the proposed solution. In particular, this environment is to include compliers, source code editors and support for the development of SQL databases, static and dynamic web pages, and graphical user interfaces.
7. **Continuous Integration, Code Coverage, and Testing:** A tool or set of tools that can be used to automate the production of builds, the execution of regression tests, and the reporting of test results is to be provided. Test results should be integrated with defect tracking to the extent possible. Code coverage tools should also be part of the proposed solution.
8. **Software Metrics:** Open source tools capable of measuring properties of delivered software should be incorporated into the solution. In particular, tools that count code, count functions or methods, and calculate code complexity are desirable capabilities.

# Deliverables

At least the following plans and technical documents are to be delivered.

1. Technical Specification:
   1. Agreed functional requirements (signed off by sponsor/instructor)
   2. Context diagram(s)
   3. Architectural / Configuration Diagrams representing the proposed solution
   4. Descriptions of each tool comprising the solution including the rationale for selecting each, key assumptions, limitations, and configuration requirements
      * Note: custom and proprietary parts of the solution, if included, are also to be similarly documented
2. Project Management Plan: High level plan for delivery of the contemplated solution. This plan should include the implementation team organization, tasks, schedule, allocation of resources and a rough estimate of the labor and other costs to complete the project.
3. Deployment & Support Strategy: Overview of how the tools will be deployed, including details regarding any potential tool integration, maintenance of the open source code, installation remotely to all distributed team members, and ongoing support.
4. Final Presentation: This is a PowerPoint presentation highlighting the proposed solution during the last class prior to delivery of final documents.