

# Software Process and Project Plan

for Trillium Health - Grant Management

Version: 1.2

Prepared by: Akshay Karnawat

Date: November 26, 2014 (updated)

Team Members: Shannon Trudeau (smtofny@gmail.com)

Matt Metcalf (mrm9084@gmail.com)

Brian To (bxt5647@rit.edu)

Akshay Karnawat (axk6802@rit.edu)

---

## REVISION HISTORY

VERSION	DESCRIPTION	DATE
1.0	Initial	9/19/2014
1.1	Added introduction, management and technical process	9/21/2014
1.2	Modified Process to include RUP, Updated other sections	11/26/2014

---

# TABLE OF CONTENTS

<b>INTRODUCTION</b>	<b>4</b>
SYNOPSIS	4
OVERVIEW	4
GOALS AND SCOPE	4
DELIVERABLES	4
<b>MANAGEMENT PROCESS</b>	<b>5</b>
WORK ACTIVITIES & BREAKDOWN	5
MONITORING AND CONTROLLING MECHANISMS	5
PROJECT SCHEDULE	5
RESOURCE ALLOCATION	6
RISKS AND MITIGATION PLAN	6
MEASUREMENT & METRICS	7
<b>TECHNICAL PROCESS</b>	<b>7</b>
METHODOLOGY	7
PROCESS RATIONALE	8
SOFTWARE ARTIFACTS	8

---

# INTRODUCTION

## Synopsis

Trillium Health provides many services to the communities of Rochester, including operating a full service clinic, pharmacy, needle exchange, social and care management services, community education, prevention, and many other things for a variety of high needs individuals in the Rochester/Bath/Geneva area. Since Trillium operates as a non-profit 501(3)(c) organization, they rely heavily on grant funding to continue their operations. We will create a software application to manage the grant process (and all grant documentation) from start to finish that will monitor, track, store, and alert people to complete necessary tasks as they are due, as well as help manage the initial process of searching for and applying to grants.

## Overview

The primary function of this project is to monitor, track, and store grants. Every user with a certain role will have their own account and a personalized dashboard that will allow them to view and manage grants. This will give the user a sense of their tasks and assist them in getting started on their day. The system will also assist in providing general information about the grants such as name, address, associates contacts, documents, etc. This will help the users when there are any problems within the process of the grant. Another major and an important function of this project is to manage documents that are associated with the grant. Since each grant has multiple documents such as salaries, budgets, resumes, notes, certification, statistics, etc., managing them is a process heavy task. With the help of this application, the users will have a centralized place for all documents and their revision history for each grant process.

## Goals and Scope

Our top level goal is to have a well established application that will allow users to manage, track, and store grants; alert people when a certain task is due; have the system store documents and its revision history throughout the process of the grant; and have the system gather all the documentation for auditing and compliance purposes.

Additionally to the main scope defined above, the application will also provide integration to outlook, status updates through emails, and task templates for grants. These features are additional and will be integrated into the application after the main features are fully completed and tested. The main scope and additional features of this project are defined in detail in the Software Requirements and Specification Document.

## Deliverables

There will be multiple deliverables throughout the life cycle of this project. These deliverables include: the project and process plan, software requirements specification, design and architecture document, test plan, and finally,

---

the final product. Along the way, there will be multiple project demos as a measure of this team's progress and will allow the team to receive feedback from Trillium Health.

## MANAGEMENT PROCESS

### **Work Activities & Breakdown**

The work breakdown of this project is broken down into major deliverables. The project team spends time each week to structure and identify the major deliverables and subdivide them into smaller sections. The smaller sections are further decomposed and assigned to a team member. Each team member works on a task until the next group meeting or the due date specified. Once everyone is done with their section, each team member looks over the work of other team members to provide feedback and bring up any issues relating to the project.

### **Monitoring and Controlling Mechanisms**

The team uses Trello to maintain a list of all the tasks that are to-do, in-progress, and done. Additionally, the team also uses Google drive as a shared document repository and as a way to provide any feedback on the deliverables sections.

### **Project Schedule**

The project schedule is broken down into weeks and is detailed below.

---

Week	Schedule
1 - 5 (Fall 2014)	Identify Scope Identify Risk Create a Process and Project Plan Create a Software Requirements Specification Document Identify Technologies used
6 (Fall 2014)	Lockdown Requirements (have sponsors sign-off) Present a rough draft of SRS
7 (Fall 2014)	Negotiate a final requirements document Begin Design by the end of week 7
8 - 9 (Fall 2014)	Rough draft of Architecture and Design Document Review rough draft of Design doc (have sponsors sign-off)
10 (Fall 2014)	Set up development environment <ul style="list-style-type: none"> <li>- Repositories</li> <li>- Continuous Integration</li> <li>- Issue management</li> <li>- Deployment</li> <li>- Testing</li> <li>- Database</li> <li>- Smoke Test (functional test)</li> </ul>
11 - 15 (Fall 2014) 1 - 10 (Spring 2015)	Software Development Life cycle <ul style="list-style-type: none"> <li>- Analysis and Design</li> <li>- Implementation and Code</li> <li>- Test</li> <li>- Deploy</li> </ul>
11 - 15 (Spring 2015)	Poster Presentation Technical Report Gather training Documents for sponsors Demo Finalize and Deliver

### Resource Allocation

Every week, each team member will spend around 8-12 hours working on the deliverables, project meeting time, and any additional work needed for the project.

### Risks and Mitigation Plan

Many risks and mitigation strategies associated with people, process, product, and technologies are identified for this project. A list of these risks and their mitigation strategy are detailed in the Risk Management excel sheet.

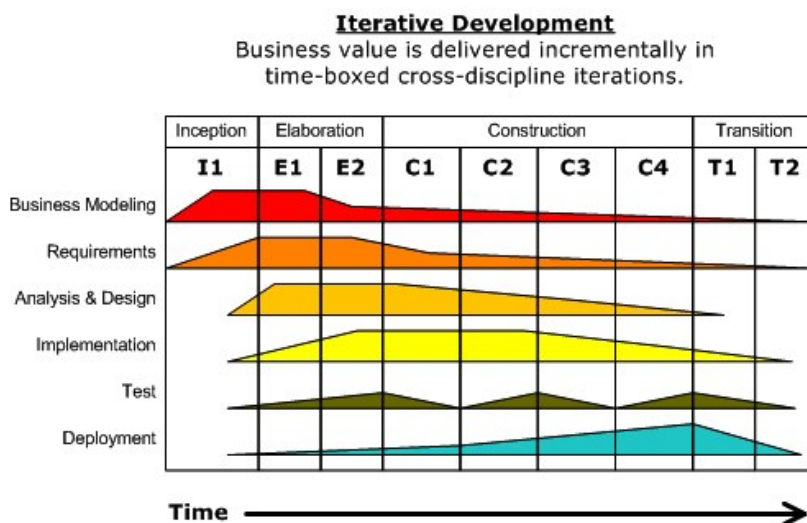
### Measurement & Metrics

The team will keep track of the total time spend each week and individually on the project allowing them to see how much effort was put into the project and help us determine if more effort is needed. The team will also keep track of the number of tasks and the number of tasks completed over each phase. This will give the team an understanding of how much of the planned scope they were able to complete over the iteration. If suppose, the project was left with open tasks for that iteration, then the team will be able to plan for the next phase accordingly. Another good metric that the team will keep track of is the number of bugs found over 1000 lines of code. This would be helpful because it will give the team an understanding of how good their code base is and if they need to improve on it for the next phase.

## TECHNICAL PROCESS

### Methodology

We decided to use the Rational Unified Process (RUP) which is an iterative software development process methodology for this project. The picture below gives a brief overview of this methodology. The table on the next page gives our detailed breakdown for each of the construction phases.



**Inception Phase:** The team will identify the project scope, requirements (functional and non-functional) and risks at a high level but in enough detail that work can be estimated.

**Elaboration Phase:** The team will deliver a working architecture and identify the technologies being used that mitigates the top risks and fulfills the non-functional requirements.

**Construction Phase:** The team will incrementally fill in additional architecture with production ready code produced from analysis, design, implementation, and test of the functional requirements. The team will follow the 5-week per iteration construction plan. More details are provided below.

C1-C3	Week 1	Analysis and Design
C1-C3	Week 2	Implementation
C1-C3	Week 3	Implementation
C1-C3	Week 4	Testing and Release
R1-R3	Week 5	Buffer for any planned risks

**Transition Phase:** The team will finally deliver the system into the production operating environment and provide any necessary documentation for the support team and the users.

## Process Rationale

The team choose the Rational Unified Process (RUP) and the iterative development methodology since the project has many cycles. The project sponsors want the application to be modifiable and adaptable with other applications in the future. The team can focus on developing the application iteratively, manage requirements, use component-based architecture, continuously verify software quality, and control changes to software.

## Software Artifacts

Ultimately, the entire team will be responsible for creating and maintaining the documents needed for this project on a week to week or an iteration to iteration basis, if needed. The documents that will be delivered are as follows:

1. Process and Project Plan
  2. Software Requirements Specification
  3. Architecture and Design
  4. Test Plan
  5. Final Deliverables (includes source code, database schemas, training documents, etc.)
-