Kreogist Mail
Development Documentation

Software Project Management Plans

January 25, 2016

Reference Number: KMKOT02.
This document is capable with IEEE Std 1058-1998.

All information provided here is subject to change without notice. Contact Kreogist Dev Team to obtain the latest Kreogist product specifications and roadmaps.

Kreogist Mail may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.

No computer system can be absolutely secure. Kreogist does not assume any liability for lost or stolen data or systems or any damages resulting from such losses.

The products described may contain design defects or errors which may cause the product to deviate from published specifications. Current characterized errata are available on request.

All the other documents mentioned in this document could be found at the official site of Kreogist Dev Team. Contact Kreogist Dev Team if there's any trouble.

Intel, Intel Core and the Intel logo are trademarks of Intel Corporation in the U. S. and/or other countries.

Linux is a trademark of Linus Torvalds in the U. S., other countries, or both.

Microsoft and Microsoft Windows are trademarks of Microsoft Corporation in the U. S. and/or other countries.

Macintosh is a trademark of Apple Inc. in the U. S., other countries, or both.

UNIX is a registered trademark of The Open Group in the U. S. and other countries.

*Other names and brands may be claimed as the property of others.

First Edition (Jan 2016)

This edition applies to Version 0.1 of Kreogist Mail.

Kreogist Dev Team

Kreogist Mail
Development Documentation
Software Project Management Plans

January 25, 2016

Signature

The following signature indicates approval of the enclosed Software Project Management Plan.

Revision History

Revision	Version	Description	Date
KMKOT02	-001	Initial commit	Jan. 17th, 2016
KMKOT02	-002	Internally accepted	Jan. 25th, 2016

Preface

This document is an update to the specifications contained in the "Affected Documents" table below. This document is a part of product (project) Kreogist Mail.

This document may also contain information that was not previously published.

Affected Documents

Document Title	Document Number
Kreogist Mail Software Quality Assurance Plan	KMKOT04
Kreogist Mail Software Verification and Validation Plan	KMKOT05

Related Documents

Document Title	Document Number
Kreogist Mail Software Requirement Specification	KMKOT01
Kreogist Mail Software Design Specification	KMKOT03

Contents

List of Figures

List of Tables

1 Overview

This section of the document is an introduction to the software development portion of the Kreogist Mail project (the project). It will describe the purpose of the project and the objectives that are to be accomplished, the assumptions and constraints that underlie the effort, the deliverables that will be produced by the project, and a summary of the project schedule and budget.

1.1 Project Summary

1.1.1 Purpose, Scope and Objectives

The purpose of the project is to analyze the requirements of, design, implement, and maintain the software for Kreogist Mail (Mail), according to the requirements specified by the client.

All activities directly related to the purpose are considered to be in scope. All activities not directly related to the purposes are considered to be out of scope. For example, network availability is not within the scope of this project.

The objectives of the project are as follows:

- complete the project by the project due date;
- complete the project within budge;
- provide all deliverables identified in section 1.1.3, i.e.*Project Deliverables* by the project due date;
- fulfill all stated requirements, as in the SRS, of the software product deliverable.

1.1.2 Assumptions and Constraints

The project will be planned with the following assumptions:

1. this project is a component of a larger project;
2. this project will deliver only the Mail components of the larger project;
3. initial estimates for the project as provided in this SPMP are +/- 45%;
4. the software products will be based on the following structures/projects/foundations:
    - Qt 5.5.1 or later. With its core, gui, widgets and network module installed;
    - Kreogist Account system, hosted on bmob.cn;
    - Kreogist GUI components from Kreogist Mu.

Mail will be deployed on mainstream desktop Linux release, Mac OS X and Windows.

5. we will be able to acquire the expertise of two outside consultants to assist with the user interface designing.
6. this SPMP is submitted as a firm-fixed-price (FFP) bid; the project shall not exceed the established budget.

Assumptions and constraints of application please check out section *Constraints* and *Assumptions and Dependencies* in *Kreogist Mail Software Requirements Specification*[1].

The project will be planned with the following constraints:

- budget
  $0 (software portion only)
- time
  1 year
- staff
  1. two outside consultants will be required to assist in the requirements and detail design phases of the project.
  2. programmers from Github and OSChina open source community to help us improve the quality of the project.
- maintenance
  The project will receive for a 5-year support. The following version support will be updated in other projects.

1.1.3 Project Deliverables

This project starts at January 4th, 2016. Project should be done before January 31th, 2017. Kreogist Dev Team should delivered the following products:

1. Software program and library binaries
2. Software source code
3. Software documentation
   - Compile documentation
   - Technical Documentation generate with Doxygen
4. Installation of software program and library binaries on target hardware
5. Project documentation

- Software Requirements Specification (SRS), ID: KMKOT01
- Software Design Specification (SDS), ID: KMKOT03
- Software Project Management Plan (SPMP), ID: KMKOT02
- Software Quality Assurance Plan (SQAP), ID: KMKOT04
- Software Verification and Validation Plan (SVVP), ID: KMKOT05

1.1.4 Schedule and Budget Summary

The project has the following high-level schedule:

- COMP3500 course guide.

The project has a budget of $0. Once the software product is delivered, annual maintenance costs should be no larger than $0. Server renting fee is not costed in the budget.

The project will be tracked using the Earned Value Management System (EVMS).

1.2 Evolution of the Plan

The plan is considered to be a dynamic document and will be updated monthly by default and on an unscheduled basis as necessary. Scheduled updates to the plan will occur once every month, on the last business day of the month.

Notification of scheduled and unscheduled updates to the plan will be communicated via e-mail to all project participants according to the section 5.3.4, i.e.*Reporting Plan*.

Once the initial plan is finalized, a baseline of the plan will be created. Changes to the plan will take place against this baseline. The plan will only receive further baselines if significant change in scope occurs.

Changes in this information will lead to a new SPMP with a new version number, but with the same status. However, if these changes lead to changes in the milestones planning of the project, described in section 5.2.2, i.e.*Schedule Allocation*, these changes are discussed with the leader first, before they are incorporated in the document. This will be done during progress meetings. The detailed planning for each phase is described in the appendices of this document. These appendices are updated at different moments in time during the project, but before the start of the phase they refer to.

2 References

[1]. Kreogist Dev Team. "Kreogist Mail Software Requirements Specification", January, 2016.

[2]. Free Software Foundation, "GNU Free Documentation License", See `http://www.gnu.org/licenses/fdl.html` (last checked January 13th, 2016), November 3, 2008.

[3]. Wikipedia, "Qt (software) - Wikipedia, the free encyclopedia", See `https://en.wikipedia.org/wiki/Qt_%28software%29)`last checked January 16th, 2016), January 4, 2016.

[4]. Terasoft, Inc.. "Software Project Management Plan (SPMP) for Nirvana National Bank ATM Software Project", May 08, 2004.

[5]. IEEE Software Engineering Standards Committee. "IEEE Std 1063-2001, IEEE Standard for Software User Documentation", December 20, 2001.

[6]. IEEE Software Engineering Standards Committee. "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.

[7]. IEEE Software Engineering Standards Committee. "IEEE Std 1016-1998, IEEE Standard for Information Technology–Systems Design–Software Design Descriptions", July 20, 2009.

[8]. IEEE Software Engineering Standards Committee. "IEEE Std 1058-1998, IEEE Standard for Software Project Management Plans", December 22, 1998.

[9]. IEEE Software Engineering Standards Committee. "IEEE Std 1074-1997, IEEE Standard for Developing Software Life Cycle Processes", 1997.

[10]. IEEE Software Engineering Standards Committee. "IEEE Std 1012-1998, IEEE Standard for Software Verification and Validation", 1998.

[11]. IEEE Software Engineering Standards Committee. "IEEE Std 828-2005, IEEE Standard for Software Configuration Management Plans", August 12, 2005.

3 Definitions

This section will define or provide a reference to the definitions of all terms required to properly interpret the Plan. This section shall describe the acronyms and notations used in the Plan.

Table 1: Definitions, Acronyms & Abbreviations

Acronyms	Definition
AD	Architectural Design
ADD	Architectural Design Document
ATP	Acceptance Test Plan
baseline	a work product that has been formally reviewed and accepted by the involved parties
Client	Monitor, Agent or Submitter
CM	Configuration Management
FFP	Firm-Fixed-Price; a price for fulfilling the contract that will not be under- or over-run
IEEE	Institute of Electrical and Electronics Engineers
milestone	a scheduled event used to measure progress
MG	Merge Phase
QAM	Quality Assurance Manager
RA	Reanalysis Phase
SC	Sprint Check Phase
SCMP	Software Configuration Management Plan
SDS	Software Design Specification
SM	Senior Management
SPMP	Software Project Management Plan (this document)
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
SR	Software Requirements
SRD	Software Requirements Document
SRS	Software Requirements Specification
STD	Software Transfer Document
STP	Software Testing Plan
SUM	Software User Manual
SVVP	Software Verification and Validation Plan
TR	Transfer Phase
UTP	Unit Test Plan
VPM	Vice Project Manager
VQAM	Vice Quality Assurance Manager

4 Project Organization

4.1 External Interfaces

The external structure of the person/companies/organizations having interest in the project is as shown in the following organizational chart. While the external organization chart suggests "functional" organization, this is for grouping purposes only. The leader of development team is responsible for administration of and dissemination of general organizational information to the team, but does not enjoy a strong reporting relationship with group members.

The external organization chart which shows these relationships is included in Appendix A, i.e.*Organization Charts*.

Leader will not incur any direct costs to the project; any administrative costs are integrated into the per-hour cost of the individual resources.

4.2 Internal structure, Roles and Responsibilities

In the following table, the roles which are distinguished and the person(s) assigned to each role are given. Roles within the group are described in section 5.1.2. i.e.*Staffing plan*.

Table 2: Member Role

Role	Name	E-mail
Team Leader & Team Member	Haolei Ye	u580415@anu.edu.au
Project Manager & Project Manager	Han Wang	u5868791@anu.edu.au
Quality Assurance Manager & Team Member	Jinshuai Ma	u5870682@anu.edu.au
Team Member		
Team Member		

Responsibilities of all the avatar in the project have been listed here:

Project Manager Produce and maintain a project management plan and lead the project according to this plan thus ensuring that the product is delivered on time and as specified in the SRS. The PMs management task includes but is not limited to:

- Motivating team members;
- Forming teams;
- Checking progress;
- Managing the time budget;
- Gathering all the requirements from outer, manage all the information;
- Providing feedback to the Team Leader through progress reports.

Quality Assurance Manager Guarantee that the product will be delivered as agreed and that it is of good quality. This includes but is not limited to:

- Writing the SQAP and the SVVP;
- Verifying that procedures and standards which are defined in the SQAP and SVVP are adhered to;
- Checking that all project documents are consistent;
- Arranging formal reviews;
- Refactor the unqualified codes and submit to team leader;
- Monitoring and reviewing all testing activities.

Team Leader Perform all necessary activities to ensure that a task assigned to a team is performed well and on time. This includes but is not limited to:

- Planning and coordinating team activities;
- Defining work packages and goals, assigning tasks;
- Motivating team members;
- Check out and sign all the documentation;
- Release the products and give lecture to all end-users.

Team Member Perform all necessary activities to ensure that a task assigned to a team is performed well and on time. This includes but is not limited to:

- Assisting the Team Leader or Project Manager by signaling problems in an early stage;
- Executing plans made by the Team Leader and by the Project Manager;
- Keeping track of time spent on various tasks;
- Following procedures and plans.

5 Managerial Process Plans

5.1 Start-up Plan

5.1.1 Estimation Plan

Schedule, Cost, and Resource Estimates

An estimation chart showing activities, estimated duration, estimated cost, and estimated resource requirements is included in Table 3.

Table 3: Estimation Chart

ID	Task Name	Duration	Resource Names
1	Kreogist Mail project	1 year	All of the members
2	Requirement confirmation	1 week	Project Manager, Team Leader
3	Project Initiation	1 day	Team Leader, Team Members
4	Framework development	4 weeks	Team Leader, Team Members, Quality Assurance Manager
5	Infrastructure development	4 weeks	Team Leader, Team Members, Quality Assurance Manager
6	UI Elements development	6 months	Team Leader, Team Members, Quality Assurance Manager
7	Preference system development	2 months	Team Leader, Team Members, Quality Assurance Manager
8	Plugins development	2 months	Team Leader, Team Members, Quality Assurance Manager
9	Kreogist Account system embedded	1 months	Team Leader, Team Members, Quality Assurance Manager
10	SDK refactor	2 weeks	Team Leader, Team Members, Quality Assurance Manager
11	System test	1 week	Team Leader, Team Members, Quality Assurance Manager

Estimation methods

Schedule duration and work estimation for each leaf activity in the Work Breakdown Structure (WBS) will be performed using a combination of the following methods and data sources:

1. Resource input
2. Organizational project history data
3. Kreogist project history data

Cost estimation for each activity will be performed by multiplying the amount of work expected by the hourly rate for the resources connected to the activity, multiplied by the percentage of participation that each resource expects to make toward the activity.

The resulting estimates for each leaf activity will be rolled-up to produce an estimate for the larger group of activities that the activity is a part of. The highest-level activity in the WBS (after attaching schedule, resource, and cost estimates) will therefore reflect the schedule and cost estimates for the entire project.

Re-stimation methods

When re-estimation is necessary, it will be performed using the following methods and data sources:

1. Resource input
2. Member working input

Once new estimates have been collected, and if schedule is adversely affected (+/- 10%), organizational project history data will be used to determine whether or not it would be effective to add additional resources to assist in completing the activity, taking roll-on time into consideration.

Re-estimation Schedule

Time has been allocated in the schedule for monthly SPMP updates. Necessary updates to the cost, schedule and resource estimates will be included in these SPMP updates. However, such rebaselining will only take place in extreme circumstances, such as when significant scope change has been introduced.

The purpose of these monthly updates is to force allocated time toward maintaining the SPMP and to provide a schedule on which stake holders can expect to see updates to the plan. A revised SPMP will be published following each of these update sessions regardless of whether any significant changes have been made so that it is obvious to all involved that the scheduled update has occurred.

Impromptu updates to the estimation plan will be made as necessary and communicated to those affected. In particular, detailed explanation is given below to the handling of communication of these update types:

- Resource
- Cost
- Schedule

Resource

1. If an increase in existing allocation is required
    - Affected resource
    - Project manager
2. If addition of internal resources is required
    - Team Leader
    - Project manager

Cost

1. If an increase in costs is required but does not exceed the project budget
    - Team Leader
    - Project Manager
2. If an increase in costs is required which exceeds the project budget
    - Team Leader
    - Project Manager
3. If a decrease in costs is expected
    - Project Manager

Schedule

1. If an increase in schedule is required which does not exceed the deadline
    - Project Manager
    - Quality Assurance Manager
2. If an increase in schedule is required which exceeds the deadline
    - Team Leader
    - Project Manager
    - Quality Assurance Manager
3. If a decrease in schedule is expected
    - Project Manager
    - Quality Assurance Manager

5.1.2 Staffing plan

For role of the member, please check out section 4.2, i.e.*Internal structure, Roles and Responsibilities*.

Resource Requirements

Based on initial estimates, the project will require the human resources shown in the table below. Quantity required, estimated work requirement, key work periods and affordable hourly rate ranges are included for each resource type. The key work periods are those periods where the resource will be heavily allocated and should be prepared to have significant (above 20% working time) availability to the project:

Table 4: Human Resources Requirements

Human Resource Type	Key Periods	Key Project Phase(s)	Qty
Project Manager	All	All	1
Quality Assurance Manager	All	All (but most work up-front during definition)	1
Team Leader	All	Requirements, Design, Development	1
Team Member	All	Development	2

Attendance at weekly project status meetings

Most staff will be required to attend weekly project status meetings, for which the dates are yet to be determined. All staff identified as "Leads" will be required to attend the meetings. Staff who are in a group underneath a "Lead" will not be required to attend, while staff who have a "Lead" role, or who have no subordinate "Lead" will be required to attend.

Reassignment

All efforts will be made to communicate changes in the key resource requirement dates to leader.

5.1.3 Resource Acquisition Plan

All human resources shall be acquired for the purposes of working on the project by the team leader. The team leader must present the resource requirements in detail to project manager and the functional managers of each requested resource; team leader has the ultimate responsibility for approving resources to work on the project.

The project manager shall be responsible for acquiring all non-human resources required by the project. The non-human resources identified as being required for the project are:

- Printing services
- Documentation management
- Software repository

5.2 Work Plan

5.2.1 Work activities

Work activities are illustrated by a work activities table (Table 5).

Table 5: Work Activities

ID	Task Name	Predecessors	Successors
1	Kreogist Mail project		
2	Requirement confirmation		3
3	Project Initiation	2	4, 5
4	Framework development	3	6
5	Infrastructure development	3	6
6	UI Elements development	4, 5	7, 8
7	Preference system development	6	9

Continued

ID	Task Name	Predecessors	Successors
8	Plugins development	6	9
9	Kreogist Account system embedded	7, 8	10
10	SDK refactor	9	11
11	System test	10	

5.2.2 Schedule Allocation

Schedule allocation is illustrated by a network diagram at figure 1.

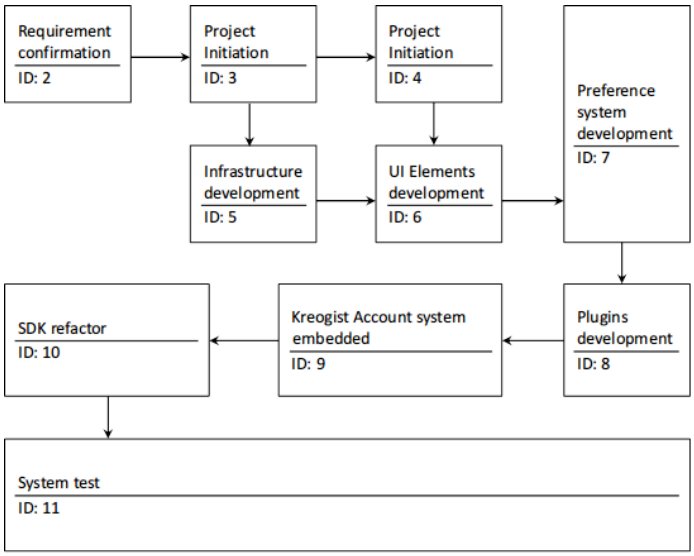

Figure 1: Network Diagram

5.3 Control Plan

5.3.1 Requirements Control Plan

Requirements tracing

Requirements have all designed in SRS. Before course COMP3500 provides any requirements tracing tools, we will trace the requirements manually. All work effort must be related to a traceable requirement, in order to limit unnecessary work and ensure integrity of the product requirements.

Prioritization

When a requirement is entered into the schedule, it is assigned a priority, as follows:

- 3 = mission critical (product must have)
- 2 = important (should exist, but not absolutely necessary)
- 1 = nice to have (should be present if time permits, but is optional)

A requirements priority will affect the attention it receives when tradeoffs become necessary, and when changes to requirements are requested. In conjunction with the above, a requirement change priority will also be used to rate the priority of incorporating change to the requirement, as follows:

- 3 = critical (change must be made to requirement)
- 2 = important (change should be made, but not absolutely necessary)
- 1 = nice to have (change should be made if time permits, but is optional)

Product requirements change control

Changes to product requirements will be considered based on their priority, their point-in-time of introduction within the overall project schedule, the extent of their impact to work products already base-lined as configuration items, and the extent of their impact to in-progress work products.

All efforts will be made to incorporate changes to priority 3 requirements. Changes to priority 2 and 1 requirements will be handled only if time permits and/or the customer is willing to negotiate a increase in project budget and schedule.

The development model being used for this project is based on up-front solidification of requirements and is appropriate for software products having the profile of the product being developed by this project. Therefore, all requirements change requests should come with the expectation that project schedule and/or budget will be affected if they are introduced after the requirements phase is complete.

Reporting

Assessments produced for each area of assessment listed below above must be communicated to team leader:

1. product scope
2. product quality
3. project schedule
4. project budget

5. project resources

6. project risks

Team leader will coordinate necessary resources to approve or reject the requirements change and associated/negotiated increases in budget and schedule. When multiple requirements changes are under consideration at the same time, the requirement-change priority will be used to determine which changes will and will not be implemented, and/or to settle issues of contention.

Team leader must be made aware of all requirements changes that are determined to require changes to project schedule, budget or resource requirements occur once the SRS is base-lined. Quantities associated with each of these items must also be reported.

5.3.2 Schedule Control Plan

As stated in section 1.1.4, i.e.*Schedule and Budget Summary*, the project will perform schedule control using the Earned Value Management System (EVMS). In addition, the Critical Path Method (CPM) will be used to control the activities most crucial to completion of the project on-schedule.

Critical Path Method (CPM)

The critical path illustrated by the network diagram (Figure 1) shall receive special attention with respect to completion on schedule. Failure to complete these activities within their allotted time will cause slippage of the entire schedule.

Bi-weekly examination of the critical path will be undertaken in order to account for activities that enter and leave the critical path as real progress data is entered against the baseline project schedule.

Activity completion status

In the section 5.1.1, i.e.*estimation plan*, it is stated that each activity (represented by a work package) estimate shall consist of sub-activity milestones which will be attached to the identification of how complete ("% complete") a work package is at a given point in time. Activity completion status will be reflected (and only reflected) by the meeting of these sub-activity milestones. Sub-activity milestones will be developed for each activity by the assigned resources as the depth of each activity becomes known. These milestones will be communicated to the project manager, who will work with the resource to attach a "% complete" value to each milestone so that the progress of each activity may be understood.

Major Milestones

The following major milestones and associated completion identifiers are defined in Table 6:

Table 6: Major Milestone

ID	Milestone	Date	Complete When...
1	Project initiation		
2	Infrastructure complete		
3	Plugins complete		
4	SDK complete		
5	Initial release		

The arrival of major milestones will be treated specially from an effort data collection perspective. With the completion of each major milestone, all team members will be expected to update their effort data, as per section 5.3.5, i.e.*Metrics Collection Plan*, so that a milestone performance report may be issued; milestone status and performance data will then be updated on the performance reporting site (see section 5.3.4, i.e.*Reporting Plan*).

Collection of progress data

At the regular weekly project status meetings, where applicable, the participants will each identify the sub-activity milestones that have been met by the work for which they are responsible for during the period in which the status meeting falls. This information will be correlated with the "% complete" value attached to the milestone and the latter value will be entered into the project plan.

Effort data are collected according to the method outlined in section 5.3.5, i.e.*Metrics Collection Plan*. These data will be used as inputs into progress measurement analysis.

5.3.3 Quality Control Plan

This subsection will describe the mechanisms to be used for measuring and controlling quality of work processes and products. Each of the mechanisms mentioned here are described in more detail in section 7.3, i.e.*Quality Assurance Plan*.

Audits

Audits of work processes will not be conducted on a schedule. However, they will be requested by the team leader and maybe one of the the following:

- Project Manager
- Quality Assurance Manager

When requested, audits will be carried out as specified in section 7.3, i.e. *Quality Assurance Plan.*

Reviews

Regularly scheduled reviews of work products will take place according to the schedule described in section 5.3.4, i.e. *Reviews and Audits Plan.*

Defect/issue tracking

Defects and other issues will be tracked with Github issue system, providing a central location for defect/issue logging and resolution status.

5.3.4 Reporting Plan

This section describes the reporting requirements for the project. Specifically, it identifies the project stakeholders, their generic information requirements, the distribution of items of communication, and the performance reporting data that will be communicated during the project.

Stakeholders

The stakeholders in the project are as follows:

- Kreogist Dev Team
- Project Manager

Ad-hoc communication (communication included in this plan) will take place directly with recipients of the item of communication.Ad-hoc communication with hierarchy levels above the team leader will be made through the team leader. Scheduled communication (communication included in this plan) will take place directly with recipients of the item of communication.

Performance Reporting

The project will report performance to plan with the following metrics:

1. Requirements
    - Requirements change count
2. Configuration

- Configuration churn

3. Quality
    - Lines of code
    - Comment percentage
    - Open defects vs. closed defects over time
4. Risks
    - Weekly risk change
    - Top 10 risks

This information will be available electronically in a format accessible by a web browser supporting the HTTP protocol.

Approvals

The team leader approval signatures are required in order to confirm consent to and validity of this reporting plan.

5.3.5 Metrics Collection Plan

This section describes the metrics that will be collected by the project and the methods that will be used to collect them. The metrics collected generally fall into one of the following three categories:

- Effort
- Reviews
- Change Requests

Effort

Effort metrics will be collected by having project team members write out reports as they work on the project. Each team member will have to write their own report and allocate time to one of the listed categories. Entry should be made at least weekly and preferably more often, especially if the team member is involved in work on more than one category in a week. This will increase the accuracy of data by reducing the impact of time on human memory of effort expended.

In order to emphasize the importance of effort metrics collection, a small percentage of every second weekly (i.e. every other week) project status meeting will be dedicated to reviewing effort metrics and those metrics to which effort metrics contribute for each week. Questionable metrics will be clarified at the meetings. The metrics will be summarized during the meeting and the information that is produced from them will be highlighted.

Reviews

Review metrics will be collected from review meeting forms, which will identify each of the reviewed problems as either "errors" or "defects". It will be the responsibility of the review minutes note taker to so identify each reviewed problem on the problem report forms. The note taker will also enter the metrics into the Github issue system.

Change Requests

Change request metrics will be processed by team leader manually.

5.4 Risk Management Plan

This section mentions a number of possible risks for the project. Also, actions or measures are described to prevent or to reduce the risks.

Four categories of risks are identified:

- Risks with respect to the work to be done;
- Risks with respect to the management;
- Risks with respect to the resources;
- Risks with respect to the customer.

The risks for each category are listed below. For each risk, a description, a probability to occur, the action associated and the impact of the risk are given.

5.4.1 Risks with respect to the work to be done

We only discuss the most important risks.

Miscommunication

Probability Medium

Prevention After a meeting, one group member creates an interview report. Every participant and every person who should have been a participant of the meeting should get a copy of this report. Team members should not hesitate to ask and re-ask questions if things are unclear.

Correction When it becomes clear that miscommunication is causing problems, the team members involved and the customer are gathered in a meeting to clear things up.

Impact High

Time shortage

Probability High

Prevention Care is taken to plan enough spare time.

Correction When tasks fail to be finished in time or when they are finished earlier than planned the project planning is adjusted. If time shortage becomes severe, user requirements, which have low priority, are dropped after consultation with the project manager and the customer.

Impact High

Design Errors

Probability Medium

Prevention The design should be reviewed very critically. The advisor should be consulted frequently on his opinion about the feasibility and the correctness of certain design decisions.

Correction When errors in the design are noticed the advisor should be consulted to help correct the design errors as soon as possible. Also all the work, that depends on the faulty design, should be halted until the error is corrected.

Impact High

Illness or absence of team members

Probability High

Prevention Team members should warn their team leader or the PM timely before a planned period of absence.

Correction By ensuring that knowledge is shared between team members, work can be taken over quickly by someone else if a person gets ill. When work needs to be taken over by someone a re- division is made of his other tasks so that the workload does not get too high. Planned absence is dealt with in the planning.

Impact Medium

Server crash

Probability Low

Prevention All products are stored in the project repository, which is backed up by all members at local.

Correction When a product gets lost from its working store it is recovered from the most recent backup.

Impact Medium

5.4.2 Risks with respect to management

Illness or sudden absence of the project manager

Probability Low

Prevention There are very few things in which the presence of the PM cannot be missed for a short period of time. Nevertheless the project manager will inform the team leader of a planned period of absence in time so that the team leader can prepare to take over.

Correction By keeping the team leader up-to-date on the project status he will have enough knowledge to take over in case of illness or absence of the project manager.

Impact Low

5.4.3 Risks with respect to resources

Unavailability of the technical advisor when needed

Probability Medium

Prevention Meetings with the technical advisor can be planned in advance and time has been reserved in his schedule for counseling the team.

Correction A different appointment is made, or another expert is consulted.

Impact Medium

5.4.4 Risks with respect to the customer

The customer changes his mind about the requirements

Probability High

Prevention It is obviously explained to the customer, that after he has accepted a version of the SRS, the SRS cannot be changed by the customers wish only.

Correction If the customer changes his mind during the requirement phase his new requirements can be incorporated in the SRS. Procedures described in SQAP detail how the SRS may be changed after approval, and how to implement these changes.

Impact High

The customer is not available when needed

Probability Medium

Prevention Meetings with the customer can be planned well in advance. The customer has been given room in his schedule for his Software Engineering related work.

Correction When the customer is not available, meetings may have to be rescheduled.

Impact Medium

5.4.5 Summary

It is obvious that problems will occur during the project. To avoid problems the following rules should be followed by all team members:

- Try to signal problems as early as possible and report them to the PM, so that action can be taken;
- Pay attention to communication and make sure everybody understands the things the same way;
- Focus on the agreed user requirements, which express the wishes of the customer;
- Minimize friction between people by helping and supporting each other;
- Follow guidelines that are posed in SQAP to aid coordination and to ensure product quality.

5.5 Project Closeout Plan

This section describes the nature of the activities that will be used to closeout the project when it is completed.

As noted in the closure checklist, project participants will be gathered for the purpose of a Post-Performance Analysis (PPA). The PPA allows data to be gathered about their performance and experiences so that project processes may be tuned to improve performance on future projects (see section 7.6, i.e.*Process Improvement Plan* for how process improvements will be implemented). The PPA will also be used at the end of project phases (also expanded in section section 7.6, i.e.*Process Improvement Plan*).

The PPA will be carried out according to the following seven-step process:

1. PPA meeting invitation will be sent to project participants. The invitation will include an instruction to assemble all available data from the following categories so that it may be collected and archived:
    - dimensional data for all work products (how many, how big, how often produced, etc.)
    - lessons learned (risk logs, correspondence, etc.)
    - change requests (requirements, specifications, etc.)
    - time and effort data (estimates in hours/dollars for WBS items, networks, schedules, etc.)
    - questionnaire responses (based on distributed questionnaire; a different questionnaire for team leaders and team members will be used)
2. Allow sufficient time for the team to assemble their data and formulate responses with a report.
3. Assemble the team for a PPA meeting
    - Before starting, communicate to participants that the meeting is for data collection and not finger pointing
    - The meeting will be short and tightly focused on data collection
4. Additional meetings will be held until everyone who has something to contribute (i.e. all sources of information and materials) has made their contribution
5. Collected data and material will be categorized into one of the following two categories
    - Process data
    - Product data
6. Collected material will be categorized into one of the following two categories:
    - For archive
    - For disposal
7. A concise report of the PPA results will be published to summarize the findings, with the following goals:
    - The report will link to as many of the archived documents as possible
    - The report will be easily accessible to all project managers so that any contained information can be used to augment future projects

6 Technical Process Plans

6.1 Process Model

List of processes not used

The following IEEE 1074 processes were not used in the Software Life Cycle Process (SLCP) constructed for this project:

- **Concept Exploration**
  Since this project is a subproject contracted to Kreogist Dev Team, as part of a larger project initiated by Kreogist Dev Team, we were presented with what was to be delivered. A concept exploration was not necessary because the concept had already been explored by Kreogist Dev Team.

- **Retirement**
  Retirement of the software is not within the scope of what Kreogist Dev Team was asked to do. The project only exists for the purpose of delivering Mail application to Kreogist Dev Team.

List of processes used, but not elaborated

- **Maintenance**
  Requirements for maintenance were provided by Kreogist Dev Team.

- **Operation and Support**
  Kreogist Dev Team will operate or support the Mail application.

Process model diagram

The process model for Mail is our original agile model. It's a kind of Scrum, but we simplified it and focus on product itself. The process model diagram is included as Figure 2.

The project is divided into several loop. Each loop will be divided into four phases. These phase are:

- SC (sprint check) phase
- TR (transfer) phase
- MG (merge) phase
- RA (reanalysis) phase

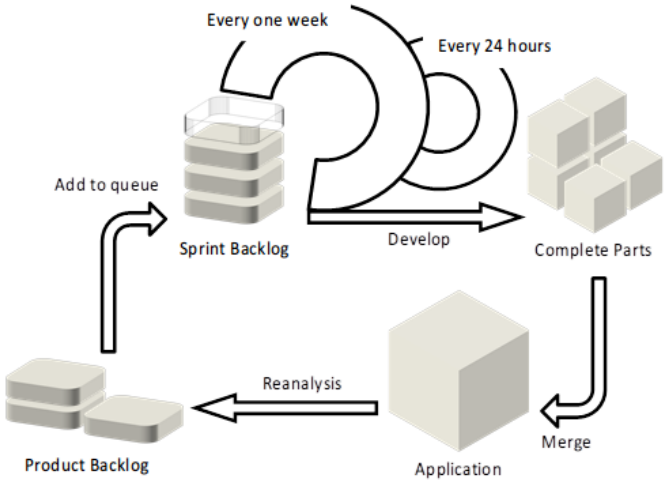

Figure 2: Mail Developing Model

In SC, team determine all the backlogs which will be done in this sprint. Once all the backlogs are written on the list, no one could add anything more on the list. In TR, team will ensure that all the parts merge to server should be fully tested and guarantee the quality of the codes. All the modules will be merged to master branch after checking in MG. All of the members should test the product and add new ideas to product backlogs. Documents might be changed at any time of the development.

Individual process diagrams

Each of the process boxes on the "Process model diagram" are treated with individual diagrams to overcome the shortcomings of the larger diagram introduced due to layout space. These diagrams are included in Figure 2.

6.2 Methods, Tools, and Techniques

Development methodology

We will use our own agile development methods, called Agile Incremental Methodology.

The decision to use the waterfall methodology is due to the following characteristics of the project:

- The product definition is unstable;
- Requirements and implementation of the product are not very clearly understood;
- Technical tools and hardware technology are familiar but we don't have any experience or any products could reference;

- Agile Incremental Methodology has proven successful for projects of this nature performed by Kreogist Dev Team and the consulting firm we use (Kreogist Mu, Kreogist Cuties) in the past.

The Software Project Management Plan (SPMP) shall be based on the IEEE Standard for Software Project Management Plans (IEEE 1058-1998).

Development techniques

The requirement passed down to this project from the larger Kreogist project is that the software be based on an open architecture using a cross-platform platform. This architecture allows us to use object-oriented methods and tools for analysis, design, and implementation. We will use Object Modeling Technique (OMT) for this purpose.

Tools

The following work categories will have their work products satisfied by the identified tools:

- **Team member desktop foundation**
  1. Microsoft® Windows® platform
     - Microsoft® Windows® 7 Ultimate (64-bit)
     - Oracle® VM VirtualBox [virtual machine support]
     - Microsoft® Office 2010 productivity application suite
     - CTeX suite [LaTeX support]
     - Foxit Reader 3.2 [viewing PDF files]
  2. Mac OS X platform
     - Mac OS X 10.9 (Mavericks) / 10.11 (El Capitan)
     - iWork productivity application suite
     - Microsoft® Office for Mac 2016 productivity application suite
     - MacTeX suite [LaTeX support]
     - Adobe Acrobat [creating/viewing PDF files]
     - Preview application [viewing PDF files]
  3. Linux platform
     - Ubuntu 14.10
     - LibreOffice 4.4 [Image creating]
     - PDF viewer application [viewing PDF files]
- **Project management**
  1. Github repository management

2. TechLauncher

- **Document publishing**
  1. CTeX suite [document preparation and revision]
  2. MaxTeX [document preparation and revision]
- **Quality**
  1. Sonar@OSC [OSChina provides Sonar Qube quality management platform]
- **Design**
  1. Adobe Photoshop CS 6.0 [Image creating]
  2. Adobe Photoshop CC [Image creating]
  3. LibreOffice 4.4 [Image creating]
- **Implementation**
  1. Qt 5.5 [programming language and object code generation]
  2. Qt Creator 3.0+ [development tools]
  3. GNU Debugger [debugging tools]
  4. Microsoft Visual Studio 2013 [64-bit Windows environment compiler]
  5. XCode 5/6+ [64-bit Mac OS X environment compiler]
  6. GNU Compiler Collection [32/64-bit Linux environment compiler, 32-bit Windows environment compiler]

Document distribution

All documents distributed electronically will first be created in the Adobe PDF format using LaTeX PDF compiler.

Change management policy

Once a work product has been finalized and approved, all changes to that work product must be submitted to the project manager via E-mail, where the changes will be reviewed and either approved or denied by the project manager, based on the risk profile and perceived benefit of the change to be made. Changes that are approved may be implemented against the work product, while changes that are denied must not take place against the work product.

For internal changes (those changes that originate within the project or within Kreogist Dev Team), the severity and potential impact of the result of not implementing the change will be measured against the disruptiveness of implementing the change. In particular, changes that are 30 working days or less downstream of the approval of the work product that they are changing will be treated more liberally than those that are more than 30 working days downstream.

For external changes (those changes that originate outside of the project and outside of Kreogist Dev Team), negotiation will take place with respect to the budget and schedule changes that will be required in order to implement the requested change.

6.3 Infrastructure Plan

Physical server access

All the repository will be hosted on Github. We will make a backup at Git@OSC for emergency backup. All the member will contains a clone for the code repository.

Network configuration

Configuration management will be provided by third party and management by project manager.

Software licensing

Licenses for operating systems and software tools are recycled between projects. We will choose all the open-source license which is capable with GNU General Public License version 2 or later.

6.4 Product Acceptance Plan

This section will describe the methods of acceptance for each of the project deliverables identified in section 1.1.3, i.e.*Project Deliverables*; the headings of this plan relate to the deliverable categories in that section. Acceptance of work products is ultimately achieved when approval is granted by the person with such responsibility, as described in section 4.2, i.e.*Internal structure, Roles and Responsibilities.*

Project documentation

All "project documentation" items, with the exception of the SPMP, are approved by the project manager and will be reviewed by both the project manager and the person with lead authority in production of the document to ensure that the document meets all of the requirements of the phase into which it will be fed.

The SPMP will be approved by Kreogist Dev Team.

Software program and library binaries

When the software program and library binaries are ready to be installed on the target hardware, the project manager will hold a review session with Kreogist Dev Team steering committee in order to report outstanding known issues with the software, once testing has been completed. The Kreogist Dev Team steering committee will deliver a decision to the project manager on whether or not the list of issues is acceptable for

procession with installation. If the list is not acceptable, the Kreogist Dev Team steering committee will work with the project manager to reduce the list of issues to a list that is acceptable to Kreogist Dev Team. Acceptance will occur when this list is satisfactory to the Kreogist Dev Team steering committee.

Installation of software program and library binaries on target hardware

Installation of the software products on the target hardware is the final project deliverable. All the testing will be done with the Kreogist Dev Team members. A review committee will be hold after one platform is fully tested. Acceptance will occur when all the platform is done and there's no known issue during the test.

Software source code

All the source code will be provided by Kreogist Dev Team and released as GNU General Public License version 2 or later on Github even during the development state.

Software documentation

Technical Documentation will be generated with Doxygen from the source code. Kreogist Dev Team should provides a compile documentation for different platforms. Acceptance will occur when the compile documentation could be done without any problems.

Approvals

The team leader approval signatures are required in order to confirm consent to and validity of this reporting plan.

7 Supporting Process Plans

7.1 Verification and Validation Plan

This section briefly describes the Verification and Validation (V&V) approach for the project. Further detail will be provided by the external Software Verification and Validation Plan (SVVP), when it is developed by Quality Assurance Manager.

Scope

Formal validation and verification will be performed on the following project work products and are listed below in order of occurrence:

1. Software requirements
2. Software architecture
3. Software interface design
4. Database design
5. Implemented software interfaces

The main V&V activities performed on these work products will be inspections and reviews. Audits may also be performed on request.

All other work products will be informally verified and validated to some degree, but they will not receive formal verification and validation from the verification and validation team members.

Responsibilities

The verification and validation team consists of the following resources:

- Quality Assurance Manager (Lead)
- Team Leader
- All the team members

Each of the validation and verification activities are included in the project work activities(see section 5.2.1). The specific responsibilities of resources and resource collaborations are identified in section 4.2, i.e.*Internal structure, Roles and Responsibilities.*

The team "Lead", identified above, has responsibility for focusing and coordinating the V&V effort of each resource listed in this section and is ultimately responsible for the outcome of the activities of the team.

Tools & Techniques

Each of the items listed in the "Scope" subsection of this section will be verified and validated to ensure that they account for all items in the products of the preceding activity. The first item, which has no precedent, will be verified and validated against documented customer meetings to ensure that all requirements are included in the SRS.

Tracing will be used to trace the existence of features between phases back to the original requirements and avoid the introduction of unnecessary work into the products. In particular, the following will be traced:

1. User requirements to software requirements
2. Software requirements to interface requirements
3. Architecture requirements to interface requirements
4. Interface requirements to database requirements
5. Software tests to interface requirements
6. Acceptance tests to user requirements

The information produced by tracing will be used during software inspections. Software inspections will ensure that work products are faithfully representing the goals set out for them by the predecessor documents.

Black-box and black-white-box-mixing testing will be performed on the implemented software interfaces to ensure that the outputs of each interface are consistent with what is input, based on the interface design.

The following tools will be used to assist with V&V:

- Valgrind
- Github issue repository tracing system.

Reviews

Regular peer reviews will be held to review in-progress work products. The procedure for scheduling these reviews is included in section 7.4, i.e.*Reviews and Audits Plan.*

Reporting

For each verification and validation of a configuration item, a corresponding report will be issued by the team. The report will consist of:

- unique report ID
- problems discovered, and, if known, corresponding solutions
- acceptance or rejection of the item (rejections should be explained)

7.2 Documentation Plan

This section describes the documentation plan for the projects deliverable and non -deliverable documentation work products. All deliverable work products appear in section 1.1.3, i.e.*Project Deliverables*.

The table headings are defined as follows:

- **Document**: the documentation work product described by the remaining columns in the row
- **ID**: the documentation work product identified number
- **Template/Standard (T/S)**: the template or standard on which the document is based (may be organizational or external). See section 2, i.e.*References* for template/standard details.
- **Reviewer**: the person responsible for reviewing the document.
- **Distribution list**: expected recipients of the review copies and baseline versions of the document

Deliverable documentation work products

Table 7: Documentation Delivered List

Document	ID	T/S	Reviewer	Distribution list
Software Requirements Specification (SRS)	KMKOT01	IEEE Std 830-1998	Team Leader	Preparer, Reviewer, Project Manager, Team member
Software Design Specification (SDS)	KMKOT02	IEEE Std 1016-1998	Team Leader	Preparer, Reviewer, Project Manager, Team member

Continued

Document	ID	T/S	Reviewer	Distribution list
Software Project Management Plan (SPMP)	KMKOT03	IEEE Std 1058-1998	Team Leader	Preparer, Reviewer, Project Manager
Software Quality Assurance Plan (SQAP)	KMKOT04	IEEE Std 730-2002	Team Leader	Preparer, Reviewer, Project Manager
Software Verification and Validation Plan (SVVP)	KMKOT05	IEEE Std 1012-1998	Team Leader	Preparer, Reviewer, Project Manager, Team member

7.3 Quality Assurance Plan

This section will describe the plans for assuring that the quality of delivered work products is consistent with what is expected for the project. Further detail will be provided by the external Software Quality Assurance Plan (SQAP), when it is developed by Quality Assurance Manager.

Scope

The processes used to create the following products will be tracked:

1. Software Requirements Specification (SRS), ID: KMKOT01
2. Software Design Specification (SDS), ID: KMKOT02
3. Software Project Management Plan (SPMP), ID: KMKOT03
4. Software Quality Assurance Plan (SQAP), ID: KMKOT04
5. Software Verification and Validation Plan (SVVP), ID: KMKOT05
6. Software product object code
7. Software product binaries
8. End-user documentation

Reviews

Quality reviews will ensure that documentation products adhere to the standards on which they are based (as per section 7.2), and that non-documentation work products adhere to the plans/designs laid out by their input prerequisites.

Quality reviews of in-scope documentation work products will be conducted once the products are complete. Reviews of in-scope non-documentation work products will take place weekly during the periods that their production is active.

Each quality review will be in a meeting format and will require the attendance of the following participants:

- Team Leader
- Project Manager
- Quality Assurance Manager

In addition, the Lead team members of teams having involvement in the production of work products must attend.

A closure review will be held after all work products have been delivered. This review will be in a meeting format and will be for the purpose of gathering "lessons learned", and identifying process improvement opportunities.

Risk Management

SQA will assist in the following risk factors:

- **Project processes**: by ensuring process adherence, SQA will help prevent this risk factor from materializing
- **Requirements complete and clear**: by reviewing the SRS for adherence to the standard, SQA will assist in preventing this risk factor from materializing
- **Quality assurance approach**: although this risk item will depend on the quality of SQA itself, the fact that a documented approach exists should limit this risk factor to a Medium rating

Record storage

All SQA records will be stored in the project repository by Quality Assurance Manager.

7.4 Reviews and Audits Plan

This section will describe the schedule, resources, methods and procedures used to conduct project reviews and audits.

Since multiple project managers are referred to in the following tables, Manager will be used to refer to the project manager on the project described by this SPMP.

All review agendas and minutes are subject to handling as described in the documentation plan in section 7.2, i.e.*Documentation Plan*.

The table headings are defined as follows:

- **Review/Audit**: the review/audit type described by the remaining columns in the row
- **Schedule**: the schedule basis for the review meetings
- **Resources**: the resources required to participate in the review

Review and Audits List

Table 8: Review and Audits List

Review/Audit	Schedule	Resources
Joint acquirer/supplier reviews		
Software project review	6th month, 12th month	Kreogist Dev Team
Steering committee progress review	Weekly	Team Leader, Quality Assurance Manager and Team Members
Management progress reviews		
Tutor Meeting	Twice a month	Kreogist Dev Team
Developer peer reviews		
Requirements peer reviews	Weekly	Team Leader, Team members
Design peer reviews	Weekly	Team Leader, Team members
Implementation peer reviews	Daily	Team Leader, Team members
Quality assurance audits		
Project documentation reviews	Weekly, during periods when project documentation is being created	Team Leader, Team Members and Project Manager
Acquirer-conducted reviews		
Software acceptance review	Once	Kreogist Dev Team

7.5 Problem Resolution Plan

Problem reporting

All problems must be reported to the project manager using the problem reporting form designated for use on the project. When complete, the form should be submitted

electronically, via e-mail.

Problem analysis

Reported problems will be analyzed to determine the risk they pose to the project, and the short- and long-term impact they will have on project resources, schedule, and budget.

Problem reports will be analyzed against the Risk Categorization Table (see section 5.4, i.e.*Risk Management Plan*). If an existing risks status is determined to require elevation due to the problem report, this will be done. If the problem poses a new risk to the project, a new risk entry will be made to the Risk Categorization Table.

Depending on the nature and reach of the problem, the appropriate team members will be engaged to properly analyze the problem, determine resolution steps, and estimate time required to resolve the problem. Mandatory participants are:

- Team Leader
- One of the Team Member

As time is more important than budget or resources on this project, emphasis will be on determining the problem's impact on project schedule. This must include an analysis of the impact of diverting resource attention away from planned project activities toward resolving problems.

Root cause analysis will be performed on the problem if time permits and/or a serious process flaw is suspected to be the cause or to have contributed to the cause. Associated possible process improvements will be documented by Quality Assurance Manager. See section 7.6, i.e.*Process Improvement Plan* for process improvement plans.

Problem prioritizing

Based on analysis of the problems, and given that time is the most important factor on this project, the problems will be prioritized based on the extent of their impact to schedule if they are allowed to persist. The problems will be classified as follows:

- Critical (highest priority): problem will impact and/or has impacted delivery time of activities on the critical path
- High: problem has impacted and continues to impact delivery time of activities not on the critical path; will affect critical path if not resolved
- Medium: problem has an ongoing impact to schedule but is not expected to affect critical path

- Low (lowest priority): problem has/had a one-time impact, and/or is so minor that critical path will never be affected

Problem processing

All the problems will be state on the issue list on Github issue tracking system. Once a issue is submitted, we will give it some labels, highlight it as bug, problem, etc. Quality Assurance Manager will record it. One of the team member will be allocate as assignee. The assignee will be responsible for the issue.

Roles

The following table illustrates the roles of project team members in the problem resolution process:

Table 9: Roles of Members in Problem Processing

Team Function	Role(s)
Team Leader	• Allocate issue to team member. • Authors problem summary. • Check the bug and modified the code.
Team Member	• Check the bug and modified the code. • Give the report to team leader.
Quality Assurance Manager	• Must participate in problem resolution meetings. • Record all the solving process.
Project Manager	• Manage/modify the documentation when necessary.

7.6 Process Improvement Plan

This section describes plans for process improvements obtained during problem resolution and through periodic assessment of the project through PPAs.

Post-performance analysis

Post-Performance Analysis (PPA) allows data to be gathered about their performance and experiences so that project processes may be tuned to improve performance on future projects. The PPA will be carried out according to the following seven-step process:

1. PPA meeting invitation will be sent to project participants. The invitation will include an instruction to assemble all available data from the following categories so that it may be collected and archived:
    - dimensional data for all work products (how many, how big, how often produced, etc.)
    - lessons learned (risk logs, correspondence, etc.)
    - change requests (requirements, specifications, etc.)
    - time and effort data
2. Allow sufficient time for the team to assemble their data and formulate responses to the questions on the questionnaire.
3. Assemble the team for a PPA meeting
    - Before starting, communicate to participants that the meeting is for data collection and not "finger pointing".
    - The meeting will be short and tightly focused on data collection
4. Additional meetings will be held until everyone who has something to contribute (i.e. all sources of information and materials) has made their contribution.
5. Collected data and material will be categorized into one of the following two categories: Process data and Product data.
6. Collected material will be categorized into one of the following two categories: For archive and For disposal.
7. A concise report of the PPA results will be published to summarize the findings, with the following goals:
    - The report will link to as many of the archived documents as possible
    - The report will be easily accessible to all project managers so that any contained information can be used to augment future projects

The PPA procedure will be used to produce process improvements based on input from participants of project phases the have closed. Any project process improvements that may benefit the ongoing performance of this project will be considered for implementation so that they may benefit the remaining project phases. Potential changes to organizational processes that may produce benefit from PPA input will be documented and deferred for analysis independently of this project.

Problem resolution input

Project process improvements are those that result from the problem resolution efforts described in section 7.5, i.e. *Problem Resolution Plan*. If a root cause analysis is performed and justified process improvements are identified, Quality Assurance Manager will work with the project manager and other key resources directly involved with the process in question to develop changes to the problematic process. If an organizational process is at fault, a temporary workaround will be devised by the same participants which will last for the duration of the project. The problems and temporary workarounds will be documented so that the organizational process that caused the problem can be inspected to determine whether the changes used in this project may be of benefit to the organizational process. If the workaround was only of particular application to the current project, the documentation will be stored in the project repository so that future project managers will be aware of changes required in the process for projects of a similar nature in future.

Other process improvements

Process improvements, while the project is in progress, will not normally result from anything other than PPAs and problem resolution input for this project in order to keep the project focused. However, any suggestions for process improvements may be forwarded to the project manager at any time. Suggestions that are well-substantiated and supported by metrics may be considered for implementation in mid-project.

8 Additional Plans

There are no additional plans.

A Organization Charts