Software Project Management Plan (SPMP)

Project – SyncGallery +

Revision History

Date	Revision	Description	Author
29-Jan-12	1.0	Document Created	W. Ghorishi
07-Mar-12	1.1	Document Updated	W. Ghorishi

Acronyms

Acronym	Definition
SPMP	Software Project Management Plan
SRS	Software Requirements Specification
PQA	Product Quality Assurance
MRD	Marketing Requirements Document
SDD	Software Design Documentation
STP	Software Test Plan
PDM	Product Management
PJM	Project Management
ENG	Engineering
RCM	Release & Configuration Management

Table of Contents

1 Purpose

This Software Project Management Plan (SPMP) is developed for the SyncGallery + project with the intent to define the project's deliverables and deadlines in addition to describing the project organization, management and technical processes used as well as detailed work packages and deliverables that will contribute towards the completion of this project. It will also direct the development of this project by defining the high-level project objectives, milestones and deliverables, as well as the allocated resources and budget.

It should be noted that the full set of functional and non-functional requirements for this project are defined in the Software Requirement Specification (SRS) which will be referenced in this document.

2 Introduction

2.1 Project Overview

After significant research and competitive analysis conducted by product management organization, it was discovered that there is a gap in the market for an image editing application that will allow users to not only share, but also add their own creative touches to these image, using various image processing functions. Furthermore, given the high-quality cameras integrated into mobile platforms, explosive popularity of smart phones, as well as the phenomenal success of various social media applications, this project will target the development of a high-end image processing application for the mobile platform.

2.2 Project Deliverables

While the end goal of this project is development and delivery of a software application, fully compliant with the requirements defined in the related SRS, the following defines the list of deliverable as well as associated due dates.

It should be noted that the delivery of the software application without the following artifacts will be deemed incomplete and hence blocked by the Product Quality Assurance (PQA) organization from release.

Furthermore, all items released by the project team will be provided on a CD-ROM device with a clear list, description as well the revision of the deliverables, as baselined by the configuration management system.

2.2.1 Development & Planning Documentation

Document	Description	Owner	Schedule	Sign-Off Entity
MRD	Marketing Requirements Document	PDM	Jan. 15, 2012	N/A
SPMP	Software Project Management Plan document	PJM	Feb. 03, 2012	PDM/ENG/PQA
SRS	Software Requirements Specification document	ENG	Feb. 03, 2012	PDM/PQA/PJM
SDD	Software Design Documentation	ENG	Feb. 03, 2012	PDM/PQA/PJM
STP	Software Test Plan	PQA	Feb. 03, 2012	PDM/ENG/PJM
UISTP	Unit/Integration & System Test Plans	PQA	Feb. 20, 2012	PDM/ENG/PJM
SMP	Software Maintenance Plan (Phase C)	PJM	Feb. 20, 2012	PDM/ENG/PQA

2.2.2 Software Releases

Release	Description	Owners	Schedule
Pre-Alpha	Proof of concept release aimed at verifying major assumptions	ENG	Jan. 31, 2012
Alpha	Feature complete release of SW. May contain critical and/or major defects	ENG/PQA	Feb. 14, 2012

Beta	Feature complete release of SW. No critical/major defects	ENG/PQA	Mar. 06, 2012
1.0 Release	Zero-defect SW including post beta feature enhancement	ENG/PQA	Mar. 07, 2012

While ENG and PQA teams will both be held accountable for meeting the schedules and quality criteria, ENG team is expected to provide a baseline release candidate for each of the above releases to the PQA no later than 5 days prior to the release. This will allow for resolution of any blocking defects and re-iteration through the necessary PQA test cycles.

Furthermore, each release will be distributed to all stakeholders and will include the following artifacts on a CD-ROM:

- Release notes including feature readiness & defect summary
- Detailed test report including all reported defects and their resolution status
- Source code as well as build infrastructure required to create an executable image
- Loaded application, running on a Smartphone (upon request)
- Applications users' guide (not applicable to the Pre-Alpha release)

2.3 Evolution of SPMP

After the initial release of this document is reviewed and signed off by the required entities, it will be put under revision control system and a link reflecting the latest revision of this document will be made available to all stakeholders through the project website.

This plan will be maintained and reviewed by the PJM team, at each of the above milestones and for all deliverables highlighted to ensure it remains up-to-date. Furthermore, any of the stakeholders including PJM team can trigger a change to this plan, at any point in time, outside of the delivery milestones mentioned earlier. Regardless of whether the SPMP is changed due to scheduled or unscheduled reviews, any changes to it require the sign-off of all entities participating in its original revision.

2.4 Reference Material

IEEE Std 1058-1998, *IEEE Standard for Software Project Management Plans*, IEEE 1998

Team SyncGallery+ , Software Requirements Specification – Project SynGallery +, 2012

Team SyncGallery+ , Software Design Documentation – Project SynGallery +, 2012

Team SyncGallery+ , Software Test Plan – Project SynGallery +, 2012

Team SyncGallery+ , Software Maintenance Plan – Project SynGallery +, 2012

3 Project Organization

One of the important functions of the SPMP is to highlight how the project is organized in terms of its process model, organizational structure, internal and external interfaces as well as roles and responsibilities. The organization for the SyncGallery + project is described below.

3.1 Process Model

Given the aggressive schedule, need for a range of cross functional disciplines as well as the close-loop communication required, various elements of agile-scrum development methodology will be incorporated into this process.

As per scrum methodology, the team will be organized into a group of “pigs” & “chickens”, where “pigs” are the individuals that are actively contributing to the development of the project (e.g. developers, PQA engineers, etc.), while the “chickens” are the individual stakeholders that will monitor the project, but cannot interfere in its day to day operations (e.g. PDM).

The communication in the team will be governed by daily-stand-up meetings, lasting no longer than 15 minutes, covering last day’s accomplishments, current day’s activities as well as any impediments that may affect upcoming tasks. Also, one of the “pigs” in the group will be designated as the scrum-master who in addition to his individual deliverables will be responsible for resolving issues blocking the progress of other team members.

Various functional roles in the group include:

Scrum master: Responsible for defining detailed project backlog/tasks through definition of a SRS, leveraging an existing MRD. This role will also work with individual team members in assigning weights to each of the scrum backlog tasks defined. Other responsibilities include addressing impediments blocking team members, measuring performance, holding sprint reviews as well as interfacing with outside entities to obtain support needed for the project. This role remains active throughout the project life cycle.

Architects/Designers: Responsible for creating a SDD, based an already defined SRS. From time to time, there may be a need to create functional prototypes in order to validate certain assumptions made.

Developers: Responsible for taking a SDD and implementing the SW as described in that specification. Developers will also be responsible for unit-testing as well as pre and post release maintenance (defect resolution and feature enhancement) for the product.

Test Engineers: Responsible for defining a STP, functional and system-level test case development through creation of a traceability matrix going as far back as the SRS.

Verification & Validation Engineers: Leverages the test cases generated by the Test Engineering team and is responsible for black-box testing through periodic (regression) as well as release-based test cycles. This role will also act as the driving force to close all outstanding defects reported. Prior to availability of test cases, this team will be responsible for creating the regression test framework. Other associated process roles include ensuring adherence to quality and development standards defined.

Release & Configuration Management: Responsible for maintaining baselines and ensuring the integrity of the releases delivered to Verification & Validation Engineers. The RCM function will also be responsible for producing release base artifacts such as release notes and users' guide.

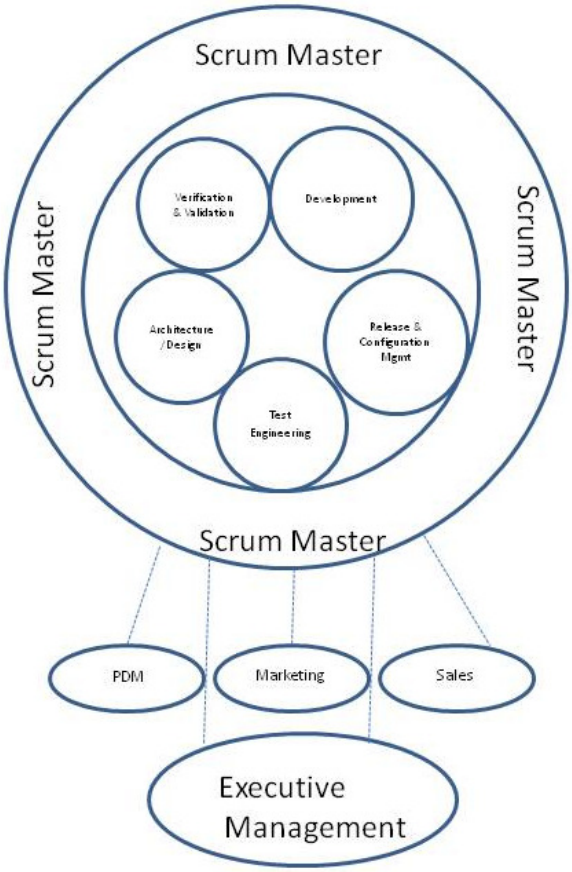


Fig 1. Project Organization Diagram

3.2 Organizational Structure

Adhering to the agile scrum methodology, the team is a self-governing entity. While each member ultimately reports to the appropriate functional managers in the organization, the project itself is managed under the guidance of the scrum master and as such all resources assigned to the project will be reporting to him/her.

3.3 Organizational Interface

To control the flow as well as accuracy of information and minimize interruptions for team members and prevent unannounced feature creeps, all external communications will be handled by the scrum

master. Of course, relevant information will be disseminated to the team members through the processes put in place (i.e. stand-up meetings, updating of scrum backlog tasks, prioritizing of sprint tasks, etc.)

3.4 Project Responsibilities

The following table shows how the resources assigned to this project map to the roles already defined:

Role	Name(s)
Scrum master	Will Ghorishi
Architect/Designer	Kobe Sun, Hitha Dinesh
Developer	Jacky Tan, Daniel Liu, Kobe Sun
V&V Engineer	Sarah Yang, Hitha Dinesh
Test Engineer	Sarah Yang, Will Ghorishi
RCM Engineer	Hitha Dinesh, Will Ghorishi

To compensate for the dynamic changes in scope and minimize risks to the delivery schedules the following re-assignments had to be made:

- Re-focus a significant portion of Hitha's bandwidth away from design to Test case development
- Re-focus a significant portion of Will's bandwidth away from test engineering to documentation
- RCM function was pushed back to the developers

4 Managerial Process

The following provides an overview of the management process used for this project, including objectives and priorities, assumptions, dependencies and constraints, risk management as well as staffing techniques applied.

4.1 Objectives and Priorities

The following capture the overall management objectives, in an ascending order of priority:

- **Market Appeal:**
    - Delivers all of the features specified by the MRD
    - The user-Interface (UI) is intuitive and easy to learn by non-technical individuals
    - Developed in a scalable manner, allowing future enhancements
- **Quality:**
    - Provides stable operation without any defects
    - Offers competitive performance level
- **Time to market:**
    - Meets the agreed upon milestones and schedules highlighted in this SPMP
- **Budget:**
    - Resources required are all identified and accounted for at the beginning of the project
    - Tools & equipment required have all been acquired prior to the start of the project

4.2 Assumptions, Dependencies and Constraints

The following is a list of assumptions made by this SPMP, after detailed discussions with various stakeholders:

- The resources assigned will remain fully dedicated to this project until completion
- The resources are fully versed on the technologies used and hence require no training
- The requirements highlighted in the Feature and Functionality Specification (FFS) document are reflective of the market demands
- The requirements will not change for the initial release of the application
- Subsequent enhancement requirements will be addressed during the post-release maintenance cycle and will be augmented to the FFS document
- Any changes in scope of the project will trigger a review of the committed schedule and require sign-off of the updated SPMP by all stakeholders

4.3 Dependencies

To reduce cost, improve time-to-market and increase quality and reliability, the popular and extensively stable open-source software packages like DropBox (cloud file storage) and Aviary (image processing) will be used in the development of SyncGallery+.

4.4 Constraints

The initial release of SyncGallery+ will only target mobile platforms running Android Ice-cream Sandwich operating system and Google+ social media. However, the application architecture and design will enable expansion into other platforms.

The SyncGallery+ project is a resource and time constraint initiative. Features will need to be well managed to ensure on-time and on-budget delivery.

4.5 Risk Management

The following chart highlights the major risks, assigns a probability of occurrence, magnitude of impact (0% - No real impact, 25% - Minimal Impact, 50% - Considerable Impact, 75% - Severe Impact, 100 % - Project Viability Impact) and mitigation steps taken to reduce the exposure of SyncGallery+ project to them. These risks management plan was developed prior to the start of the project.

ID #	Risk Description	Probability %	Magnitude %	Mitigation Steps
1	Resource Bandwidth	50	100	-Addition of a 6th resource -Frequent Agile stand-up meetings to review deliverables
2	Resource Retention	50	100	-All resources have committed to stay within the organization for the duration of the SyncGallery+ project
3	Offsite PM	100	25	-Use of conference bridge & other social networking tools (skype) -Frequent communication & coordination meetings
4	Lack of image processing experience	100	50	-Use of a 3rd party open source integratable stack called Aviary which supports basic image editing & image processing functions
5	Lack of experience with Cloud Computing	100	50	-Use of a 3rd party open source integratable library called Dropbox
6	Lack of experience with Android OS	100	50	-Evaluation of other open source applications developed for Android
7	Tools & Development Environment	25	75	-Good set of publicly available development environment for Android -Developers/testers who do not have an Android ICS phone will focus on using the Android Emulator running on a PC
8	Platform compatibility	100	75	-Limiting the scope of the deliverables to Android mobile platforms -Support of the application only using the latest Google ICS release
9	Achieving quality objectives	50	100	-Through Software Quality Assurance Test Plan (SQATP) -Agile development methodology. Integrate & Test as functionality becomes available -1:1 Developer/Tester headcount ratio
10	Marketability	100	100	-Focus on time to market, leveraging agile methodology -Platform targetting to ensure verified quality criteria -Extendible architecture to enable ease of post release feature requests -Integration with Google+ social media application

The following is a revised risk assessment table, completed at the end of the project, reflecting the revised exposure level to each risk (notice it's no longer caller probability since it's a reflection of the past.

ID #	Risk Description	Exposure Level	Magnitude %	Mitigation Steps
1	Resource Bandwidth	100	100	-Addition of a 6th resource -Frequent Agile stand-up meetings to review deliverables
2	Resource Retention	0	100	-All resources have committed to stay within the organization for the duration of the SyncGallery+ project
3	Offsite PM	0	25	-Use of conference bridge & other social networking tools (skype) -Frequent communication & coordination meetings
4	Lack of image processing experience	0	50	-Use of a 3rd party open source integratable stack called Aviary which supports basic image editing & image processing functions
5	Lack of experience with Cloud Computing	0	50	-Use of a 3rd party open source integratable library called Dropbox
6	Lack of experience with Android OS	100	50	-Evaluation of other open source applications developed for Android
7	Tools & Development Environment	25	75	-Good set of publicly available development environment for Android -Developers/testers who do not have an Android ICS phone will focus on using the Android Emulator running on a PC
8	Platform compatibility	100	75	-Limiting the scope of the deliverables to Android mobile platforms -Support of the application only using the latest Google ICS release
9	Achieving quality objectives	100	100	-Through Software Quality Assurance Test Plan (SQATP) -Agile development methodology. Integrate & Test as functionality becomes available -1:1 Developer/Tester headcount ratio
10	Marketability	100	100	-Focus on time to market, leveraging agile methodology -Platform targetting to ensure verified quality criteria -Extendible architecture to enable ease of post release feature requests -Integration with Google+ social media application

The main items to highlight here is that:

- Resource bandwidth has definitely impacted the delivery of the RC SW on-time. This was mainly due to other commitments outside of the SyncGallery+ project as well as the stringent documentation requirements imposed by the customer
- Resource retention proved to be a non-issue as all assigned resources were committed to the project
- Leveraging telephone bridging techniques, the PM function worked flawlessly
- Lack of image processing and cloud computing experience were compensated by using 3rd party open-source libraries, namely Aviary and Dropbox and hence the exposure was completely eliminated
- Delays in delivery of the SW and extensive need for documentation reduced the level of quality assurance originally planned by the team. However, based on all testing conducted so far, the product being released is definitely RC quality.
- The team's lack of experience with marketing and resource assignments also contributed in making the marketing efforts spent on the product less than desirable

4.6 Monitoring & Control Mechanisms

Given the small sized team and aggressive schedule, agile with scrum development methodology will be used for the SyncGallery+ project. Hence all communication within the team will be done under the context of daily stand-up meetings and overall release readiness assessments will be done under the context of sprint reviews of backlog tasks and stories.

4.7 Staffing Approach

Given the fixed budget and the resources already available, there will not be any need for additional resources. The resources allocated to this project are also all fully experienced with all aspects of the technology leveraged and hence no additional training is required.

5 Technical Process

The following provides an overview of the technical process used for this project, including the development methodology, tools, internal and external documentation.

5.1 Process Model

The project will leverage the agile development methodology with scrum. This means that the team will define stories for high level milestones in the project and associate tasks to each story. The deliverables will be broken into sprints with the goal of delivering functional and qualified features at the end of each sprint. Hence the all members of the team, including test engineers as well as the verification & validation engineers will be focusing on the same set of deliverables as the deliverables in the team.

At the end of each sprint, a sprint review meeting will be held and the progress will be measured not just in terms of tasks completed from the backlog, but also the readiness (code complete and QA'ed) of the features committed for the sprint.

During the course of the sprint, there will be daily 'stand-up' meetings focused on each team member reporting what they accomplished the day before, what is their plan for the day and if they have any impediments or dependencies that will need to be addressed by the scrum master.

5.2 Tools

The source code for this project including all open-source elements as well as third party binaries leveraged will be kept under strict revision control system, provided by SVN. Furthermore, all defects reported by the quality assurance function (verification and validation engineers) will be logged and tracked using TRAC. Also, to avoid having to have all team members purchase an Android ICS mobile phone, the Android emulator running on a WIN8 platform will also be used for development and testing of the SyncGallery+ project. Of course, each of the releases will undergo the complete test cycle on an actual platform.

5.3 Documentation

The following is a list of all documentation as mandated by the technical process

5.3.1 Software Requirements Specifications (SRS)

The Software Requirements Specification provides a concise description of each of the functional, performance and UI attributes of the software. These requirements will ultimately be reference in a traceability matrix to validate their achievement. This will be a separate document and hence only referenced in the SPMP

5.3.2 Software Design Description (SDD)

This specification defines all major components of the software as well as how they interface with each other. The overall architecture of the software is also described in this document. This will be a separate document and hence only referenced in the SPMP.

5.3.3 Software Test Plan (STP)

The Software Test Plan describes the methods used to validate requirements at all levels in the development cycle including architecture & design (referencing SDD), implementation (referencing code modules), integration and system validation (referencing functional requirements ultimately tracing back to the SRS)

5.3.4 User Documentation

This includes the planning for user support, including users' manuals, online help infrastructure as well as other user support facilities.

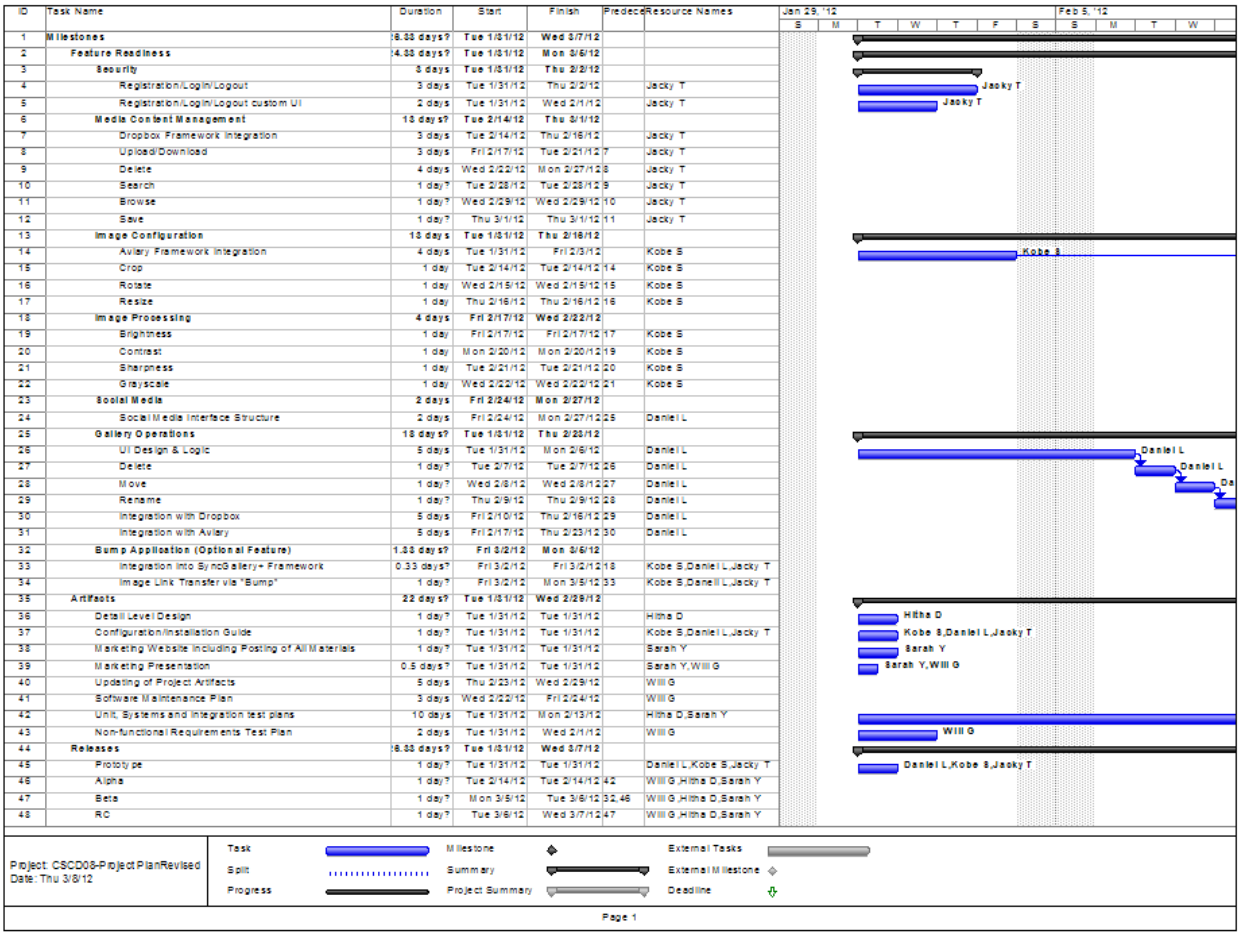
5.3.5 Project Support Functions

Each of the functional elements in the project will publish their own support plan for this project. The minimum requirement for each of these plans is to identify owners and resources for that function in the organization and determine the budget allocated to this project. The above highlights the list of mandatory plans for the SyncGallery+ project.

6 Work Packages, Schedules and Budgets

6.1 Work-Breakdown Schedule, Milestones & Releases

The detailed work packages/milestones as well as releases delivered for this project are as follows:



ID	Task Name	Duration	Start	Finish	Predece	Resource Names
1	**Milestones**	18.33 days?	Tue 1/31/12	Wed 3/7/12		
2	**Feature Readiness**	14.33 days?	Tue 1/31/12	Mon 3/5/12		
3	**Security**	3 days	Tue 1/31/12	Thu 2/2/12		
4	Registration/Login/Logout	3 days	Tue 1/31/12	Thu 2/2/12		Jacky T
5	Registration/Login/Logout custom UI	2 days	Tue 1/31/12	Wed 2/1/12		Jacky T
6	**Media Content Management**	13 days?	Tue 2/14/12	Thu 3/1/12		
7	Dropbox Framework Integration	3 days	Tue 2/14/12	Thu 2/16/12		Jacky T
8	Upload/Download	3 days	Fri 2/17/12	Tue 2/21/12	7	Jacky T
9	Delete	4 days	Wed 2/22/12	Mon 2/27/12	8	Jacky T
10	Search	1 day?	Tue 2/28/12	Tue 2/28/12	9	Jacky T
11	Browse	1 day?	Wed 2/29/12	Wed 2/29/12	10	Jacky T
12	Save	1 day?	Thu 3/1/12	Thu 3/1/12	11	Jacky T
13	**Image Configuration**	13 days	Tue 1/31/12	Thu 2/16/12		
14	Aviary Framework Integration	4 days	Tue 1/31/12	Fri 2/3/12		Kobe S
15	Crop	1 day	Tue 2/14/12	Tue 2/14/12	14	Kobe S
16	Rotate	1 day	Wed 2/15/12	Wed 2/15/12	15	Kobe S
17	Resize	1 day	Thu 2/16/12	Thu 2/16/12	16	Kobe S
18	**Image Processing**	4 days	Fri 2/17/12	Wed 2/22/12		
19	Brightness	1 day	Fri 2/17/12	Fri 2/17/12	17	Kobe S
20	Contrast	1 day	Mon 2/20/12	Mon 2/20/12	19	Kobe S
21	Sharpness	1 day	Tue 2/21/12	Tue 2/21/12	20	Kobe S
22	Grayscale	1 day	Wed 2/22/12	Wed 2/22/12	21	Kobe S
23	**Social Media**	2 days	Fri 2/24/12	Mon 2/27/12		
24	Social Media Interface Structure	2 days	Fri 2/24/12	Mon 2/27/12	25	Daniel L
25	**Gallery Operations**	18 days?	Tue 1/31/12	Thu 2/23/12		
26	UI Design & Logic	5 days	Tue 1/31/12	Mon 2/6/12		Daniel L
27	Delete	1 day?	Tue 2/7/12	Tue 2/7/12	26	Daniel L
28	Move	1 day?	Wed 2/8/12	Wed 2/8/12	27	Daniel L
29	Rename	1 day?	Thu 2/9/12	Thu 2/9/12	28	Daniel L
30	Integration with Dropbox	5 days	Fri 2/10/12	Thu 2/16/12	29	Daniel L
31	Integration with Aviary	5 days	Fri 2/17/12	Thu 2/23/12	30	Daniel L
32	**Bump Application (Optional Feature)**	1.33 days?	Fri 3/2/12	Mon 3/5/12		
33	Integration into SyncGallery+ Framework	0.33 days?	Fri 3/2/12	Fri 3/2/12	18	Kobe S,Daniel L,Jacky T
34	Image Link Transfer via "Bump"	1 day?	Fri 3/2/12	Mon 3/5/12	33	Kobe S,Daniel L,Jacky T
35	**Artifacts**	22 days?	Tue 1/31/12	Wed 2/29/12		
36	Detail Level Design	1 day?	Tue 1/31/12	Tue 1/31/12		Hitha D
37	Configuration/Installation Guide	1 day?	Tue 1/31/12	Tue 1/31/12		Kobe S,Daniel L,Jacky T
38	Marketing Website including Posting of All Materials	1 day?	Tue 1/31/12	Tue 1/31/12		Sarah Y
39	Marketing Presentation	0.5 days?	Tue 1/31/12	Tue 1/31/12		Sarah Y,Will G
40	Updating of Project Artifacts	5 days	Thu 2/23/12	Wed 2/29/12		Will G
41	Software Maintenance Plan	3 days	Wed 2/22/12	Fri 2/24/12		Will G
42	Unit, Systems and Integration test plans	10 days	Tue 1/31/12	Mon 2/13/12		Hitha D,Sarah Y
43	Non-functional Requirements Test Plan	2 days	Tue 1/31/12	Wed 2/1/12		Will G
44	**Releases**	18.33 days?	Tue 1/31/12	Wed 3/7/12		
45	Prototype	1 day?	Tue 1/31/12	Tue 1/31/12		Daniel L,Kobe S,Jacky T
46	Alpha	1 day?	Tue 2/14/12	Tue 2/14/12	42	Will G,Hitha D,Sarah Y
47	Beta	1 day?	Mon 3/5/12	Tue 3/6/12	32,46	Will G,Hitha D,Sarah Y
48	RC	1 day?	Tue 3/6/12	Wed 3/7/12	47	Will G,Hitha D,Sarah Y

6.2 Dependencies

1		– Dependencies	14 days	Mon 1/16/12	Thu 2/2/12
2		– Development Server	0 days	Tue 1/31/12	Tue 1/31/12
3		Server Setup	0 days	Tue 1/31/12	Tue 1/31/12
4		Account Creation For Team	0 days	Tue 1/31/12	Tue 1/31/12
5					
6		– Tools	13 days	Mon 1/16/12	Wed 2/1/12
7		– Source Repository	13 days	Mon 1/16/12	Wed 2/1/12
8		SVN Installation	1 day	Mon 1/16/12	Mon 1/16/12
9		Project & Source Tree Creation	1 day	Tue 1/31/12	Tue 1/31/12
10		Team Review	1 day	Wed 2/1/12	Wed 2/1/12
11		– Defect Tracking	2 days	Tue 1/31/12	Wed 2/1/12
12		Trac Installation	0 days	Tue 1/31/12	Tue 1/31/12
13		Project Creation	1 day	Tue 1/31/12	Tue 1/31/12
14		Team Review	1 day	Wed 2/1/12	Wed 2/1/12
15		– Development Environment	5 days	Mon 1/16/12	Fri 1/20/12
16		Android ICS framework & Emulator Insalla	5 days	Mon 1/16/12	Fri 1/20/12
17					
18		– External Services	14 days	Mon 1/16/12	Thu 2/2/12
19		– Cloud	14 days	Mon 1/16/12	Thu 2/2/12
20		DropBox Evaluation	5 days	Mon 1/16/12	Fri 1/20/12
21		Account setup/Configuration	1 day	Tue 1/31/12	Tue 1/31/12
22		DBMS Prototype	2 days	Wed 2/1/12	Thu 2/2/12
23		– Social Media	5 days	Mon 1/16/12	Fri 1/20/12
24		Interface Evaluation	5 days	Mon 1/16/12	Fri 1/20/12
25					

6.3 Budget

The following resources have been fully assigned to the SyncGallery+ project and are expected on average to dedicate 2 hour-days worth of effort to this project. The list of resources includes:

Kobe Sun – Product Management & Architecture & Implementation

Hitha Dinesh – UI Design & Verification & Validation & Test Engineering

Daniel Liu – Product Management, Architecture & Implementation

Jacky Tan – Product Management, Architecture & Implementation

Sarah Yang – Verification & Validation & Test Engineering & Product Marketing

Will Ghorishi – Project Management & Documentation & Test Engineering

6.4 Major gaps & learnings related to the original schedule

The following highlights some of the major observations and learnings as the schedule was re-baselined within the project to reflect the current status of the deliverables:

- Lack of understanding of Google+ & Facebook interfaces, complications in their integration and mid-project SRS-level change to support a generic Social Media Interface extended the number of resources. Learnings & Corrective Steps: Need to create an advanced technology team to be familiar with the technologies that will be integrated into future products.

- Significant weight of and stringent requirements for the artifacts to be produced for the project took away from the available resource pool for the SQA and development team and hence put additional load on the remaining resources. Learnings & Corrective Steps: The level of documentation and process required should be made proportional to the scope of the project. Unfortunately, the customer's requirements were the driver behind this extreme level of documentation requirements and hence short of negotiating this with the customer for future projects, nothing can be done.

- Availability of sufficient Android ICS platforms reduced the SQA cycles spent on the actual platform and further delayed the schedule. Learnings & Corrective Steps: Future projects will need to budget for all HW/SW needs of the team.

- Last minute SRS-level change from NFC-Beam support to the "Bump App" for image link transfer between two phones extended the delivery of RC beyond the original Mar. 2nd deadline. This was again due to lack of adequate HW platforms. Learnings & Corrective Steps: Future projects will need to budget for all HW/SW needs of the team.