LarkDetect

[Project Plan]

Version 1.3

By: Ang Zhan Phung, Ang Yong Xin, Brenda Ng Xin En, Emmanuelle Vania, Sam Jian Shen, How Mo Xuan

Revision History

Revision Number	Date	Primary Author(s)	Comments
1.0	Feb 25, 2020	Ang Zhan Phung, Ang Yong Xin, Brenda Ng Xin En, Emmanuelle Vania, Sam Jian Shen, How Mo Xuan	First version
1.1	March 3, 2020	Ang Zhan Phung Edmund	Amendment of Work Package and Cost Estimate
1.2	March 3, 2020	Emmanuelle Vania	Amendment of Risk Management
1.3	March 6, 2020	Brenda Ng Xin En	Amendment of Work Package and Gantt Chart

Table of Contents

1 Introduction

1.1 Project Overview

LarkDetect is an online platform that aims to provide early diagnosis for dementia. According to a study by Well-being of Singapore Elderly (WiSE), families are unprepared when their loved ones are diagnosis with dementia. By taking the Trail Making Test in LarkDetect, it will be able to determine the risk level of the user being diagnose with dementia.

1.2 Project Description and Scope

The LarkDetect is an online platform for age 40 and above where they are required to do the test annually. The system will allow the records to be stored and analyzed for individual assessment.

The basis of the LarkDetect requires users to perform two Trail Making Tests on the web platform. The following constraints have been taken into account:

- The user needs to understand English to understand the instructions.
- The user may be both a doctor and user, therefore, they will need to select the roles of the user.
- The user will be deemed inactive when they exceed the maximum session period of 3 mins. (the maximum time will have an increment of 30sec every 10 years, starting from age 40)
- The doctor will be diagnosing one user at a time.

These items are given as an example. Final possible constraints will be discussed in a later document.

Given these preferences, after the user login to their respective accounts, they can either view their past reports/statistics or do the trail making test.

The first test that the user is required to do is the Cognitive Processing Speed where the user need to connect a set of 25 dots in an increment manner as quickly as possible. Before the start of the test, the user will be provided with the instructions.

The second test is the Executive Functioning Test, where the connection of the dots would include alternating number and letters (1-A, 2-B, … L). Instructions of the test will be given before the test.

The system will record their timing taken by the user to complete each individual test. The number of errors will also be included for the doctor to diagnose.

The system will generate a report showing the individual time taken and individual error rate, the user information, and the average time taken by users of the same age group.

2 Project Organization

2.1 Team Structure

The following is the list of executive roles, as required by CMM level 3.

- Senior Management: Ang Zhan Phung Edmund
- Software Configuration Manager: Ang Yong Xin
- Software Engineering Project Group: Sam Jian Shen, How Mo Xuan
- Software Quality Assurance Engineer: Emmanuelle Vania
- Representative of Customer: Brenda Ng Xin En

2.2 Roles and Responsibilities

Project Manager: Ang Zhan Phung Edmund

- Oversees project progress
- Approves and executes project plan
- Assigns tasks and reports status of project to team members
- Manages and motivates team members
- Represents the team to the outside world

Requirement Specification Analyst: Brenda Ng Xin En

- Collects information from client interviews
- Develops concepts of the system for designers
- Creates requirement specification document

System Architect: Ang Yong Xin

- Designs logical system based on requirements
- Translates logical design into detailed design
- Creates detailed design document

Usability Analyst: Sam Jian Shen

- Designs User Interface
- Ensures stability and response time of the system meet the requirements
- Creates user manual

Software Configuration Engineer: Ang Yong Xin, How Mo Xuan

- Integration of coded modules into functioning system

Developer: Sam Jian Shen, How Mo Xuan

- Implements product based on detailed design document

Quality Assurance Engineer: Emmanuelle Vania

- Ensures acceptable software quality
- Designs testing strategies
- Creates and manages test plan
- Verify software requirements
- Executes test procedures

2.3 Team Communication

TLA communication channels include the following:

- Weekly meetings are held on Tuesdays.
- Group announcements and updates are sent through email: angz0043@e.ntu.edu.sg
    - WhatsApp group chat for informal discussion.
- Video Conference are held for urgent discussions.
- Split up into subgroups as necessary, in order to work more co-operatively on specific problems.

3 Process Definition

3.1 Lifecycle Model

Throughout the LarkDetect project, Agile lifecycle model will be used. This methodology is more flexible compared to the Waterfall model as it breaks a project down into small incremental builds which can be implemented very quickly.

This methodology is more flexible than the traditional Waterfall SDLC due to repeated iterations involving design, coding, unit testing, integration, and quality assurance. The Waterfall SDLC is not a viable choice due to the short timeline available for LarkDetect project to reach delivery quality.

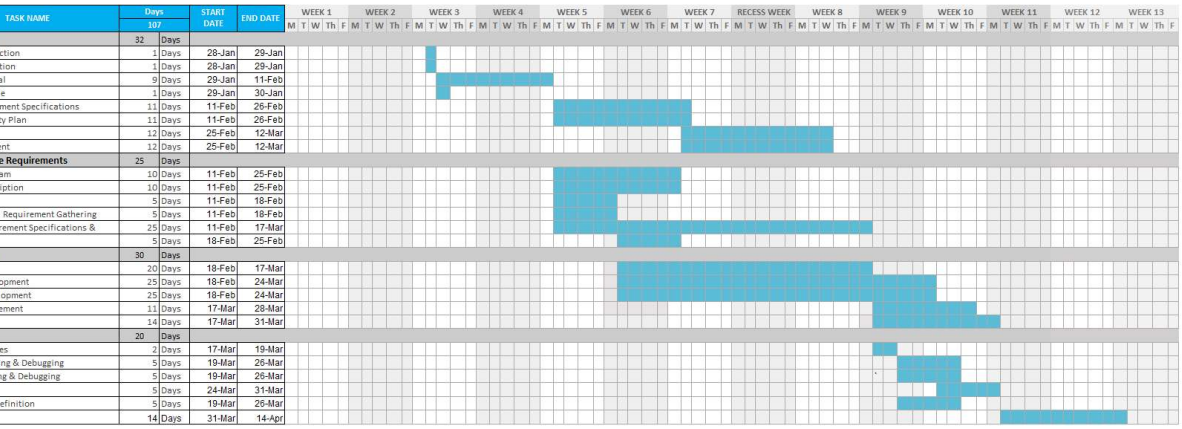
CoronaSG has chosen to avoid such methodologies as Spiral due to concerns over the short timeline. Should design procedures, for example, need to be revisited within the first release date, it is likely that the project will overshoot its critical schedule.

CoronaSG intends to deliver the first iteration of functionality on the System Delivery date indicated in the Estimations section of this document. After further client interaction, further iterations should occur as necessary.

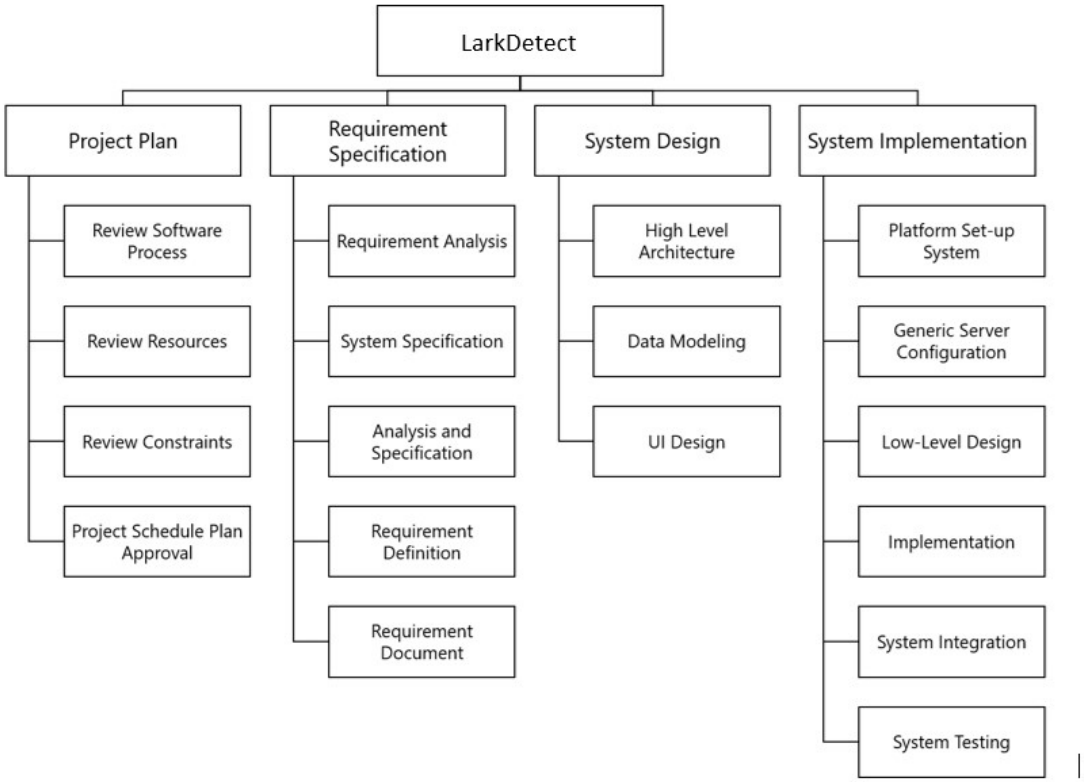
4 Schedule

4.1 Activity Dependencies and Schedule

TASK NAME	Days		START DATE	END DATE
	107			
Project Planning	32	Days		
Project Introduction	1	Days	28-Jan	29-Jan
Team Organisation	1	Days	28-Jan	29-Jan
Project Proposal	9	Days	29-Jan	11-Feb
Project Schedule	1	Days	29-Jan	30-Jan
System Requirement Specifications	11	Days	11-Feb	26-Feb
Software Quality Plan	11	Days	11-Feb	26-Feb
Project Plan	12	Days	25-Feb	12-Mar
Risk Management	12	Days	25-Feb	12-Mar
Analysis/Software Requirements	25	Days		
Use Case Diagram	10	Days	11-Feb	25-Feb
Use Case Description	10	Days	11-Feb	25-Feb
Lo-Fi	5	Days	11-Feb	18-Feb
Non-Functional Requirement Gathering	5	Days	11-Feb	18-Feb
Software Requirement Specifications &	25	Days	11-Feb	17-Mar
Dialog Map	5	Days	18-Feb	25-Feb
Implementation	30	Days		
1st Prototype	20	Days	18-Feb	17-Mar
Back End Development	25	Days	18-Feb	24-Mar
Front End Development	25	Days	18-Feb	24-Mar
Quality Management	11	Days	17-Mar	28-Mar
Test Plan	14	Days	17-Mar	31-Mar
System Testing	20	Days		
Create Test Cases	2	Days	17-Mar	19-Mar
White Box Testing & Debugging	5	Days	19-Mar	26-Mar
Black Box testing & Debugging	5	Days	19-Mar	26-Mar
Report	5	Days	24-Mar	31-Mar
CMMI Level 2 Definition	5	Days	19-Mar	26-Mar
Release Plan	14	Days	31-Mar	14-Apr



4.2 Work Breakdown Structure



4.3 Work Packages

The entire project work is broken down by the important phases of the software development life cycle. They include the following:

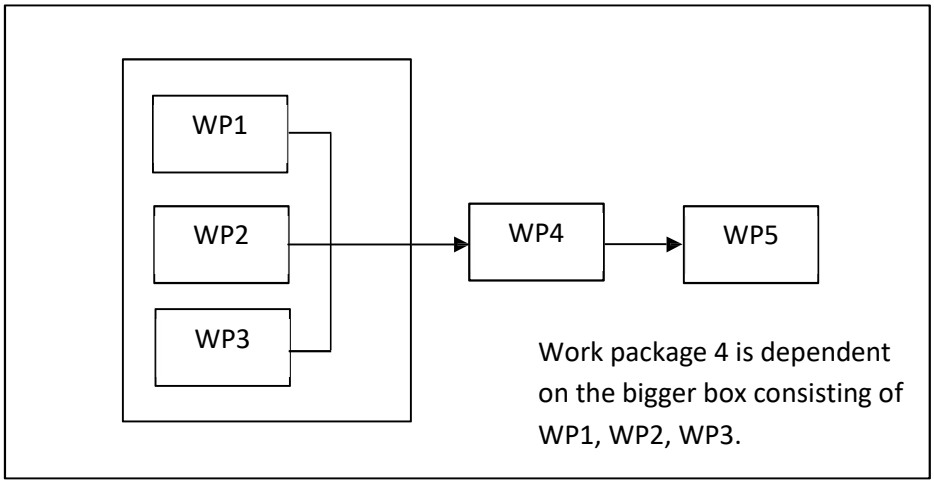
1. Project Plan
2. Requirement Specification
3. System Design
4. Coding & Unit Testing
5. Integration & System Testing

4.4 Activity Dependencies

The following table describes the dependencies of the deliverable work packages:

Work Package #	Work Package Description	Duration	Dependencies
1	Project Plan	32 days	--
2	Requirement Specification	25 days	--
3	System Design	10 days	--
4	Coding & Unit Testing	30 days	1, 2, 3
5	Integration & System Testing	20 days	4

The following Activity Network Diagram describes the above in more graphical detail:



4.5 Work Package Details

Work packages are listed below. A team member, indicated in bold, has been assigned as primarily responsible for each work package and will coordinate that package.

Project	LarkDetect
Work Package	1 — Project Plan (1 of 5)
Assigned To	**Ang Zhan Phung**, Ang Yong Xin, Brenda Ng Xin En, Emmanuelle Vania, Sam Jian Shen, How Mo Xuan
Effort	32PD
Start Date	Tuesday, 28/01/20
Purpose	To develop an overview of the project, to be refined in later work packages.
Inputs	None
Activities	This work package includes providing a brief overview of the project, the scope of the project, problem statement, technical approach and proposed project management plan for the software development cycle. The team members responsible for this work package will be recording down the details of the discussion and putting them into a written report.
Outputs	A written document of the Project Proposal.

Project	LarkDetect
Work Package	2 — Requirement Specification (2 of 5)
Assigned To	**Ang Zhan Phung**, Ang Yong Xin, Brenda Ng Xin En, Emmanuelle Vania, Sam Jian Shen, How Mo Xuan
Effort	25PD
Start Date	Tuesday, 11/02/20
Purpose	To establish a common understanding between the customer and the software project team of the customers' requirements to be addressed by the project
Inputs	Customer's requirements
Activities	This work package includes the collection of relevant information and requirements from the customer, identifying risks and determine feasibility of the project and the writing of customer requirements and build requirements.
Outputs	A written document of the System Requirement Specifications.

Project	LarkDetect
Work Package	3 — System Design (3 of 5)
Assigned To	**Ang Yong Xin**, Ang Zhan Phung, Sam Jian Shen, Brenda Ng Xin En, Emmanuelle Vania, How Mo Xuan
Effort	10PD
Start Date	Tuesday, 11/02/20
Purpose	To design a logical system for the project such that it meets the System Requirement Specification.
Inputs	System Requirement Specification (Work Package 2)
Activities	This work package includes the design of system architecture, use case and dialog map diagrams, user interfaces and databases. Software and hardware for the project are decided and documented into a design document.
Outputs	A written document of System Design.

Project	LarkDetect
Work Package	4 — Coding & Unit testing (4 of 5)
Assigned To	**Ang Yong Xin**, Ang Zhan Phung, Sam Jian Shen, Brenda Ng Xin En, Emmanuelle Vania, How Mo Xuan
Effort	30PD
Start Date	Tuesday, 18/02/20
Purpose	To implement the system design into source code through coding, according to the requirement specification and associated documents. This work package includes such additional activities as preliminary unit testing.
Inputs	System Requirement Specifications (Work Package 2), System Design (Work Package 3)
Activities	Programmers will implement the modules according to the design specifications noted in the Specification document.
Outputs	Source code and header files

Project	LarkDetect
Work Package	5 — Integration & System Testing (5 of 5)
Assigned To	**Emmanuelle Vania**, Ang Yong Xin, Ang Zhan Phung, Sam Jian Shen, Brenda Ng Xin En, How Mo Xuan
Effort	20PD
Start Date	Tuesday, 17/03/20
Purpose	To integrate all modules of the system and perform testing. Identify bugs, errors and defects, checking for interoperability. Fix all identified defects and repeat testing to ensure that developed systems meet the technical requirements.
Inputs	Coding & Unit testing (Work Package 4)
Activities	The Integration and testing team creates the test cases for System Testing, performing White Box and Black Box Testing & Debugging, checking every possible outcome. After all testing is completed, the team will document all results into a Test Analysis Report.
Outputs	A test report.

5 Project Estimates

5.1 Code Size Estimation using Function Points

We calculated unadjusted function point based on the complexity of functions provided by this system. Code size is then estimated by adjusted function point.

5.1.1 Unadjusted Function Points

LarkDetect supports the following proposed functions:

User:

- Perform Cognitive Processing Speed Test
- Perform Executive Functioning Test
- View Report of personal statistics
- Register as a new user
- Login

Doctor:

- View individual patient details
- Register as a doctor
- Login

The measure of unadjusted function points is based on five primary component elements of these functions: Inputs, Outputs, Inquiries, Logical Files, and Interfaces. Each element ranges from Low Complexity, Medium Complexity to High Complexity. The detailed evaluation of the complexity is as follows:

Rating Inputs:

- Register as new user: (NRIC, password, first name, last name, date of birth, roles)
- Gathering Trail Making Test Result: (Time taken, error rate)
- Login Preferences (User or Doctor)

Files Type Referenced (FTR)	Data Elements		
	1-4	5-15	Greater than 15
Less than 2	Low (3)	Low (3)	Average (4)
2	Low (3)	Average (4)	High (6)
Greater than 2	Average (4)	High (6)	High (6)

Rating Outputs:

- Displaying report (Individual Information, results of test – Time and error rate)
- Displaying doctor view for keying in the patient NRIC to view report
- Display Random Generated Trail Making Test

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
Less than 2	Low (4)	Low (4)	Average (5)
2 or 3	Low (4)	Average (5)	High (7)
Greater than 3	Average (5)	High (7)	High (7)

Rating Inquiries:

- Data retrieval of Date of Birth to determine the session timeout duration of the test
- Login

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
Less than 2	Low (3)	Low (3)	Average (4)
2 or 3	Low (3)	Average (4)	High (6)
Greater than 3	Average (4)	High (6)	High (6)

Rating Logical Files:

- Registration
- Test Results

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (7)	Low(7)	Average (10)
2 to 5 RET	Low (7)	Average (10)	High (15)
6 or More RET	Average (10)	High (15)	High (15)

Rating Interfaces:

- FontAwesome
- Jquery

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (7)	Low(7)	Average (10)
2 to 5 RET	Low (7)	Average (10)	High (15)
6 or More RET	Average (10)	High (15)	High (15)

Summary of above analysis:

Element	Complexity	Detail
Inputs	Medium	Registration
	Low	Login
	High	Gathering Trail Making Test Result
Logical Files	Medium	Registration
	High	Test Results
Outputs	High	Display Report
	Low	Display Doctor's view
	High	Display Random Generated Trail Making Test
Inquiries	High	Data retrieval of Date of Birth
	Low	Login
Interfaces	Medium	FontAwesome
	High	Jquery

Calculation of Unadjusted Function Points:

Characteristic	**Low**		**Medium**		**High**	
Inputs	1	× 3	1	× 4	1	× 6
Outputs	1	× 4	0	× 5	1	× 7
Inquiries	1	× 3	0	× 4	1	× 6
Logical Files	0	× 7	1	× 10	1	× 15
Interfaces	0	× 5	1	× 7	1	× 10
Unadjusted FP	10		21		44	
Total=L+M+H	75					

5.1.2 Adjusted Function Points

Influence Factors	**Score**	**Detail**
Data Communications	5	Application is more than a front-end, and supports more than one type of teleprocessing communications protocol.
Distributed Functions	4	Distributed processing and data transfer are online and in both directions.
Performance	3	Response time or throughput is critical during all business hours. No special design for CPU utilization was required. Processing deadline requirements with interfacing systems are constraining.
Heavily used	2	Some security or timing considerations are included.
Transaction rate	3	Daily peak transaction period is anticipated.
On-line data entry	4	More than 30% of transactions are interactive data entry
End-user efficiency	2	Four to five of the efficiency designs are included
On-line data update	3	Online update of major internal logical files is included.
Complex processing	2	Any one of the complex components
Reusability	4	The application was specifically packaged and/or documented to ease re-use, and the application is customized by the user at source code level.
Installation Ease	1	No special considerations were stated by the user *but* special setup is required for installation.
Operational Ease	1	Effective start-up, back-up, and recovery processes were provided, but no operator intervention is required (count as two items).
Multiple sites	0	User requirements do not require considering the needs of more than one user/installation site.
Facilitate change	3	Flexible query and report facility is provided that can handle complex requests, for example, *and/or* logic combinations on one or more internal logical files (count as three items).
Total score	37	
Influence Multiplier = Total score × 0.01 + 0.65 = 37 × 0.01 + 0.65 = 1.02		
Adjusted FP = Unadjusted FP × Influence Multiplier = 75 × 1.02 = 76.5		

Scoring (0 – 5)
0 = No influence
1 = Insignificant influence
2 = Moderate influence
3 = Average influence
4 = Significant influence
5 = Strong influence

5.1.3 Lines of Code

According to Capers Jones statistics, each Function Point requires 53 lines of code if the application is implemented using Java.
Therefore, we have: **Lines of Code** = 76.5FP × 53 LOC/FP = **4054.4 LOC**

5.2 Efforts, Duration and Team Size Estimation

To estimate the effort and duration required for the project, we use function points as the basis to calculate Effort, Duration, Team size and finally the schedule. The estimates are expanded to account for project management and extra contingency time to obtain the total average effort estimates. From these averages, the duration of each work package in working days is estimated based on the following calculations.

- Working days include 5 days in a week.
- Effort = Size / Production Rate = (4054.4 LOC) / (31 LOC/PD)[1] = 130.78 PD
- Duration = 3 × (Effort) $^{1/3}$ = 3 × (130) $^{1/3}$ = 15.2 Days
- Initial schedule = 15.2 Days / 5 days a week = 3.04 Weeks
- Team size = 130.78 PD / 15.2 D = 8.604 P = 9 Persons
- Working hours include 8 hours in a working day.
- Total person-hours (PH) = 130.78 PD × 8 hours = 1046.24 PH

5.2.1 Distribution of Effort

1990's Industry Data	Work Package	Distribution	Estimates
Preliminary Design 18 %	Project Plan	9%	92.18
	Requirement Specification	9%	92.18
Detailed Design 25 %	User Interface	7%	71.70
	Technical Architecture	11%	112.67
	Data Modeling	7%	71.70
Code & Unit Testing 26 %	Code & Unit testing	21%	215.09
	Online Documentation	5%	51.21
Integration & Test 31 %	Integration & Quality Assurance	31%	317.51
	Extrapolated total effort		1024.24
	2% for project management		20.92
	3% for contingency		31.38
	Total effort		1076.54

These duration estimates are based on the assumption that each team member works an equal amount on any given work package. Numbers differs from PH due to rounding.

[1] Lines of code per Person Day statistics based on Industrial Benchmarks, 1997: 31 LOC/PD for United States; 62 LOC/PD for Canada

5.3 Cost Estimates

Hardware:

Developer workstations:

	Unit Price	Qty	Cost
Acer Swift 5 Laptop Intel Core i7 8th Gen 8GB RAM 1TB IDE drive	$2200	7	$15,400

Software:

Amazon Database, or Other Free License-based Software:

Amazon Database	$0.00

Software License Provided by Third Party:

Microsoft Office 365	$0.00

Other Resources:

Staff:

7 Employees with 1076.54 working hours with $22.70/hour	$24,437.46

Stationary:

Paper, photocopying and other miscellaneous cost	$50.00

Total:

$39,887.46

6 Product Checklist

The plan is that the items listed below will be delivered on the stated deadlines.

Project Deliverable	Estimated Deadline
Project Plan	Mar 12th, 2020
Requirements Specification	Mar 8th, 2020
Module/System Test Plan	Mar 14th, 2020
System Release (Demo)	Mar 17th, 2020

7 Best Practice Checklist

Practice
Document what we do; all documentation must be in a standardized format.
Pay attention to requirements, check for ambiguity, completeness, accuracy, and consistency. The requirement documentation must contain a complete functional specification.
Keep it simple. Complexity management is one of the major challenges. Strive to: • Minimize interfaces between modules, procedures and data. • Minimize interfaces between people, otherwise exponential communication cost • Avoid fancy product functions, design as long as the functionality meets the customer requirements
Require Visibility. We must see what we build otherwise we can measure the progress and take management action. This includes: the manager must have good communication with his or her employees; require developers to make code available for review; review design for appropriateness.
Plan for continuous change. We must: • All manuals designs, test, source code should have revision numbers and dates revision history comments, change marks to indicate the changes • New revisions should be approved before being made and checked for quality and compliance after being made • Use a configuration management system and make processes • Required maintenance
Don't underestimate. We must be careful to obtain accurate estimates for: time, effort, overhead, meeting time, and especially effort on integration, testing, documentation and maintenance.
Code reviews are a much more efficient method to find software defects. Plan and manage code reviews between team members
Software testing will use both black box and white box testing. It will involve unit, functional, integrating and acceptance testing.

8 Risk Management

Besides the general risk management, the following risks have been identified for the LarkDetect project:

More changes to requirements than anticipated
Impact Severity: High
Probability: 25%
Impacts: Depending on the stage at which changes occur, could range from needing to update the requirements documentation to a needing to do a complete redesign.
Risk Reduction: Be rigorous in eliciting requirements. Ensure that requirements are understood correctly by the development team and make customer aware of potential repercussions of requirement changes. If it's unchangeable or difficult to change, the development team will provide the cost for changes for the client to consider.

Specification Delays
Impact Severity: High
Probability: 15%
Impacts: Delay in finalizing the specification will push the schedule for all following stages of the project. It may also cause the actual time and cost to complete the project to overshoot from what was projected.
Risk Reduction: Monitor progress of specification carefully. Inquire customer to provide any inadequate information in a timely manner and potential consequences should they fail to do so.

System size underestimated
Impact Severity: Moderate
Probability: 30%
Impacts: More work will need to be spent on design and coding; could negatively impact schedule.
Risk Reduction: Update estimates often as project progresses. Management to take note of any potential delays in completion.

Staff Turnover (staff leaving before project complete)
Impact Severity: Medium
Probability: 5%
Impacts: There would be more work for remaining employees, and any specialized skills or knowledge would be lost.
Risk Reduction: Offer benefits and incentives to staff. More than one member to be assigned to each work assignment. Ensure all members to work closely with each other such that everyone would be familiar with the development work. Therefore, work can be redistributed/each member could take over the work easily.

Problems coordinating within group
Impact Severity: Moderate
Probability: 40%
Impacts: Members may be unaware of what is expected of them; managers may not be able to measure progress; portions of projects not completed.
Risk Reduction: Follow communication plans as documented in section 2.3

Customer cancels project
Impact Severity: High
Probability: 1%
Impacts: All work will have been wasted.
Risk Reduction: Keep in close contact with customer. Ensure that they have some market research indicating a demand for this product. Prior to development, ensure that both parties are bound by an agreement with liabilities should unexpected risks such as this occur.

9 Quality Assurance

The project will achieve the quality assurance by following the standard set by the company. The specific procedures and details shall be provided in the Quality Plan.

Specific test procedures and details shall be provided in the Module/System Test Plan.

In addition, LarkDetect shall make use of two testing methodologies:

- **Unit Testing** involves testing system components individually.
- **In-Place Testing** involves testing of the whole system as a unit.

Furthermore, these methodologies will be used to test two important aspects of the LarkDetect:

- **System Function** will be tested to ensure that software flaws are eliminated, and
- **Algorithmic Function** will be tested to ensure that heuristic aspects of the project (such as user performance report) perform realistically to provide value to the users.

CoronaSG's methodology makes broad use of realistic test cases. Detailed test data is an important part of the final project delivery. Although CoronaSG's client is expected to furnish and enter data regarding report generation in LarkDetect, CoronaSG shall provide a comprehensive and detailed subset of this data for testing purposes. CoronaSG will validate code and heuristic result ranking technology using realistic scenarios.

10 Monitoring & Control

The development of LarkDetect will be monitored closely to ensure all milestones are met according to the project schedule. Many procedures are required in order to be able to successfully monitor the progress of a software project. The following procedures are considered the most important:

Quantitative measurement of resource consumption: Estimates of LarkDetect's resource requirements, primarily in terms of human resources, can provide a quantitative measurement of project progress when compared to progress in terms of project milestones. The percentage estimates of each milestone's resource requirements provided in this document allow for easy progress tracking.

Identification of major project risks: Early identification of major risks to the project allows for placement of preventative measures before problems can develop. Major risks have been identified in the Risk Management section of this document, along with the measures being taken to avoid them.

Regular reviews of project progress: Throughout the duration of the LarkDetect project, CoronaSG shall meet weekly to review the progress of all project tasks, including management, planning, analysis, development, and testing.

Timeline Planning and task decomposition: This document outlines an estimated timeline for the project. A reasonably accurate timeline can be assembled by hierarchically decomposing tasks into measurable subcomponents and estimating requirements for each. At the same time, this decomposition can assist in task assignment and balancing. Throughout the implementation phase, these subcomponents can allow for fine-grained measurement of progress. Project subcomponents and timeline estimates are included in the Estimates and Work Breakdown Structure sections of this document.