SOFTWARE DEVELOPMENT PLAN
(Assignment 5)

Submitted by
Rutuja Mandhare

in partial fulfillment of the requirements for
CSE7315 -- Planning and Managing a Software Project

Spring, 2015

LOCATION: **("In class")**
DAYTIME PHONE: **(646) 361-2909**
E-MAIL: **rmandhare@smu.edu**

Grading area:

Expected Features of SDP			
1. Product Information (10 points)			
Unknowns and Assumptions; Product description and constraints, including customer concerns, key technical features, and architecture (5)		Product Risks and Measures / Computer Resource Capacity and Performance Indicators (5)	
2. Project Information (20 points)			
Project Summary, Integrated Master Schedule (IMS) (2 of them), Software Schedule (5)		WBS for Project (Appendix A) (5)	
Project Organization / Software Organization / Skills Required (5)		Project Risks, Management Measures and Indicators (5)	
3. Process Information (30 points)			
SW Development Process, Artifacts, Rationale & Tailoring (10)		Reviews (technical, management) (5)	
Tools / Development Environment (5)		Testing & Evaluations; Action Item Procedures (5)	
		Process Risks, Measures and Quality Indicators (5)	
Appendices D and E: Risk and Measurement Plans (20 points)			
Risk Analysis & Management Plan(10)		Measurement Plan (10)	
Appendices F and G: Support Process Information (10 points)			
SW Configuration Management / Development Library (5)		SW Quality Assurance / Corrective Action Process / Audits (5)	
Personal WBS and Earned Value Workbook (10)			
Personal WBS (5)		Earned Value Workbook (5)	

Total Grade: _____

Software Development Plan for Mr. Television

Assumptions and Unknowns

No.	Unknown	Assumption / Resolution	Related Risk(s)
1	Does the device has speaker inside	We are assuming it has speaker	If yes then it will cause the performance.
2	Minor functions performed by the 'Miscellaneous stuff'' in processor#1	Minor functions performed by the 'Miscellaneous stuff' in processor#1	Would lead to product risk
3	Will the memory in the processors will be sufficient for the software to be loaded. Need to check during size estimation	Assumed that processors will have enough memory will be able to accommodate the software	There can be a risk that the memory is less in any of the processors thereby leading to a situation where the software will not be deployed properly.
4	How sure are we that the recent version of the operating system will be delivered in the proposed date	Assumption that there will be recent version of operating system soon.	Risk that the recent version may not be delivered on time.
5	Are FANCYMIKES will be able to deliver their software.	Assumption that FANCYMIKES are not in financial trouble.	There is a risk when FANCYMIKES file for bankruptcy. In that case, the sound functions will not be delivered and changes cannot be accommodated

6.	Will there be enough qualified staff to build this software	There will be enough qualified staff.	If not the project might delay.
7.	The image quality be acceptable to customers?	Marketing research must be performed before launch the product.	Image on output device may not be of good quality which might be not acceptable by customers.
8	Will device require wired connection?	It might require Wi-Fi connection only.	If it required wired connection then requirements might change and which cause the schedule and delay the project.

1.0 Product Summary

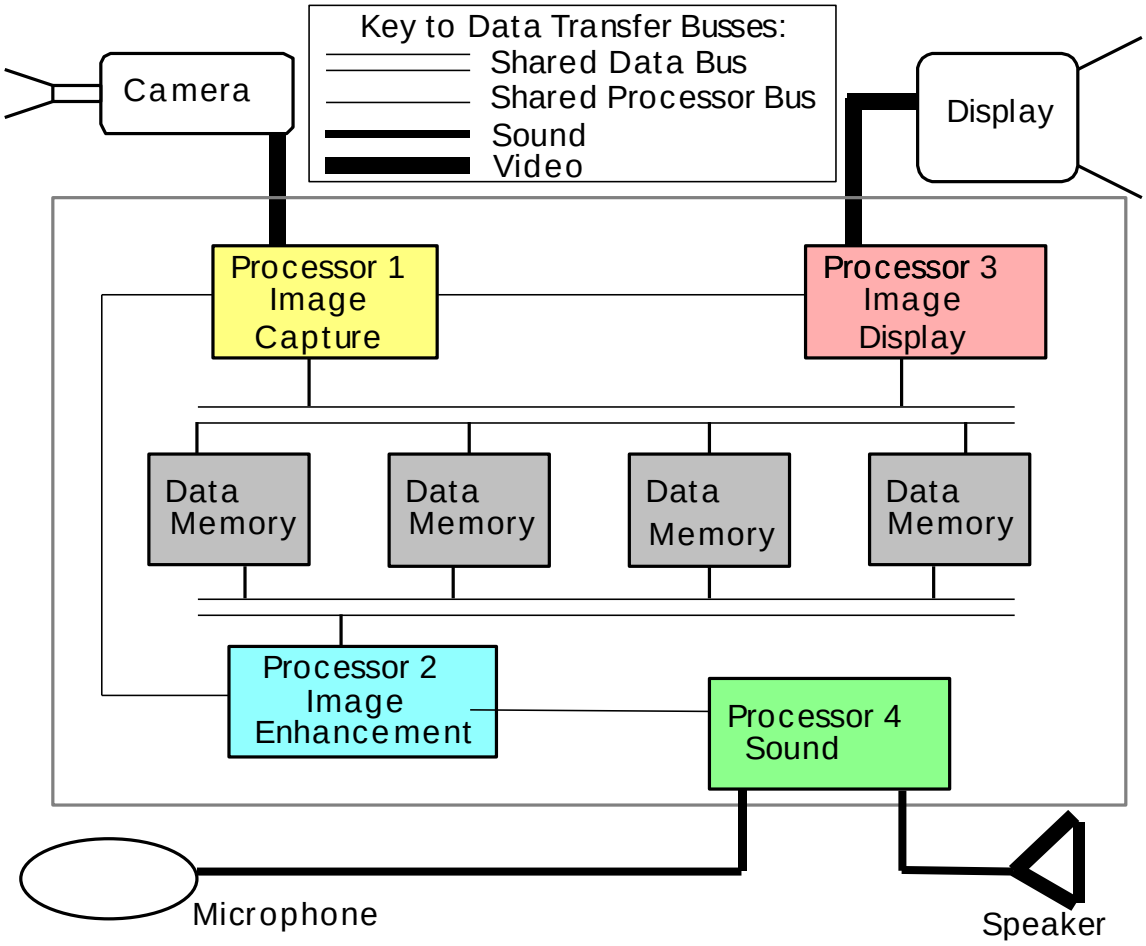
"Mr. Television", an enhancement product of "Mr. Microphone Plus" transmits both enhanced voice using patent technologies and video by improving the image. Mr. Television" looks a lot like a cellular telephone with a built-in camera, except the camera faces toward the performer. It is held in the hand of the performer with the camera and small display screen facing the performer. It may also be mounted on a stand and connected through a cable or Bluetooth to an external video camera, for more professional applications. The wireless Bluetooth connection built into the device carries the sound and video to any Bluetooth-enabled digital television system and/or amplifier. Typically, it is connected to a large screen television or video projector.

1.1 Product Description

1.1.1 Background

"Mr. Microphone Plus" is a popular product which enhances the voice of the singer. The customers who bought this product felt the need of a video (actual singer) along with audio. They would like to see the actual singer being displayed on the television/projector screen. Therefore the idea of "Mr. Television" was put forward, which consist of camera (pointing at the performer), video processing electronics, a small display (also facing the performer), and a wireless Bluetooth transmitter. The new product will display the performer's image on one or more digital television set(s), which typically face the audience. The reason for "Mr. Television" project is to improve the business to a bigger market. The customers who buy the product wanted an enhanced feature of the singer to be seen in a television/projector. Engineering is to supply the complete "Mr. Television" system, a product resembling cellular telephone in a single plastic case, and consisting of internal electronics and hardware and software components

1.1.2 Hardware and Software Architecture



Processor 1	Processor 2	Processor 3	Processor 4	Operating System
Frame Grabber	Expert system	Frame putter	Sound Functions	
Bit Twiddler	Bit Twiddler	Bit Twiddler		
Miscellaneous Stuff		Miscellaneous Stuff		

The 4 processors mentioned in the diagram are contained within a single “cell” processor, represented by the large grey box in the diagram. This device also contains 8 memory banks.

Processor 1: It captures the images and copies a frame of data from camera to graphics memory.
Components of processor 1:

Frame Grabber: frame grabber is copying a frame of data from the camera to one of the data memories and also provides the master timing signals.

Bit Twiddler: Performs minor data correction functions on the data from the camera.
Miscellaneous stuff: It performs various minor housekeeping functions. Its main function is to provide the master timing signal to the other processors.

Processor 2: It has the following components:

Expert System for Image Enhancement – Improves the image. This is the major function of the "Mr. Television" product.
Bit Twiddler: Performs minor data correction functions on the data from the camera.

Processor 3: contains the Frame Putter, which is similar to the Frame Grabber in Software Item #1. It also needs an almost identical bit twiddler, but a somewhat different version of the miscellaneous stuff.

Processor 4: It is the Sound application software and it processes sound, fully synchronizing it with the video output of processor 3. Other than this synchronization, it is fully independent of the other processors.

1.2 Key Technical Features

Hardware
The four processors mentioned in the diagram are contained within a single "cell" processor device. This device also contains eight memory banks, as follows:

- 4 internal memory banks, 1 for each of the four processors (128 MB each)
- 4 "external" shared data memory banks (1 Gigabyte each). Each of these has a separate data path (so they do not interfere with each other).

Each processor runs the Power PC instruction set and is capable of executing at 4 Gigaflops. The four processors interconnect via a shared bus that allows each to signal the others. In addition, there is a shared data bus (parallel lines in the diagram 1) connecting the four 1 Gigabyte data memories. Each of these data memories is capable of holding one copy of the video image. The processors are capable of reading and writing data directly to and from any of these data memories, thus they do not need to copy the graphics data into their internal program memories.

VCR or DVD recorder. Recorded video will be displayed back on the small display
screen, recorded video and audio will be transmitted over the wireless internet outputs.
- Lightweight video camera find on a cell phone to record the performance.

Recorded data will be displayed back to a display screen and sent out over the wireless internet output.
- Internal processors of:
- Camera to digital image translation processor
- Image analysis and refinement processor
- Image display processor

- Sound reception, refinement, and amplification processor

The camera in the front captures the raw image and changes it to digital image which is sent to the graphics data memory by Processor 1(Image Capture) by means of shared bus. The application software in processor 1 receives and transfers the data to one of the four data memories. The processor 2 refines the image from one data memory to another. The processor 3 takes the improved image from the data memory and is responsible for display. The processor 4 is responsible for the sound and works independently except for the fact that it should transmit sound in synchronization with the video

Software

All software is to be written in Python, Java, or C.

Each of the four functions is to be developed as an independent software item, running on a separate processor within the cell processor chip.

The software is to be developed using a professional quality development process.

Software Configuration Management

Software Quality Assurance

The software will be divided into four software items, corresponding to the four processors. Each software item contains two software elements within its internal memory: an operating system and an application program.

1.3 Product Risks

Product Risk Summary				
Risk Id	Risk	Evidence	Kind of Risk	Description
1	Risk that there is not enough memory so system will not be able to work properly.	06 mrtvmemo(1)- it looks like we will not be able to increase the size of any of the memories used in the system. Software size is "very Big	Technical	Software size in the processors is very big and memories cannot be increased so due to this system won't work properly.
2	Risk that there is no recent version of operating system.	06 mrtvmemo(1)- We have not yet obtained the most recent version of the operating system.	Technical	The recent version has not been obtained. So, the processors may not be compatible with the current OS. Also, there is no 100% guarantee when the latest version will be delivered.
3	Risk that assumptions of	06 mrtvmemo(1)- As you know, you have a	Technical	Risk that the developers will

Product Risk Summary				
Risk Id	Risk	Evidence	Kind of Risk	Description
	requirements of sound enhancement and image enhancement are same.	requirements document for the Mr. Television system that says nothing at all about requirements for image or sound enhancement.		implement the image enhancement using the requirements of sound enhancement may cause the image enhancement in the product to fail.
4	Risk that the prototype is defective, leading to issues in product.	05mrtvschedule (1)– Prototyping - "Although neither is working yet, the engineers (all from the R&D lab) are confident that it will work soon"	Technical	The engineers responsible for developing actual products are somewhat skeptical that the prototype of expert system and special hardware (developed by R&D lab) will work soon, having tried to make real products out of R&D prototypes in the past.

1.4 Technical and Performance Measures (Metrics)

Product Measures (Metrics) Summary		
Measure	What will be Measured	Corresponding Risk
Processor memory	The size estimation of processors are to be collected. The estimates are to be checked with processor size.	1
Operating System	Need to make sure that the updated version of OS is compatible with coding	2
Image Enhancements	The sound enhancement requirements that will be applicable for image enhancement will be collected. Other image enhancements algorithms can be compared and implemented	3
Prototype	Need to make sure the product work as need.	4

2.0 Project Description

2.1 Project Overview

The “Mr. Television” project is a multi-year project for the prototyping, development, production and support of a next generation Karaoke-like product. The Product Development phase is about to begin after R&D and Prototyping. The "Mr. Television" project has an overall lifecycle that consists of the several phases. The R& D phase took three years and completed a year ago and prototyping is about to complete.

Deliverables for Development phase consist of the following items:

i. All Software and complete design specifications (plus one working version) of the Hardware identified in the Software Requirements Specification.
ii. All Documentation identified in the Software Requirements Specification. All documentation is to be produced in accordance with the company's standard format.

Preliminary Schedule Planning for Product Development Phase

The software staff members have collected information from each of six organizations. Each has identified their key tasks and dependencies. These six are:
Program Management
System Engineering
System Integration
Mechanical Engineering
Electrical Engineering
Software Engineering

2.2 Project Lifecycle

The "Mr. Television" project has an overall lifecycle that consists of the following phases:

- **R&D:** This phase took three years and completed a year ago.

- **Prototyping** - This phase began last year and is nearly complete.

Product Development - This is the phase that is about to begin and for which we develop the software development plan. The intent is to develop real hardware and software that can be manufactured. Management would like this phase to be done in about a year but we know from past history that it is more likely to take two years.

Pre-Production - This phase will begin as soon as product development is complete. It will last about 6 months and its purpose will be to develop manufacturing capability and, in parallel, do beta tests of the product with potential customers to see if the product will sell. There will be some software support needed during this phase, but you are not expected to estimate that yet.

Production - This phase will begin as soon as pre-production is complete. Its duration will depend on how well the product sells.

2.3 Integrated Master Schedule (IMS)

It is a hierarchy of program events, in which each event is supported by specific accomplishment and each accomplishment is based on satisfying specific criteria to be considered to complete.

2.3.1 Initial IMS

ID	Task Name	Duration(months)	Resource Names	Key Dependencies
1	Mr. Television Project			
1.1	Requirements	25		
1.1.1	System requirements	8.5	System Engineering Lead	
1.1.2	Mechanical requirements		Mechanical Engineering lead	
1.1.3	Electrical requirements		Electrical Engineering lead	
1.1.4	Software requirements		Software Engineering lead	
1.1.4.1	Collect software requirements for Processor#1	2		
1.1.4.2	Collect software requirements for Processor#2	1		
1.1.4.3	Collect software requirements for Processor#3	1		
1.1.4.4	Collect software requirements for Processor#4	1		
1.1.4.5	Develop test plan	1		Software requirements
1.1.4.6	Generate other artifacts during requirements analysis	1		
1.1.4.7	Review requirement phase	5 days		
1.2	Design			
1.2.1	Top level design			Requirements
1.2.1.1	Top level design processor#1	2		
1.2.1.2	Top level design processor#2			

1.2.1.3	Top level design processor#3	2		
1.2.1.4	Top level design processor#4			
1.2.2	Detailed design	1		Requirements
1.2.2.1	Detailed design processor#1	1		
1.2.2.2	Detailed design processor#2	2		
1.2.2.3	Detailed design processor#3	1		
1.2.2.4	Detailed design processor#4	2		
1.2.3	Design review	1		Top level design
1.2.3.1	Design review for processor #1	2		
1.2.3.2	Design review for processor #2	2		
1.2.3.3	Design review for processor #3	2		
1.2.3.4	Design review for processor #4	1		
1.2.4	Produce artifacts for design phase	1		
1.2.5	Review design phase	4		
1.3	Coding		Software eng. lead	
1.3.1	Analyze requirements and design artifacts	3		
1.3.2	Discuss with team members	2		
1.3.3	Develop Code	2	Design	
1.3.3.1	Develop code for Processor#1	2		
1.3.3.2	Develop code for Processor#2	2		
1.3.3.3	Develop code for Processor#3	2		
1.3.3.4	Develop code for Processor#4	2		
1.3.4	Unit Testing		Requirements	Developers
1.3.4.1	Unit Testing for Processor#1	1		
1.3.4.2	Unit Testing for Processor#2	1		
1.3.4.3	Unit Testing for Processor#3	1		
1.3.4.4	Unit Testing for Processor#4	1		
1.3.5	Integrate Components	3		
1.3.6	Review Coding Phase	2		
1.4	Testing			
1.4.1	Test Planning	1		
1.4.2	Test case development	2		
1.4.3	Test Environment	2		

1.4.4	Test Case execution	1		Testers
1.4.5	Testing Phase Review	1		
1.5	Deployment	1		
1.6	**Configuration Management**	1		

2.3.2 Final IMS

ID	Task Name	Duration(months)	Resource Names	Key Dependencies
1	Mr. Television Project			
1.1	Requirements	25		
1.1.1	System requirements	8.5	System Engineering Lead	
1.1.2	Mechanical requirements		Mechanical Engineering lead	
1.1.3	Electrical requirements		Electrical Engineering lead	
1.1.4	Software requirements		Software Engineering lead	
1.1.4.1	Collect software requirements for Processor#1	4		
1.1.4.2	Collect software requirements for Processor#2	1		
1.1.4.3	Collect software requirements for Processor#3	1		
1.1.4.4	Collect software requirements for Processor#4	1		
1.1.4.5	Monthly Review	1 day	Project Manager	
1.1.4.6	Develop test plan	1		Software requirements

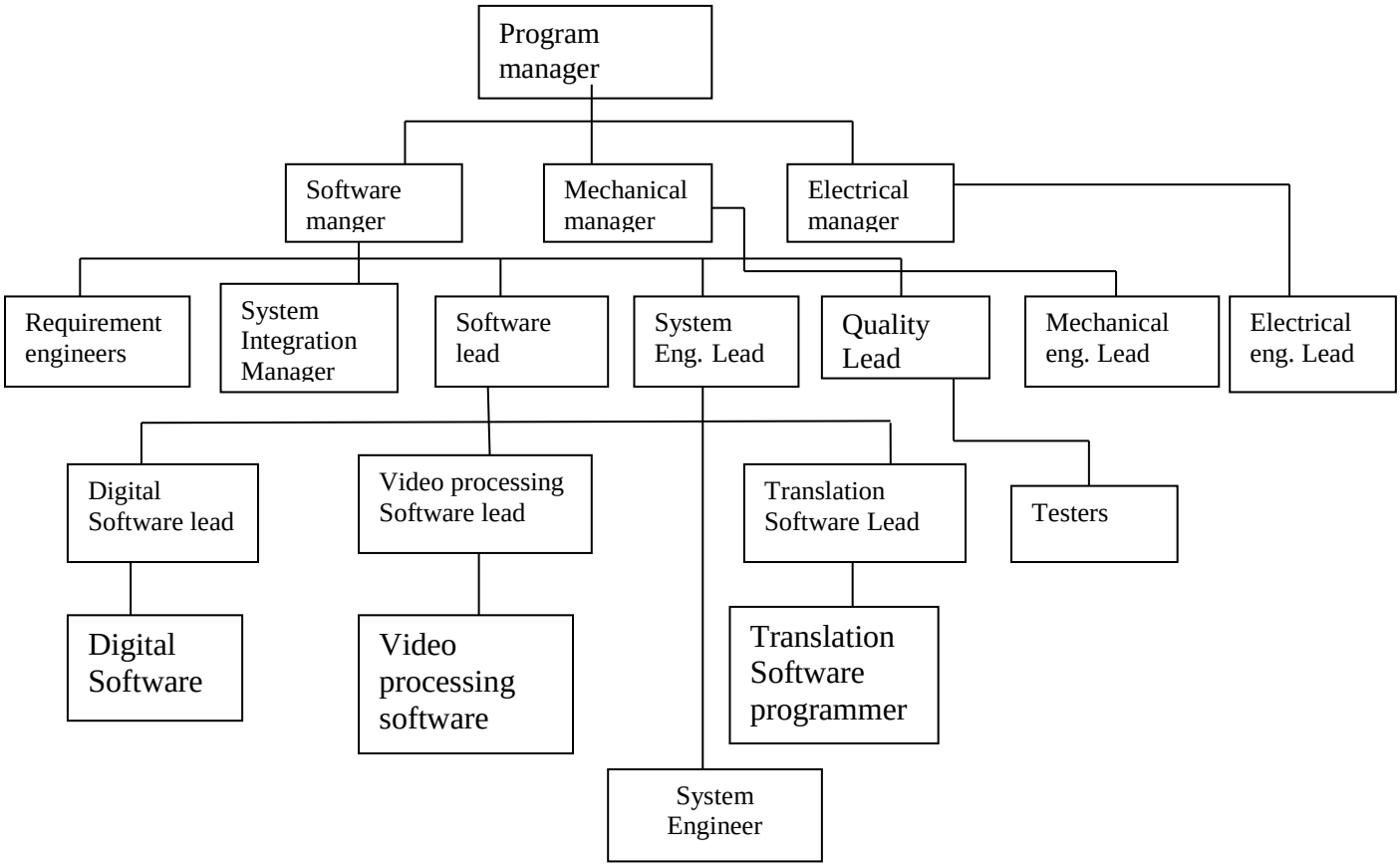
1.1.4.6	Generate other artifacts during requirements analysis	1		
1.1.4.7	Review requirement phase	5 days		
1.2	Design			
1.2.1	Top level design	3		Requirements
1.2.1.1	Top level design processor#1	1		
1.2.1.2	Top level design processor#2	1		
1.2.1.3	Top level design processor#3	1		
1.2.1.4	Top level design processor#4	1		
1.2.1.5	Monthly Review	1 day	Project Manager	
1.2.2	Detailed design	3		Requirements
1.2.2.1	Detailed design processor#1	1		
1.2.2.2	Detailed design processor#2	1		
1.2.2.3	Detailed design processor#3	1		
1.2.2.4	Monthly Review	1 day	Project manager and Design lead	
1.2.2.5	Detailed design processor#4	1		
1.2.3	Design review	1 day		Top level design
1.2.3.1	Design review for processor #1	1 day		
1.2.3.2	Design review for processor #2	1 day		
1.2.3.3	Design review for processor #3	1 day		
1.2.3.4	Design review for processor #4	1 day		
1.2.4	Produce artifacts for design phase	1 day		
1.2.5	Review design phase	2 days		
1.3	Coding	3	Software Engineering lead	
1.3.1	Analyze requirements and design artifacts	3 days		
1.3.2	Discuss with team members	2 days		
1.3.3	Develop Code	2		Design
1.3.3.1	Develop code for Processor#1	2		

1.3.3.2	Develop code for Processor#2	2		
1.3.3.3	Monthly review	1 day	Dev lead	
1.3.3.4	Develop code for Processor#3	2		
1.3.3.5	Develop code for Processor#4	2		
1.3.4	Unit Testing	4 days	Developers	Requirements
1.3.4.1	Unit Testing for Processor#1	1		
1.3.4.2	Unit Testing for Processor#2	1		
1.3.4.3	Unit Testing for Processor#3	1		
1.3.4.4	Unit Testing for Processor#4	1		
1.3.5	Review Coding Phase	2 days		
1.4	Integrate Components	3		
1.4.1	Monthly review	1 day	Project manager	
1.5	**Configuration Management**	2		
1.5.1	Monthly Review	1 day	Project manager	
1.6	**Quality Assurance**	2		
1.6.1	Test Planning	1		
1.6.2	Test case development	2		
1.6.3	Audits	1 day	Project manager	
1.6.4	Test Environment	2		
1.6.5	Monthly review	1 day	Project manager	
1.6.6	Test Case execution	1 1		Testers
1.6.7	Testing Phase Review	1		
1.7	Deployment	1		

2.3.3 How the IMS has Changed and Why

The order of few phases have been re-ordered as per the Effort estimation. i.e. Coding follows Integration, Configuration Management and QA. Also, Integration have been made as a separate phase in the final IMS. The duration for each of the phase have been changed as per the effort estimation. Since, the waterfall model is tailored to overlap little bit, the IMS is changed as per the tailored process. Monthly reviews have been added for every month in all the phases. Audit have been added in QA.

2.4 Project Organization Structure



2.5 Software Position and Roles

Role	Job Classification or job grade level	Responsibilities of This Role	Skills Required for This Role	Approximate Number Needed on Project
Program Manager	Senior 10+ years experience	Program manager monitors project progress and performance, manages communication with stakeholders and manages project output in line with the program plan. He also looks after the overall management and coordination of the program.	Advanced skills in financial planning and administration and additionally contract arrangements over the program. Capacity to innovatively characterize the project pacing and get incremental profits before the project is finished. Knowledge in establishing and executing the administrative system for both constituent program and project itself.	1
Software manager	Senior 7+years experience	Software manager keeps the development team on track and reports the progress to the Program Manager. He manages all project planning and prioritization. He monitors individual employee's performance.	Strong technical and analytical skills along with expert knowledge of computer programming languages and platforms. Strong leadership and managerial abilities to train and evaluate performances of the staff. Excellent communication skills to collaborate with various levels of management.	4
		Software Lead reports the	Strong technical and	

Role	Job Classification or job grade level	Responsibilities of This Role	Skills Required for This Role	Approximate Number Needed on Project
Software Lead	Senior 5+ years of experience.	progress of all the software teams to the Software Manager. He is responsible for coordinates development, unit and integration testing of all the software components. He goes about as a state of contact for all the issues identified with Translation, Digital and Video Processing.	analytical skills for software development of television products. Strong leadership and managerial abilities. Expert knowledge of computer programming languages like Java, C, C++ and Fortran. Expert knowledge in software design principles for Digital, Translation and Video Processing.	5
Digital Software Lead	Senior 5 years of experience.	Digital Software Lead reports the progress of the Digital Software Programmers to the Software Lead. He is responsible for coordinates development, unit and integration testing for all the members developing Digital Processing Software. He goes about as a state of contact for all the issues identified with Digital Processing.	Strong technical and analytical skills for digital software development of television products. Strong leadership and managerial abilities. Expert knowledge of computer programming languages like C, C++ and Java. Ability to translate from C to Java. Expert knowledge in software design principles for Digital Processing.	3
Video Processing Software Lead	Senior 5 years of experience.	Video Processing Software Lead reports the progress of Video Processing Software Programmers to the Software Lead. He is responsible for coordinates development, unit and integration testing for all the members developing Video Processing software.	Strong technical and analytical skills for Video Processing software development of television products. Strong leadership and managerial abilities. Expert knowledge in computer	

Role	Job Classification or job grade level	Responsibilities of This Role	Skills Required for This Role	Approximate Number Needed on Project
		He goes about as a state of contact for all the issues identified with Video Processing.	programming languages like C, C++ and Fortran. Expert knowledge in design principles for Video	3
Translation Software Lead	Senior 5 years of experience.	Translation Software Lead reports the progress of Translation Software Programmers to the Software Lead. He is responsible for coordinates development, unit and integration testing for all the members developing Translation Software. He goes about as a state of contact for all the issues identified with the Translation Software.	Processing. Strong technical and analytical skills for Translation Software development of television products. Strong leadership and managerial abilities. Strong knowledge of computer programming languages like C, Fortran and Java. Expert knowledge in design principles for Translation Software.	3
Digital Software Programmer	Junior 1-4 years of experience.	Digital Software Programmers report their progress to the Digital Software Lead. They are responsible for the development, unit and integration testing Digital Processing software components. They also report the issues with the Digital Software to the Digital Software Lead.	Strong technical skills for Digital Software development of television products. Prior experience with companies working on similar projects. Strong knowledge of design principles for Digital Processing. Strong knowledge of either C++ or Java.	5 5
Video Processing Software Programmer	Junior 1-4 years of experience.	Video Processing programmers report their progress to the Video Processing Software Lead. They are responsible for development, unit and integration testing of Video	Strong technical skills for Video Processing software development of television products. Prior experience with companies working on similar projects.	5

Role	Job Classification or job grade level	Responsibilities of This Role	Skills Required for This Role	Approximate Number Needed on Project
		Processing software components. They also report the issues with the Video Processing software to the Video Processing Software Lead.	Strong knowledge of design principles for Video Processing. Proficient in either C or Fortran.	
Translation Software Programmer	Junior 1-4 years of experience.	Translation Software programmers report their progress to the Translation Software Lead. They are responsible for the development, unit and integration testing of the Translation Software components. They also report the issues with the Translation Software to the Translation Software Lead.	Strong technical skills for Translation Software development of television products. Prior experience with companies working on similar projects. Strong knowledge of design principles for Translation Software. Proficient in either Java of Fortran.	5
System Engineer	Junior 1-4 years of experience.	Complete configuration, installation and support of equipment in a Microsoft Windows environment Troubleshoot and resolve computer and telephony related issues when contacted by clients by providing both on-site and remote support Maintaining software applications, operating systems and regular maintenance. Managing assigned projects and program components to deliver services in accordance with established objectives.	Technical, analytical, interpersonal and organization skills required. Associates Degree. Ability to communicate in an understandable, polite and friendly manner, both written and verbal.	

Role	Job Classification or job grade level	Responsibilities of This Role	Skills Required for This Role	Approximate Number Needed on Project

2.6 Project Risks

Project Risk Summary				
Risk Id	Risk	Evidence	Kind of Risk	Description
1	Risk that there might be schedule delays because the processors may need more modifications than planned.	mrtvswdesignconcept(1)-Operating system-This software, although "tried and true", will probably require more modification than the systems engineers believe.	Schedule and cost	The software may require more modification than calculated by system engineers. As there will be more modification there will be more work and rework which will directly affect schedule and cost.
2	Risk that "The schedule may be delayed because the image enhancement requirements may not be available for several months"	06 mrtvmemo(1)- The image enhancement requirements are not likely to be available for several months at the earliest.	Schedule	There are no image enhancement requirements for few months after the start of the project. So it takes more time to incorporate the image enhancement requirements after the requirement phase is completed.
3	Risk that the prototype is defective, leading to schedule delays	05mrtvschedule – Prototyping. The engineers responsible for developing actual products are somewhat skeptical of this, having tried to make real products out of R&D prototypes in the past	Schedule	The engineers responsible for developing actual products are somewhat skeptical that the prototype of expert system and special hardware (developed by R&D lab) will work soon, having tried to make real products out of R&D prototypes in the past.
4	Risk that design, development and testing impacts of safety standards will not be understood by the project team early enough, leading to rework	00 mrtvsow -Applicable Standards and Documents	Cost and Schedule	The engineering staff should make sure that all applicable government safety standards are followed with this product.

2.7 Key Project Measures (Metrics)

Project Measures (Metrics) Summary		
Measure	What will be Measured	Corresponding Risk
OS modification	The OS modifications will be monitored. This has to be figured out while estimating size.	1
Defective prototype	Measure the prototype with the requirements.	3
Image Enhancements	The sound enhancement will be used for image enhancement.	2
Safety Standards uptake	Requirements for Safety Standards Requirements Traceability Matrix – trace requirements through design, code, and test scripts Tests to verify adherence to safety standards	4

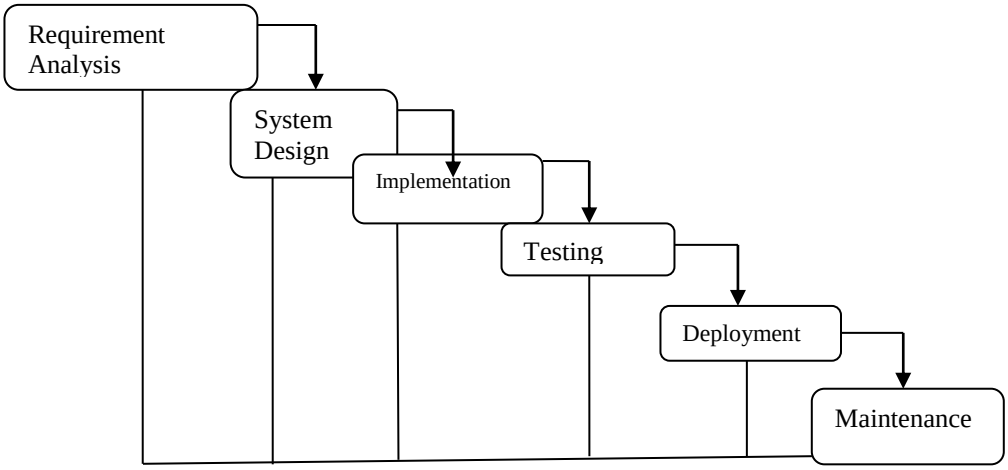
3.0 Software Development Process

3.1 Software Lifecycle Description

The waterfall model is a model which was developed for software development; that is to create software. It is called as such because the model develops systematically from one phase to other in a downward fashion, like a waterfall. It is the first process model to be introduced and it is also called as linear-sequential life cycle model. It is a phased progression of activities i.e. each phase begins after the previous phase is over.

Phases of waterfall model:

- Definition Study/Analysis
- Basic Design
- Technical Design/Detailed Design
- Construction
- Testing
- Integration
- Management and
- Maintenance.



Requirement Analysis: The first phase involves understanding what you need to design and what is its function, purpose etc. The specifications of the input and output or the final product are studied and marked.

System Design: The requirements gathered are evaluated and a proper implementation strategy is formulated acc. to software environment. It is further divided into Top-level design and detailed design.

Implementation: The actual coding of the software is carried out at this stage. The flowcharts / algorithms are converted into instructions written in a programming language.

Testing: All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

Deployment: After the testing is done, the product is deployed in the customer environment or released into the market.

Maintenance - It is an ongoing process. The additional bugs will be fixed and business scenarios will be added.

3.2 Rationale for Lifecycle(s)

Waterfall model is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of model is basically used for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. It works well for a project like "Mr. Television". Also, the technology used is stable and not dynamic. There is no requirement of any initial version to be released early and so waterfall model can be suitable.

3.3 Description of Software Development Process

Process Tasks		
Phase	Task	Description
Requirements	Identify Software Requirements	Requirements must be determined and agreed to by the customers, users, and suppliers of a software product before the software can be built.
	Analyze Requirements	Requirement analysis encompasses that task that determines the needs or conditions to meet for new or altered product. It is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed.
	Test Plan	A test plan specifies what you want to test and how to run those tests. A test plan can be applied to a specific iteration of our project.
	Software Requirements Review	After the test plan is completed, the software requirements phase is reviewed and if passed, design phase is started.
Design	Create Top Level Design	It identifies various components and how components interact with each other as well as a more detailed description of each of those components that allow them to be constructed.
	Develop Detailed Design	After high-level design, a designer's focus shifts to low-level design .Each module's responsibilities should be specified as precisely as possible. Constraints on the use of its interface should be specified
	Software Design Review	After Done with all design part review the progress and check whether it is ready for coding.

Process Tasks		
Phase	Task	Description
Coding	Write code	Develop the code based on the requirements and design.
	Unit testing	Once the code is developed, the developers test the individual units of source code are tested to ensure individual modules work fine.
	Integrate Components	The different software units are integrated to form a product which would be sent for testing.
	Coding Phase review	This is the final stage of coding phase. Once the components are integrated, the final product is obtained. The managers and leads discuss to review if all the specification are met and whether the product can be approved to test.
Test	Test Planning	This is similar to what happens in requirements phase. Planning activities like what to test, resources, test environment, schedules are planned.
	Test case development	The typical different user actions that would test a feature or an aspect of a feature would be documented with the expected and actual results.
	Test Environment setup	We simulate the test environment with the goal of replicating the end users requirements.
	Test Execution	Here, we execute the test cases to see whether the test cases are pass or not

List the artifacts that will be created by your software development process.

Table of Artifacts				
Name of Artifact	Generated By (Process step)	Contents/Use	Format	Used by (Process Step)
Requirement Specification	Requirement analysis	Description of software system. Used to gather information about how the system is expected to perform.	MS Word document	Software requirements, Design, Coding, Testing
Test Planning	Requirement analysis	Test design, Test Cases. Used in testing phase.	MS word document, MS Excel	Testing
		Overview of an entire		

Table of Artifacts				
Name of Artifact	Generated By (Process step)	Contents/Use	Format	Used by (Process Step)
High level Design document	Design	system. Used to understand the data/work flow between component systems.	MS word document	Coding, Testing,
Low level Design document	Design	Contains designs of individual modules identified during high level design (Detailed functional logic). It can be represented using diagrams like class diagrams, UML diagrams. Useful while coding to know the logics	MS word documents, UML diagrams	Coding, Testing
Coding	Coding	Code (python, java, C). To make the functions work.	Source code	Code
Testing	Testing	Defects, Test case results are produced in this phase. This will be used to fix the defects in the software	MS Excel	Coding, Testing

3.4 Rationale for Software Development Process

Waterfall model process is simple to implement and also the amount of resources required for it are minimal. In this model, output is generated after each stage (as seen before), therefore it has high visibility. Deadlines can be set for the completion of each phase and evaluation can be done from time to time, to check if project is going as per milestones. It provides a template into which methods of analysis, design, coding, testing and maintenance can be placed. This methodology is preferred in projects where quality is more important as compared to schedule or cost. The waterfall model process is perfect for Mr. Television process. The drawback of this process is, it is often difficult to get customer requirements explicitly. Even a small change in any previous stage can cause big problem for subsequent phases as all phases are dependent on each-other.

3.5 Tailoring of Software Development Lifecycle and Process

Tailoring Summary		
Process Step or Artifact	What was Changed	Rationale for the Change
Entry and Exit Criteria	Toward the end of every stage, check that the exit criteria have been met. Toward the start of every stage, check if all the entrance criteria have been met.	Each phase is started only if it satisfies the entry criteria. Also check if a phase is really completed.
Review phases	After every stage, there will be a phase review to ensure that the phase is effectively finished	This avoids re-work .
Overlap Phases	As and when task in one stage are finished for an activity, the next stage task for that action will start. Ex: while Requirements task are going on, Design task may be begun for those requirements that are firm, clear and surely knew.	This will reduces the duration since we are starting early(overlapping the phases).

3.6 Technical and Management Reviews

Review	Purpose	Attendees	Evaluation Criteria	Exit Criteria
Requirements review	Corporate will be hesitant to give significant funding until we finish this review.	Software Engineers, Team lead, Managers	Each requirement can be followed to the requirement trace matrix. All requirements are Complete, sufficient, Consistent, Unambiguous, Testable and so on.	Must have complete requirements from Mechanical, Electrical and System Engineering departments.
Top Level Design Review	This would be useful for our validity with corporate.	Design Engineers, Team lead, Managers	The top level design of the components satisfies top level design standards.	Initial mechanical, electrical design and software architecture must be complete.
Final Design Review	Needs to know where we remain with a specific end	Design Engineers,	Final design of	Initial mechanical,

Review	Purpose	Attendees	Evaluation Criteria	Exit Criteria
	goal to choose item strategy.	Team lead, Managers	the components satisfies design standards.	electrical design and software architecture must be complete.
Coding(construction) Review	To give the developed programming for testing need to guarantee that everything has been built in as per requirements and design.	Developers, Team leads	A completed and reviewed program design - Source program listing - Code Review checklist - Coding standard - Defect Type standard.	Software for Video, digital processing and Translation have been completed.
Integration Review	To integrate the components with the hardware.	All Managers, Team lead, Engineers	The product satisfies the performance and memory requirements specified by the Functional Spec. All priority a bug have been fixed and closed.	Video depends on video hardware; digital processing depends on processor hardware; translation depends on camera hardware. All these are integrated into hardware

3.7 Software Development Environment

All software will be written in C, Java or Python

Phase	Tools
Requirements	MS Word document
Design	MS Excel
Coding	C, Java, Python
Testing	White box, Black box

Integration	Maven, ETL testing
Hardware & others	Printers, scanners, Computers, network, portal or IM for staffs to contact each other

3.8 Verification, Testing and Integration

It is the process of evaluation a software item to detect differences between given input and expected output. Testing assesses the quality of the product. Testing is a process that should be done during the development process. It is also called as a verification and validation process. Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. A test plan is developed by the manager with the inputs from test engineers about the strategy that will be used to verify and ensure that the product meets its design specifications and other requirements.

Testing technique	Description	Responsibility
Black box testing	Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system.	Test engineers
White box testing	White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.	Developers/Unit testers
Automation testing	In software testing, test automation is the use of special software to control the execution of tests and the comparison of actual outcomes with predicted outcomes.	Automation test engineers
Integration Testing	Integration testing is testing in which a group of components are combined to produce output. The tests are based on the integration testing plan.	Integration test engineers

3.9 Evaluations

Evaluation is very important step, in which system analyst has to make sure everything goes according to plan. So at the end each component system analyst with meet with team in a form of general meeting so they remain on same page. This PEER review includes engineer, QA, Developer, SW Manager and other expert engineer.

Name of the Artifact	**Generated by : (Process Step)**	**Evaluation criteria**	**Evaluation Type**	**People Involve**	**Used By: (Process Step)**
Requirements Analysis	Requirements	Does all the requirements are mentioned to start design phase	Requirements Review	Requirements engineers, Managers	All Phases
High level design document	Design	To define high level architecture of Mr. Television project in clear manner	Design Review	Design Engineers, Team lead, Managers	Design Phase
Low Level Design	Design	To determine how to do "Mr. Television" project in a clear and logical manner	Design Review	Design Engineers, Team lead, Managers	Design Phase
Coding	construction	Coding is based on requirements and design. Does it satisfy all requirements?	Inspection	Developers, Team Lead	Coding Phase
Testing of documents	Testing	All the test cases are executed??	Testing Review	Software testers, developers	Integration
Implementation	Integration	Does the code being executed in the	Integration Review	All Managers, Team lead, Engineers	

		equipment legitimately??			

3.10 Process Risks

Process Risk Summary				
Risk Id	Risk	Evidence	Kind of Risk	Description
1	Risk that the processors will need more modifications.	mrtvswdesignconcept(1)- Operating system This software, although "tried and true", will probably require more modification than the systems engineers believe.	Schedule	When there are modifications at a later stage, then the changes would need to be incorporated again which would lead to cost and schedule changes. For example: During coding phase, when they figure out that there are some modifications then the same has to be incorporated in the previous phases. The more the previous phases are re-visited the more time and cost it takes to complete.
2	Risk that there is no image enhancement requirements for few months after start of project.	06 mrtvmemo(1)- The image enhancement requirements are not likely to be available for several months at the earliest.	compliance	The image requirements will be obtained later. In the waterfall model, when the requirements are obtained at a later phase the cost and schedule would be affected. Also, it will be difficult for the resources to go with the process due to changes.

Process Risk Summary				
Risk Id	Risk	Evidence	Kind of Risk	Description
3	Risk that re-work could occur due to overlap of phases	Tailoring Process	Schedule	Due to overlap of Phases, the Risk is that changes to other requirements may impact the design activity.

3.11 Process Measures

Process Measures Summary		
Measure	What will be Measured	Corresponding Risk
Processor memory	The size estimation of processors	1
Image Enhancements	Measure when the image enhancements will be received. Accommodate changes for image enhancements	2
Overlap of phases	Start tasks only if there are no other dependencies. If there are no dependencies then only start the tasks.	3

Appendix A – WBS for Mr. Television

WBS #	Description and Comments	Source Document Code (1)	Paragraph Number in Source Document
1	Mr. Television Project		
1.1	Requirements		
1.1.1	System requirements	mrtvschedule	4.3
1.1.2	Mechanical requirements	mrtvschedule	4.2
1.1.3	Electrical requirements	mrtvschedule	4.3
1.1.4	Software requirements		
1.1.4.1	Collect software requirements for Processor#1	Mrtvsystemdesign summary	7.1
1.1.4.2	Collect software requirements for Processor#2	Mrtvsystemdesign summary	
1.1.4.3	Collect software requirements for Processor#3	Mrtvsystemdesign summary	
1.1.4.4	Collect software requirements for Processor#4	Mrtvsystemdesign summary	
1.1.4.5	Develop test plan		
1.1.4.6	Generate other artifacts during requirements analysis		
1.1.4.7	Review requirement phase	mrtvswrequirements	7.2
1.2	Design		
1.2.1	Top level design	Mrtvsystemdesign summary	7.3
1.2.1.1	Top level design processor#1	Mrtvsystemdesign summary	7.4

1.2.1.2	Top level design processor#2	Mrtvsystemdesign summary	3.0
1.2.1.3	Top level design processor#3	Mrtvsystemdesign summary	4.0
1.2.1.4	Top level design processor#4	Mrtvsystemdesign summary	3.0
1.2.2	Detailed design		
1.2.2.1	Top level design processor#1	mrtvswrequirements	3.2.1
1.2.2.2	Top level design processor#2	mrtvswrequirements	4.2
1.2.2.3	Top level design processor#3	mrtvswrequirements	4.3
1.2.2.4	Top level design processor#4	mrtvswrequirements	3.2
1.2.3	Design review	mrtvswrequirements	3.2.2
1.2.3.1	Design review for processor #1	mrtvswrequirements	
1.2.3.2	Design review for processor #2	mrtvswrequirements	
1.2.3.3	Design review for processor #3	mrtvswrequirements	
1.2.3.4	Design review for processor #4	mrtvswrequirements	
1.2.4	Produce artifacts for design phase		
1.2.5	Review design phase		
1.3	Coding		
1.3.1	Analyze requirements and design artifacts		
1.3.2	Discuss with team members	mrtvmemo	4.2
1.3.3	Develop Code		
1.3.3.1	Develop code for Processor#1	Mrtvsystemdesign summary	4.1.5
1.3.3.2	Develop code for Processor#2	Mrtvsystemdesign summary	7.1
1.3.3.3	Develop code for Processor#3	Mrtvsystemdesign summary	7.2
1.3.3.4	Develop code for Processor#4	Mrtvsystemdesign summary	7.3
1.3.4	Unit Testing		
1.3.4.1	Unit Testing for Processor#1	mrtvswdesignconcept	7.4

1.3.4.2	Unit Testing for Processor#2	mrtvswdesignconcept	3.4
1.3.4.3	Unit Testing for Processor#3	mrtvswdesignconcept	6.2
1.3.4.4	Unit Testing for Processor#4	mrtvswdesignconcept	3.5
1.3.5	Integrate Components	mrtvswrequirements	3.4.1
1.3.6	Review Coding Phase	mrtvswrequirements	5.1
1.4	Testing	Mrtvswrequirements	5.2
1.4.1	Test Planning	Mrtvswrequirements	
1.4.2	Test case development	Mrtvswrequirements	
1.4.3	Test Environment	Mrtvswrequirements	
1.4.4	Test Case execution	Mrtvswrequirements	
1.4.5	Testing Phase Review	Mrtvswrequirements	
1.5	Deployment	Mrtvswrequirements	
1.6	Configuration Management		

Appendix B – Summary of Size Estimate

Actual Size Table, per P&P

SOFTWARE ITEM: SW SYSTEM	ACTUAL SIZE				
	SOURCE LINES OF CODE OR FUNCTION POINTS				BYTES OF TARGET
COMPONENTS	*MINIMUM*	*LIKELY*	*MAXIMUM*	*EXPECTED*	*MEMORY*
PROCESSOR #1	60864	116583	172227	116570	64091500
PROCESSOR #2	70864	128583	195500	130116	114007500
PROCESSOR #3	60864	116583	172227	116570	64091500
PROCESSOR #4	52042	104728	158000	104826	64040000
Total for SW Item	244634	466477	697954	468082	306230500

Equivalent Size Table, per P&P

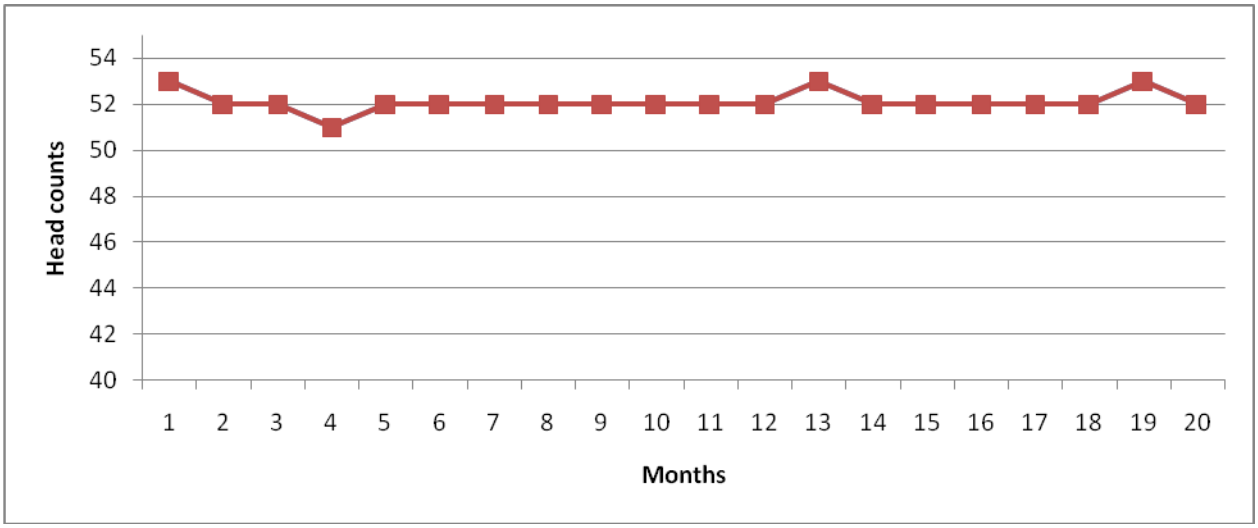
SOFTWARE ITEM : PROCESSOR #1	NEW CODE (SLOC or Function Points)	COTS/REUSED CODE SLOC or Function Pts.			MODIFIED CODE SLOC or Function Pts.			NET EQUIV. SIZE	C	JAVA	Python
COMPONENTS		ORIGINAL	EQUIV. *RATIO*	EQUIV.	ORIGINAL	EQUIV. *RATIO*	EQUIV.				
PROCESSOR #1	11000	93371	0.3	28011	12199	0.6	7319.4	46330	46330	0	0
PROCESSOR #2	21667	101250	0.02	2684	5000	0.6	3000	27351	2684	0	24667
PROCESSOR #3	5000	106570	0.02	3621	5000	0.6	3000	11621	11621	0	0
PROCESSOR #4	0	104826	0.02	2096	0	0.6	0	2096	2025	0	71
Total	37667	101504		36412	22199		13319	87398			

Appendix C – Summary of Effort Estimate

Effort Schedule

Phase	Effort	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20	TOT
Rqmts	2842	568	710	710	852																	2842
T Des	2131					852	639	639														2131
D Des	2131								852	639	639											2131
C&UT	2842											994	994	852								2842
Integ.	2131														852	639	639					2131
Confg. Mng	710																	355	355			710
Q.A	1421																			852	568	1421
SUB-TOTAL (ABE)	14212	568	710	710	852	852	639	639	852	639	639	994	994	852	852	639	639	355	355	852	568	14212
ADD-ONS -3	5665	430	280	282	135	142	350	352	144	354	349	3	1	147	145	350	352	640	635	146	428	5665
TOTAL EFFORT	19877	998	990	992	987	994	989	991	996	993	988	997	995	999	997	989	991	995	990	998	996	19877
Head-count (4)		53	52	52	51	52	52	52	52	52	52	52	52	53	52	52	52	52	52	53	52	

Headcount Graph after Adjustments



Appendix D – Risk Management Plan

Overview:
A risk management plan will be created and maintained by the Project Manager based on the discussion with Team leads and other stakeholders. In this meeting, all the potential risks are identified, mitigation plans are developed and then the status of every risk is identified. The Risk management meetings will be conducted every month. The risk management plan mainly has two parts - Risks lists and then the plan and procedures.

Risk Assessment:
Risk Identification:
There are three types of risks - Project, Product and Process.

Risk ID	Risk Description	Kind of Risk	Category
1	Risk that there is not enough memory so system will not be able to work properly.	Technical	Product
2	Risk that there is no recent version of operating system.	Technical	Product
3	Risk that assumptions of requirements of sound enhancement and image enhancement are same	Technical	Product
4	Risk that the prototype is defective, leading to issues in product.	Technical	Product
5	Risk that the processors will need more modifications	Schedule and Cost	Project
6	Risk that there is no image enhancement requirements for few months after start of project	Schedule	Project
7	Risk that the prototype is defective, leading to schedule delays	Schedule	Project
8	Risk that design, development and testing impacts of safety standards will not be understood by the project team early enough, leading to rework	Schedule and Cost	Project
9	Risk that further changes cannot be made	Schedule and Cost	Process

10	Risk that re-work could occur due to overlap of phases	Schedule	Process

Risk Causes:
There is a reason for risk to occur. For the risk not to occur, the causes should be identified.

Risk ID	Risk Description	Causes
1	Risk that there is not enough memory so system will not be able to work properly.	The memory is less
2	Risk that there is no recent version of operating system.	Recent version not been delivered.
3	Risk that the prototype is defective, leading to issues in product.	Defective hardware.
4	Risk that assumptions of requirements of sound enhancement and image enhancement are same	Not enough data
5	Risk that the processors will need more modifications	Not enough memory
6	Risk that there is no image enhancement requirements for few months after start of project	No knowledge about Image Enhancements and requirements unclear.
7	Risk that the prototype is defective, leading to schedule delays	Prototype not completed
8	Risk that design, development and testing impacts of safety standards will not be understood by the project team early enough, leading to rework	There is no training given to the team regarding the safety standards.
9	Risk that further changes cannot be made	Due to waterfall model
10	Risk that re-work could occur due to overlap of phases	If the process is tailored to overlap phases, then there would be more re-work.

Risk Likelihood and Prioritization:

Risk	Likelihood (1-10) scale	Impact (1-10) scale	Weighted Cost
Risk that there is not enough memory so system will not be able to work	7	6	42

properly.			
Risk that there is no recent version of operating system.	9	5	45
Risk that assumptions of requirements of sound enhancement and image enhancement are same	9	4	36
Risk that the processors will need more modifications	7	7	49
Risk that there is no image enhancement requirements for few months after start of project	9	9	81
Risk that the prototype is defective, leading to schedule delays	8	8	64
Risk that further changes cannot be made	9	8	72
Risk that re-work could occur due to overlap of phases	9	7	63
Risk that design, development and testing impacts of safety standards will not be understood by the project team early enough, leading to rework	9	6	54

Risk Mitigation:
A mitigation plan is a good idea to know what must be done if the risk occurs.

Risk	Risk Mitigation
Risk that there is not enough memory so system will not be able to work properly.	The size must be reduced
Risk that there is no recent version of operating system.	Have a schedule prepared for this. Make sure OS in installed on time by following up with the company.
Risk that assumptions of requirements of sound enhancement and image enhancement are same	Have a discussion and figure out clear requirements of Image and enhancement and Sound Enhancement.
Risk that the processors will need more modifications	Use less memory for all the modifications. Increase the size of the memory.
Risk that there is no image enhancement requirements for few months after start of project	Use the requirements of Sound enhancement after a detailed research.
Risk that the prototype is defective, leading to schedule delays	Add more researchers.

Risk that further changes cannot be made	Freeze the requirements.
Risk that re-work could occur due to overlap of phases	Add more business analysts to predict the re-work that might occur so that they are covered in the respective phases. Freeze the requirements.

Contingency Plan:
A contingency plan tells what should be done if the risk occurs.

Risk	Contingency plan
Risk that there is not enough memory so system will not be able to work properly.	Add more memory.
Risk that there is no recent version of operating system.	Allocate resources who are responsible for the new OS.
Risk that assumptions of requirements of sound enhancement and image enhancement are same	Use the requirements of sound enhancement. Add more resources to research on sound and image requirements
Risk that the processors will need more modifications	Add more memory.
Risk that there is no image enhancement requirements for few months after start of project	Use the requirements of sound enhancement.
Risk that the prototype is defective, leading to schedule delays	Other tasks not dependable on Prototype must be started until the prototype is completed.
Risk that further changes cannot be made	Have a separate schedule and allocate resources to work on it
Risk that re-work could occur due to overlap of phases	Add more resources to work on the re-work so that the original schedule is not affected.

RACI Chart
Requirements Assignment matrix describes the participation with respect to different roles.

	Manager	Lead	Team Members
Risk Identification	A	R	C
Risk Analysis	A	R	C
Risk Prioritization	A	R	I
Risk Mitigation	R	C	I
Risk Contingency	R	C	I

Risk Monitoring and control

New Risk Identification - When new risks are identified by any stakeholder should be reported	All
Monitoring - Individual/Team who identified the risks	Risk Owner
Review - Meeting every month to discuss the priority/status of the top 5 risks.	Team members

Appendix E – Measurement Plan

Risk	Weighted Cost	Measures	How it is gathered
Memory	42	Rate chart. Predicted memory space used.	Based on the memory available after changes being made.
operating system.	45	Rate chart	Memory and efficiency of the OS is being computed by QA and feedback from developers during the monthly status meeting.
Requirements	36	Rate chart	Based on inputs from team members and stakeholders. Updates are discussed and added every week.
Processors	49	Rate chart	After major changes made in the processor, the developers would report the processor efficiency. Also, would be done by QA expert during testing.
Schedule	64	IMS	Based on the IMS and team progress
Rework	63	In phase and Out of phase defects	Based on the analysis by Business analysts and status meeting.
Defect	54	Defect containment chart	Defect report that is being reported by QA.

Opportunities to measure:

Name	Opportunity	Measure
Image Enhancement Technology	Image enhancement would help in better video quality which leads to product success.	Product sale
Skilled Staff	There are many skilled resources in the project they would help to reduce risk.	Schedule and quality

a. Measure for Memory

There are four processors each containing 64MB memory. The OS along with processor function are installed in each of the processor, the memory used would not be identical for all the processors.

Objective:
To measure the planned memory and the actual memory consumed .

Measure:
Planned memory vs. actual memory

Impact:
Cost

How Frequently the data is collected:
Planned memory: during the size estimation
Actual memory: during coding phase

Who is responsible:
Software Engineer and Project Manager

Calculation:
After installation in the processor memory consumed to the planned memory.

Action Plan:
When the memory is exceeding or under the limit as planned, the contingency plan must be executed. some of the ideas include memory adjustments, code changes, changing scope, hardware changes.

b. Measure for Requirements:

Requirements may change over time or new requirements may be added due to various factors like feasibility, market trend, cost, schedule, etc. After the requirements are planned, the more additional requirements the more the rework and chaos.

Objective:
To measure the planned requirements and the changed/added requirements in the future.

Measure:
Planned requirements vs. Actual requirements

Impact:
Impact schedule

How Frequently the data is collected:
Planned requirements: after requirements.
Actual requirements: Every month after the monthly status meeting.

Who is responsible:
Requirement Engineers, Project Manager and Program Manager

Calculation:
Calculate the total number of requirements(planned) to the number of new requirements added/changed.

Action Plan:
The requirements must adhere to the scope. Any changes should be documented and approved by stakeholders. There should be a good change control protocol in place. Requirements must be reviewed periodically (Every two weeks) and there should be a tracking mechanism so that any new or old requirements can be traced.

Uses:

- Greater comfort and confidence level from key stakeholders
- Reduced likelihood of missed or misinterpreted requirements
- Better estimates of the time needed for requirements activity
- Increased reusability of requirements.

How to measure ?

Break down the requirements and identify who will measure those requirements. Plan the requirements measurement carefully ,document the results of the measurements, and take appropriate corrective action when necessary.

c. Measure for Schedule:

A good schedule is a plan which tells what tasks to occur when. Schedule is represented by IMS.

Objective:
This is used by the project manager to know if we are on time and the progress of each task. Any delay in the project could be identified.

Impact:
Schedule and cost

How Frequently the data is collected:
Every week during the weekly status meeting

Who is responsible:
Team leads and Project Managers

Calculation:
Compare the planned dates with the actual dates.

Action Plan:
Objective:
To meet milestones.
Metric:
variance between planned date and forecast date.
How Often to measure schedule:
Weekly. This helps us determine if we are going to meet our goal. There are some other indicators we should be measuring that might suggest that the schedule is at risk.

The schedules may go out of hand and proper action should be taken to bring back the schedule on time. a proper risk management plan would help. Changing the scope, modifying the schedule, adding resources will be useful.
Network Chart which combines the benefits of PERT and GANTT will be useful to understand the duration of tasks, their interdependencies and the critical path. But network chart won't tell if the schedule is realistic or not.

d. Measure for Rework:

If the requirements are changed or something went wrong in the past tasks/phases then there will be rework.

Objective:
To evaluate the progress and to evaluate the schedule.

Impact:

Cost and Schedule

How Frequently the data is collected:
After the end of each phase in the phase review meeting.

Who is responsible:
Team leads, Project Manager and Team members

Calculation:
Use the defect containment chart for each phase.

Action Plan:
Rework is one of the most important attribute to measure. Less rework denotes good project management, reduces cost. Any rework should be measured and understood to avoid future rework.
'Count problem' method can be used to measure rework.
Estimate of total rework = The sum of all the effort.

e. Measure for Defect:

Defect can be anything that deviates from requirements or something that causes the product not to function/perform as expected**.**

Objective:
To measure the product quality.
Impact:
Quality

How Frequently the data is collected:
During testing- weekly status meeting

Who is responsible:
Testing lead, Developer lead and Project Manager

Calculation:
No of closed defects and No of open defects

Action Plan:
Defects goes through a life cycle. It can be in many states like New, Assign, Open, Test, Verified, closed, Reopened, Deferred, Rejected. Some defects can be missed in the initial stages but later found also, which should be avoided as it would lead to rework/failure.
Defect density is used for measuring the quality of the product.
Defect density = Number of Defects/ Total size.
How often to measure Defect Density:

- During the start of every month and before the end of each phase

.

The size is measured in one of the following:

- Function Points(FP)
- source Lines of code

The advantages of measuring Defect density are:

- Used to compare the relative number of defects in various software components so that high-risk components can be identified and resources focused towards them.
- Used to compare software so that quality of each product can be quantified and resources focused those with low quality.

Appendix F – Software Configuration Management Plan

Overview:

Software Configuration Management is the process of identifying, organizing and managing modifications to software .The purpose of configuration management (CM) is to maintain integrity of the product by controlling change to the product. This section will describe the major configuration management tasks to be performed (description, purpose, artifacts used and produced), who will be responsible for each task, when the various tasks will be performed, and how (what procedures).

Configuration management (CM) refers to a discipline for evaluating, coordinating, approving or disapproving, and implementing changes in artifacts that are used to construct and maintain software systems.
There are five major functions of Configuration Management:

1.Configuration Identification: Process of naming and defining the components.
2. Configuration Control : Changes that are made in software there would be one official copy of that.
3.Change Management: Makes sure that the changes do not cause unintended problems.
4.Status Accounting: It maintains descriptions and status information of each component.
5.Configuration authentication: Verifies that the software passes its tests.

Artifacts:
The artifacts are Requirement documents, Design documents, Design models, Software code, Test plan ,etc.

Software Configuration Management Roles for Mr. Television:

Role	Responsibilities	Participants
Configuration Manager / Engineer	Responsibilities will be to : **The configuration manager is the central control point for all changes** • Coordinates with external organizations to proactively identify software and documents, including updates/fixes and customizations, as required. • Defines/communicates/enforces procedures • Assures that changes are properly tested • Manage upgrades to the	Member of CM staff

	configuration management tools. • Establish processes and procedure for build and promotion management and communicate to single affected group. • Mostly he Configures and operates Configuration management tools	
Module Owner	**Module owner is the software technical lead or representative senior engineer who has great learning on the technical viewpoints like configuration, conditions and libraries for the module**. The module owner has technical responsibility: • Knows and understands the design • Maintains and controls access to the change tracking system (including CRs) • Supports all activities relating to CM tools • He is responsible for the integrity of the module and its success. He supervises changes and does regression tests after any major change to the module. • Technical control point for changes, fixes, enhancements	Software Technical Lead
Change Control Board	Change control board oversees changes to distinctive software items, components and modules. This is critical in ensuring that unintended changes or bugs don't enter generation. **Responsibilities of the Change Control Authority:**	Software Technical Lead, Software Manager, Configuration Manager, QA Manager, Representatives of all Parts of the Software

	• **Controls, evaluates, authorizes and coordinates changes.** • Includes representatives from all affected functions, such as development, documentation, quality assurance, maintenance, test, etc. • Whether the changes can be accepted after the change is implemented • Authorizes incorporation of changes after suitable tests	

How to Plan a Software Configuration Management (SCM) Program:

1) Analyze the Current Software Process:
To find out which are the artifacts that should be under configuration management control and the decision of when should they be under different levels of control.

2) Define the software configuration management Process and Procedures:

- Define configuration control functionality
- Build the change management procedures
- Establish the responsibilities so that the control mechanism can be implemented.

3) Document the Plan:

- Fix and resolve all the problems that must be documented in an Software configuration management plan.
- Define processes and procedures for all configuration management activities
- Document all software configuration management functions, risks, obligations and automation needs in one place to serve as a kind of perspective point in the future

4) Evaluate CM Tools:

a. Tools can do many things to support CM

- Support change request and approval procedures
- Support auditing, managing and tracking
- Provide a configuration management library

b. The tools that support Software configuration management should be able match the requirements of the Software configuration management process.

c. The tools should support all of your Software configuration management needs

5) Perform a Pilot of Any New Approach:

- It is important to understand the weaknesses and correct any significant problems before applying a Software configuration management system to a large scale project.

6) Deploy the Software configuration management system:

- Train the resources.

7) Facilitate ongoing improvement

Problems with Implementing Software Configuration Management :

1.Three Important Prerequisites for Implementing Software configuration management:

a. Management Support:

Configuration management can be an expensive function.

Management support is very important to get resources and to enforce the configuration management policies in the development process .

b. Repeatable Process:

The risk of inappropriate configuration management process is very high ,without an existing repeatable process. This causes many problems so Repeatable process is important.

c. All Risks Identified:

Before enforcing the software configuration management process all software configuration management related risks should be identified.

2.Cost vs. Benefit of CM:

Configuration management can be an extravagant function. CM adds time and overhead to a task. If it is done too early then it will result in more cost for less benefit. On the other hand if it is done too late then it may result in chaos. CM is an management work however is regularly performed by software developers who may not see much about management.

3.Adequacy of tools:

Some CM tools are inadequate, slow and expensive and cumbersome to use. In other words they may not be capable for large projects. On the other hand modern tools are difficult to learn and use. They may not incorporate well with the rest of the software development tools and particularly not with different tools utilized on a substantial project, for example, system engineering or hardware design tools.

4.Software CM vs. other CM:

Conceptually software CM is similar to harder CM or project level CM. But practically software CM is different form what is needed elsewhere on a large project. The elusiveness of software has a tendency to oblige more record keeping to keep things from getting stirred up. The complexity of software often requires more sophisticated tools.

RACI Chart:

A RACI Chart is a way of documenting the roles for all important tasks that involve more than one person, so everyone knows who does what?

- Responsible – the person or role who does all the work
- Authority – the person or role who approves the work
- Consultant – someone who should be consulted due to their expertise
- Informee – someone who "needs to know" that the task is being done by others.

Appendix G – Quality Assurance Plan

Overview:

Software Quality Assurance(QA) is a process of validating the software products. The purpose of the quality assurance is to provide management and developers with visibility into the process being followed and the products being built. The goal of QA is to provide assurance that a product will meet the customer's quality expectations. It consists of activities designed to ensure that quality will be built into the product.

It is a QA responsibility to develop and implement processes and standards to improve the development life cycle and to make sure that these procedures are followed.

Quality assurance plan will include:

1.Responsibilities and roles.

2. The process and procedures

3.How decisions were made ,especially when there are disagreements

4.What tools will be used

5.What training is required

6.How records will be maintained.

Typical Practices of Software Quality Assurance:

1.Inspetion:

To evaluate whether the product meets this specified requirements and standard inspection procedure. Commonly inspected work products include Software Requirements Specifications and Test plans. The work product is selected for review and inspection meeting is carried out to review the work product. A moderator is chosen to moderate the meeting. The goal of the inspection is to identify defects. Defect in inspection is any part of the work product that will keep an inspector from approving it.

Stages in the inspection process are:

Stages in the inspection process	Roles
Planning	Moderator
Overview meeting	Author
Preparation	Inspector
Inspection meeting	Reader
Rework	Author
Follow-up	Author

When it satisfies some predefined criteria the process is ended by the moderator.
Inspection Roles:

- Author: Person who creates the work product being inspected
- Moderator: This is the leader of the inspection who plans the inspection and coordinates it.
- Reader: The person reading trough the documents, one item at a time.
- Recorder: The person who documents the defects that are found at inspection.
- Inspector: The person examines the work product that identifies possible defects.

2. Reviews:

There are different types of reviews Peer review and Code review.

Peer review:

Peer review is a type of software review in which a work product is examined by its author and by others at the same organization level(team members).The benefits of peer reviews are: It reduced rework and the development is faster due to more effective evaluation. Peers knows software very well and they will not be as much of a threat when it comes to promotions.

A minimum of one peer review will be performed before deliverable in Mr. Television Project.

Cost:

May require 5-10% of the total development cost and so this must be planned into schedule.

Code review:

In a code review the team examines a sample of code and fixes any defects in it. In a code review, a defect is a block of code which does not properly implements its requirements.

Status review:

To know the status of the project status review. Useful in tracking and project overview.

Participants:

Software Engineers, Business analyst, QA testers, QA managers.

3.Audits:

It evaluates whether Quality Assurance processes or set according to standards and company practices. The main objective of an Audit is to determine the adherence to established standards and procedures; checking their adequacy or effectiveness is a secondary objective that usually is not requested of an auditor. Auditor is that the one who reviews and checks the project management activities are executed to the highest possible standard.

Benefits of Audits:

- Monitoring and improving the process
- Make sure that the standards and procedures are followed
- Preventing the quality problems from occurring

Roles of an Audits:

- The Initiator
- The Lead Auditor
- The Recorder
- The auditors

Who should attend:

- Product Managers
- Quality Managers

How Often we do Audits:

Weekly

4.Communication:

An effective communication among the team is essential to avoid chaos.

5.Measurements:

Measurements are used so that decisions are based on fact.

6.Indipendent confirmation of compliance:

Standards, requirements, processes and procedures.

Responsibilities of Quality Assurance:

- Review all development plans for completeness
- To make sure the tests are appropriate and are run correctly.
- Determine adherence to standards.

- Inspect the product.
- Evaluate the support processes
- Monitor reviews, inspections, walkthroughs, etc.
- Find root causes when defect are found.

Roles And Responsibilities :

Roles	Responsibilities
QA Manager	Manages the Quality Assurance function.
System Owner	Helps to define product quality expectations.
QA Consultant	Audits and approves project deliverables from QA perspective. Provides guidance and assistance on process matters.
Project manager	Ensure implementation of quality activities. Provides regular and timely communication.
Project Managers manager	Monitors implementation of quality activities. Resolve conflict across organization.

Verification and Validation:

verification: The process of determining whether or not the products of a given stage of the software development cycle fulfill the requirements established during the previous stage. (IEEE definition)
The following activities are performed as part of verification:

- Produce a traceability matrix Tracing all requirements
- Evaluate requirements and relationships
- Access the criticality of requirements.

validation: The process of evaluating software at the end of the software development process (acceptance testing activity in the Testing stage) to ensure compliance with software requirements. (IEEE definition)
The following activities will be performed as part of requirements validation:

• Plan acceptance testing, including criteria for:
  •compliance with all requirement
  • adequacy of user documentation
  • performance at boundaries and under stress conditions.
• Plan documentation of test tasks and results.

• Execute the Acceptance Test Plan.
• Document acceptance test results.

QA Milestones:

Stage	**Duration(in Days)**	**Deliverable**	**Work Product**	**QA Activity**
Planning	5	Project plan	Project Schedule	Review processes and audit
Preparation	1	Functional Requirements	Revised Project Plan	Review processes and audit
Software Design	5	Design document	Traceability matrix, Revised Project Plan	Review processes and audit, Trace design components and requirements(forward and backward)
Programming and Integration	5	System documentation(Initial)	Revised Project Plan	Review processes and audit
System Testing and acceptance	20	Test results, System documentation(final)	Revised Project Plan	Review processes and audit