Comparison of Different Development Approaches for Medical Systems (Case Study – EDS, EMG)

A project report submitted in partial fulfillment of the requirements for the degree of Master of Philosophy in Computer Science(R32)

By

Manisha Suryawanshi

QR2003622

N. Wadia College,Pune

Guide

Prof. Dr. Pravin Bhadane

Yashwantrao Chavan Maharashtra Open University,
Nashik 422 222

Year 2009

EVALUATION CERTIFICATE

This is to certify that the undersigned have assessed and evaluated the Research Project **'Comparison of Different Development Approaches for Medical Systems(Case study- EDS, EMG) '** submitted by Mrs. Manisha Sanjay Suryawanshi. The project report has been accepted for the partial fulfillment of the degree of Master of Philosophy in Computer Science.

Internal Examiner

Signature

Name Dr. P. K. Bhonde

Date 5/9/2009

External Examiner

Signature

Name Dr. C. S. Dhole

Date 05/09/2009.

DECLARATION BY THE RESEARCH STUDENT

I hereby declare that this Project Report **'Comparison of Different Development Approaches for Medical Systems (Case Study – EDS, EMG) '** submitted by me is based on actual work carried out by me under the guidance and supervision of Prof. Dr. Pravin Bhadane. Any reference to work done by any other person or institution or any material obtained from other sources have been duly cited and referenced. It is further to state that this work is not submitted anywhere else for any examination.

N.Wadia College, Pune

Signature of the Research Student

Manisha Suryawanshi

QR2003622

CERTIFICATE FROM THE GUIDE

This is to certify that Mrs. Manisha Suryawanshi has completed the Research Project **'Comparison of Different Development Approaches for Medical Systems (Case Study – EDS, EMG) '** under my guidance and supervision, and submitted the project report as laid down by Yashwantrao Chavan Maharashtra Open University, Nashik. The material that has been obtained from other sources is duly acknowledged in the dissertation. It is further certified that the work or its part has not been submitted to any other University for examination under my supervision. I consider this work worthy for the award of the degree of Master of Philosophy in Computer Science.

N. Wadia College, Pune

Signature of Guide

Name: Prof.Dr.Pravin Bhadane

Seal (if any)

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude to Prof. Dr. Pravin Bhadane, Department of Electronics, Nowrosjee Wadia College, Pune-1 for his valuable guidance, unfailing patience and constant encouragement during the course of this work. I am thankful to Dr. Seema P, Dr. Mutalik for their guidance.

I specially thank Dr. S. C. Shirwaikar, M.Phil Course coordinator and Head, Department of Computer Science, N. Wadia College, Pune, for her excellent teaching and guidance during the course work. I thank her whole-heartedly for providing the necessary facilities during the entire course. I also thank all the staff members who taught us this course and introduced me to the world of research, all non-teaching staff for their help and co-operation.

I am thankful to Prof.S.S.Deshmukh, Vice Principal and Head, Department of Computer Science, Prof. S.S.Raut, Prof. M.S.Ghanekar, Prof. A.V.Sathe, Prof. S.N. Badmanji, Shri. Dalvi, Shri Gaikwad, Modern College, Pune-5 for encouraging and motivating me during this course. My special thanks all my friends, my colleagues for their kind support and all those who directly or indirectly helped and encouraged me throughout. I thank all non-teaching staff from our college for their help and co-operation.

Last but not the least, I am extremely thankful to my parents, all my family members for their constant moral support, motivation and help. Thanks a lot for being so kind and grateful.

Mrs.Manisha Suryawanshi

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Overview

In S/W there are many development approaches being practiced by the IT development professionals depending on their specific requirements.

To select the appropriate technique and implement it successfully for any typical project is always a challenge. The challenge is stronger when there are limited resources and time pressures. Output requirements and complexity of the project is another criterion that adds up to the challenge of meeting the deadlines. And when it comes to R&D project it is certain of uncertainty on delivery in time. Different S/W approaches can be used as follows :
Waterfall model in which the customer is ready with his requirements and the final delivery is specified at the beginning. It is simple to implement. It presents development as a sequential process, which proceeds downwards through the phases in the sequence. The original specifications cannot change. It has certain limitations like time delay, changing specifications are not allowed.

Time Boxing is a solution that goes hand in hand with incremental and iterative developments which can be used in combination with other conventional methods. We practice time boxing techniques in our everyday activities. Sometimes we also practice time boxing in project development, e.g. Modular development and testing. But it is limited up to development and testing only. In fact it is the main feature of incremental developments that focuses on business value of the delivery rather than completeness of the project in single go. Similarly in some scenarios waterfall model and other models can be used.

Agile methodology implies adaptability. Agile methods benefit small teams with constantly changing requirements, rather more than larger projects. Agile, changes can be made if necessary without getting the entire program rewritten. This approach not only reduces overheads, it also helps in the upgrading of programs.

1.2 Objective

Objective of this project

To study different development approaches such as Time Boxing, Waterfall Model and Agile to find its effectiveness depending on different scenarios/case study.

To demonstrate the effectiveness of these approaches in special kind of projects like embedded biomedical products for timely deliveries of business value outputs, while tackling the issues of heavy requirements, changing requirements, R&D delays, time constrained deliveries and limited resources, by the way of Practical application of time boxing and analysis of final outcome. Usage of Agile model where changes can be made at as per customer requirement and then the system can be deployed within the specified time.

1.3 Methodology

A very simple methodology has been used to evaluate the effectiveness of these approaches. The major criteria for consideration are 'On time delivery of business values' and complete requirement fulfillment.

The major steps are listed below:

- Overview of different development approaches
- Evaluate the need and suitability of particular approach for specific type of project. Consider the nature of project, its constraints, advantages and disadvantages of the suggested model for this evaluation
- Select the appropriate task / module candidate for the given approach based on above evaluation
- Determine the optimum model of implementation for entire development life cycle with consideration of analysis, prioritization of tasks on the basis of customer requirements, customer priorities, and complexity of the task etc.
- Planning and implementation of live project/s with suitable approach.
- Evaluate the results with respect to business value, time and completeness of design output with quality. Evaluation to be done by simple methods based on measurement of business values, statistical deviation measurement between scheduled activities versus

actual results. Data to be generated from internal and external customer satisfaction feedbacks, project plans and reviews during the SDLC.

1.4 Analysis, Benefits and Usefulness

The analysis is done by

- Measurement of the deviations of on time delivery with respect to its business values at each iteration.
- Comparison with results of similar products developed earlier using other techniques.

The analysis has demonstrated that the different models are effective under different circumstances for the development of respective project. A well optimized mix of conventional waterfall approach, Agile and Time boxing techniques, with the help of incremental and iterative development, can be effectively used to improve the on time deliveries of business values. Waterfall model can be used if the client requirements are known clearly before starting the project and delivery is only after all modules are completed. In Time boxing incremental and iterative focuses on business value of the delivery from time to time while in Agile modifications can be made wherever desired without writing the full program again.

In the next section a chapter wise summary is given

1.5 Organization of Thesis

This dissertation is organized into different chapters as follows:

Chapter 1 – Introduction takes us through overview of the project. It briefly explains the objective of the project. It also covers the benefits, analysis and usefulness along with methodology.

Chapter 2 – Development Approaches touches upon various development approaches like Waterfall Model, Agile Model and Time-boxing techniques. While doing comparative study of these approaches it covers concepts, advantages and disadvantages of one approach over the others approaches. This enables the project managers to choose right approach for project execution suitable for specific business scenario.

Chapter 3 – Literature Survey takes us through some past studies completed by expert which sets a very good base for this research to give new direction. It provides a detailed background of ECG and related topics. It also gives details on Time-boxing technique including various implementation scenarios, methods, dos and don'ts etc.

Chapter 4 – Case Studies demonstrates two special business cases - EDS (ECG Diagnostics System) and EMG System. In this Chapter both business systems have been discussed in detail covering features, implementation scenarios, architectures etc.

Chapter 5 – Requirement Analysis explains in detail the requirements that are gathered from end user community to the fullest possible level. The requirements are gathered from different perspectives like general requirements, functional and technical requirements, software requirements, H/W requirements and so on. Because of the embedded nature of the systems integration and system testing is very important aspect that is addressed during the requirements analysis phase.

Chapter 6 – Conclusion and Recommendations explains in brief the conclusions of the entire study and project. Also it explains some recommendations based on the success of this project. It also gives some directions for further scope on the use of the techniques used in this project.

2 Development Approaches

2.1 The Waterfall Model

It is a sequential software development process which appears like waterfall.[1] It has advantages as after first phase if over it goes to next phase and final delivery is specified at the beginning. The original specifications cannot change. It is simple to implement. It has certain limitations like time delay, changing specifications are not entertained.

In this method the entire project development is planned into systematic stages/steps of activities. Each stage/step is analyzed, planned, designed, implemented, reviewed/verified and validated sequentially so that the delivery of complete project is at the end. Each stage is fully completed then it goes to the next step.

Business value is delivered all at once, at the end of the project.

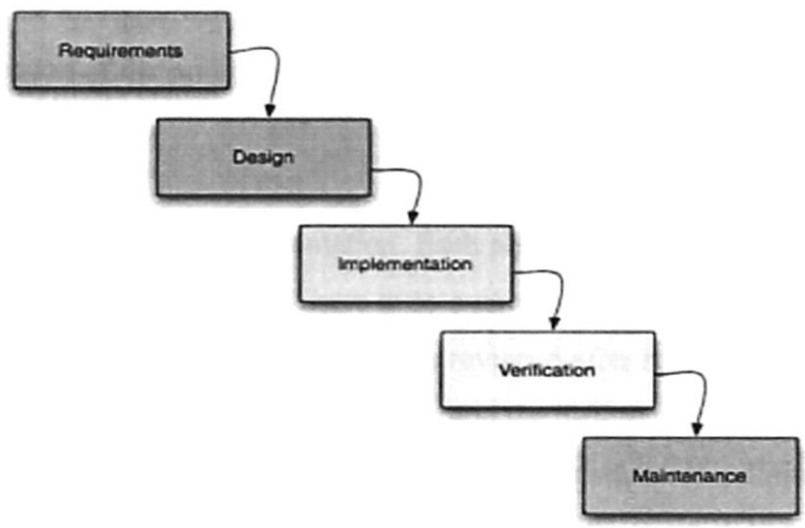

Figure 2.1 : Waterfall Method/Model [1]

[1] Available from: http://en.wikipedia.org/wiki/Waterfall_model

The activities needs to be reworked and may be required to pass through the entire development cycle, if any of the functionality fails or is defective. In this there can be delays in delivery since it waits for completeness of each and every activity before moving on to next stage without focusing the business value of the deliverables.

Advantages:[2]

- Waterfall model provides a template into which methods for analysis, design, coding, testing and support can be placed.
- It remains a widely used procedural model for software engineering.

Disadvantages:

- Real project rarely follow the sequential flow that the model proposes. Although the waterfall model can accommodate iteration, it does so indirectly. As a result, change can cause confusion as the project team proceeds.
- The waterfall model requires all requirements explicitly, but it is often difficult for the customer to state all requirements explicitly.
- A working version of the program will not be available until late in the project time-span.

Waterfall model has a set of phases that a project progresses through in a sequential order. Few resources are required for implementation. Each phase must be completed before the project can progress to the next phase. There is no overlapping between phases. In the waterfall model, the phase is closed and cannot be revisited after its completion. The software development process involves discovery, and the different phases of development often overlap. Waterfall model limits options to correct the mistake: if a limitation of the requirements is discovered during the design phase, it is too late to fix it. In waterfall the disadvantage is there is no back tracking i.e. if there is any error in any stage of software development then it cannot be corrected.

[2] Available from: http://www.blurtit.com/q533918.html

A classically linear and sequential approach to software design and systems development, each waterfall stage is assigned to a separate team to ensure greater project and deadline control, important for on-time project delivery. A linear approach means a stage by stage approach for product building, e.g.

1. The project team first analyses, then determining and prioritizing business requirements / needs.
2. Next, in the design phase business requirements are translated into IT solutions, and a decision taken about which underlying technology i.e. COBOL, Java or Visual Basic, etc. is to be used.
3. Once processes are defined and online layouts built, code implementation takes place.
4. The next stage of data conversion evolves into a fully tested solution for implementation and testing for evaluation by the end-user.
5. The last and final stage involves evaluation and maintenance, with the later ensuring everything runs smoothly.

2.2 The Agile Model

In agile software development since the software is developed in short time boxes called as the development risk is minimized. Iterations are like software project which includes all the tasks as planning, requirement analysis, design, coding, testing and documentation. At the end of iteration it is capable of releasing useful software, delivered in weeks rather than months. The principal measure of progress is working software. At the end of iteration the team reevaluates project priorities. In Agile methods customers, team members, managers or business analyst participation is encouraged on regular basis. There is communication between all project members and encourages locating the entire team in one location. Agile methods emphasize on working software as the primary measure of progress. The project should adapt to changing circumstances, and even late changes in the requirements are welcome.

Iterative Model

The project is constructed in incremental stages such that in each stage new functionalities are added. High priority functionalities can be delivered first accordingly the increments can be planned in each stage.

However, in case a glitch should result, changing the software is not only a practical impossibility, but means one has to go right back to the beginning and start developing new code, all over again. Now, as for minimal risk Agile, it is a low over-head method that emphasizes values and principles rather than processes. Working in cycles i.e. a week, a month, etc., project priorities are re-evaluated and at the end of each cycle. Four principles that constitute Agile methods are:

1. The reigning supreme of individuals and interactions over processes and tools.
2. As does, working software over comprehensive documentation.
3. Likewise, customer collaboration over contract negotiation.
4. And again, responding to change over plan follow-throughs.

Agile methods benefit small teams with constantly changing requirements, rather more than larger projects. Waterfall method stands for predictability, while Agile methodology implies

adaptability. Agile methods are good at reducing overheads, such as justification, documentation etc. keeping them as low as is possible. Agile methods involve planning what one wants and then adapting these plans to the results. Extreme Programming (XP) is an excellent example of Agile methodology i.e.:

1. Communication between customers and other team members;
2. Simple, clean designs;
3. Feedback given on Day 1 of software testing;
4. Early delivery and implementation of suggested changes.

Agile methodology means cutting down the big picture into puzzle size bits, fitting them together when the time is right e.g. design, coding and testing bits. So, while there are reasons to support both the waterfall and agile methods, however, a closer look clarifies why many software and web design firms make the more appropriate choice of employing Agile methodology. The following points enumerate the reason for choosing Agile methodology over the Waterfall method.[3]

1. Once a stage is completed in the Waterfall method, there is no going back, since most software designed and implemented under the waterfall method is hard to change according to time and user needs. The problem can only be fixed by going back and designing an entirely new system, a very costly and inefficient method. Whereas, Agile method adapt to change, as at the end of each stage, the logical program, designed to cope and adapt to new ideas from the outset, allows changes to be made easily. With Agile, changes can be made if necessary without getting the entire program rewritten. This approach not only reduces overheads, it also helps in the upgrading of program.
2. Another Agile method advantage is one has a launch able product at the end of each tested stage. This ensures bugs are caught and eliminated in the development cycle, and the product is double tested again after the first bug elimination. This is not possible for the Waterfall method, since the product is tested only at the very end, which means any bugs found results in the entire program having to be re-written.

[3] Available from: http://agileintro.wordpress.com/2008/01/04/waterfall-vs-agile-methodology

3. Agile modular nature means employing better suited object-oriented designs and program, which means one always has a working model for timely release even when it does not always entirely match customer specifications. Whereas, there is only one main release in the waterfall method and any problems or delays mean highly dissatisfied customers.
4. Agile methods allow for specification changes as per end-user's requirements, spelling customer satisfaction. As already mentioned, this is not possible when the waterfall method is employed, since any changes to be made means the project has to be started all over again.
5. However, both methods do allow for a sort of departmentalization e.g. in waterfall departmentalization is done at each stage. As for Agile, each coding module can be delegated to separate groups. This allows for several parts of the project to be done at the same time, though departmentalization is more effectively used in Agile methodology.
6. In conclusion, though on the plus side, waterfall's defined stages allow for thorough planning, especially for logical design, implementation and deployment, Agile methodology is a sound choice for software development and web design projects.

2.3 Time-boxing

2.3.1 What is Time-boxing

Time-boxing is a time management technique using a fixed deadline. A time box is simply a limited set of time to get the result. It implies in a given block of time how much work can be done. A time box is a fixed unit of development capacity. An easy way to visualize a time box is as a two-dimensional graph. Along the vertical axis is the cost of the development team (per unit time). Along the horizontal axis is time. The longer an iteration is, the wider a time box is. Time Boxing is effectively used in project management to set the delivery goals for small groups of tasks, to set the delivery date for entire projects / programs of work. The prototype are tested and evaluated at the end of time box. From a planning perspective, time boxing is useful, especially when things appear complex or initially and we are unsure of how to begin. Scheduling is divided into smaller, separate, time periods to accommodate the parts of the larger end result desired.

Time Boxing is -

- **Focusing on doing the things that matter most:**

If the time available we have is limited, a rational person should immediately think about prioritizing their outstanding tasks based on what's important and urgent.

- **Limiting the time spent on open ended tasks:**

Because by their nature there is no distinction between done and not done, an arbitrary open ended task can take anywhere between 1 min and 3 weeks. Time boxing is particularly useful as a reality check when working on open ended tasks. By limiting the time we spend on a given task, as long as it is complete though not perfect, we can objectively decide when something is done. This frees us up to work on the next task.

- **As a tool against complex tasks:**

Time boxing is useful here because it allows us to work on complex tasks over several iterations or in bite sized chunks. For example, developing a system with multifaceted functionalities is a complex task. It is difficult to complete in single stretch. So it will be more of more sense to deliver the system in meaningful small pieces / modules with iterative

functionality integrations within the predefined fixed period, constantly refining and repeating this process until the complete system is delivered.

- **As a tool against uninteresting things:**

Time boxing is useful here because it allows us to commit to an undesirable task for only a limited amount of time. It's a lot easier to start working on something we don't like if we knew we only need to work on it for the next 30 minutes. For example, a core developer may not be much interested in maintenance or up gradation tasks. Here instead of trying to get through the entire activity in one go, he may try only doing as much as he can for 30 minutes. When he has another 30 minutes to spare another day and feel so inclined, he can continue.

- **Using free gaps between commitments:**

When there is a gap between commitments i.e. some waiting time or some free brain cycles then that time is considered as null time. Time boxing can be useful during these null times. Sometimes if you know that the code will compile in a few minutes, you can choose to use that time effectively and work on a relatively simple task you which you think can be done within that short null time.

2.3.1.1 Iterative Processes

Iterative development is sometimes referred to as incremental development. The iterative model uses the lessons learned from each phase to modify the results of the previous phase. An iterative process starts with a simple implementation of the project requirements. Iteration adds more functionality until the full design is realized. The design is refined by using the lessons learned during each incremental stage of development. Learning comes from the development process and from experience using the incomplete system, where possible.

Like the waterfall model, the iterative process begins with a requirements phase followed by a design phase and an implementation phase. Lots of iterations might be required before the project is complete. After implementation, an evaluation phase is initiated to evaluate the successes and failures of the work completed. For setting the objectives for the next round of requirements, design, and implementation the user feedback, performance issues, coding difficulties, unclear or inadequate requirements, and program analysis tools are used. Another round of evaluation begins after these phases have been completed for a second time. This process repeats until the project is completed.

Customers often have only a vague idea of what they require. Designers leave out some requirements and make mistakes. Developers frequently find that initial designs do not adequately reflect the hardware, complexity of the problem, or needs of the users. By allowing modifications, iterative processes build flexibility and responsiveness into the process. Several development models are considered iterative. This approach of building corrections, design changes, and discoveries into the process, more accurately reflects how most commercial software is developed.

2.3.2 Time boxing - Concept

To understand the Time Boxing properly, let us review some project development methods suggested and used by the experts.

When the focus is on complete development in single go then Waterfall Model is used while the approaches that focuses on time and deliverables is Time Boxing that uses

1. Incremental development process
2. Iterative development process

The optimized combination of incremental and iterative development process is Time Boxing tool[4].

2.3.2.1 Incremental Development

The development may be started in modular fashion at once, but in a planned and systematic manner.

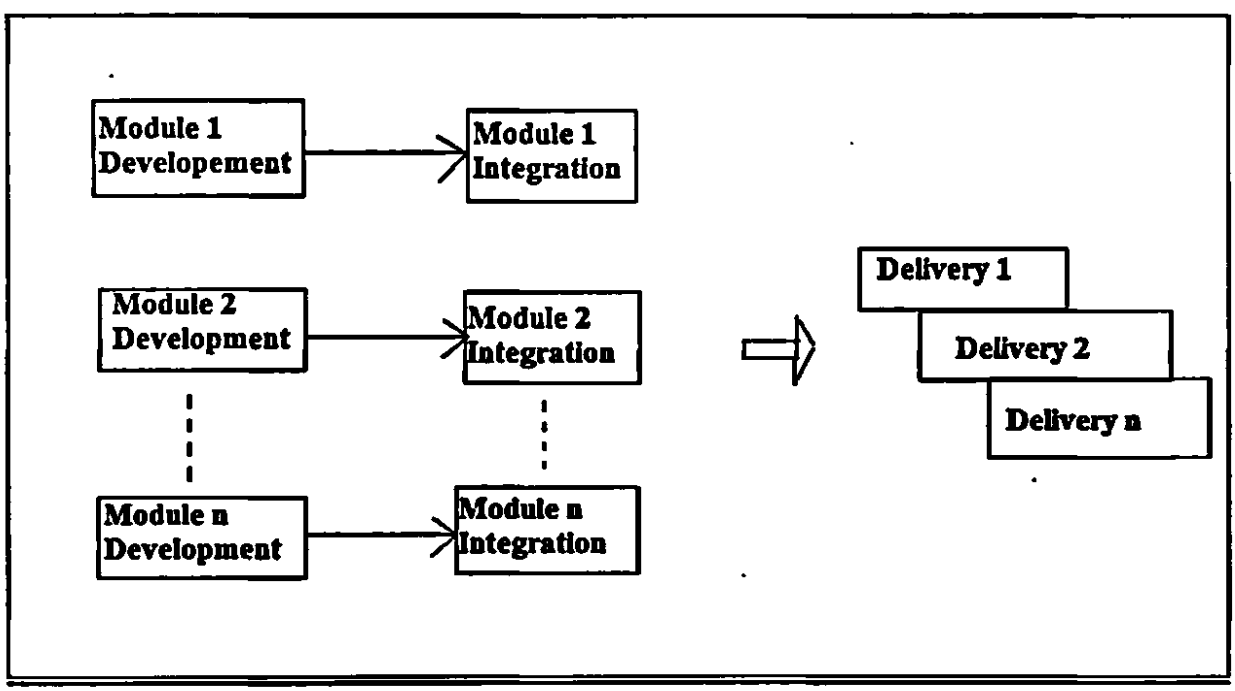

Figure 2.2 : Incremental Development

Incremental development is a scheduling and staging strategy in which different modules of the system are developed at different times and integrated as they are completed. The modules are planned, designed, developed and tested individually. We can use Waterfall

[4] Available from: http://www.cse.iitk.ac.in/users/jalote/papers/Timeboxing.pdf

method for individual module development. The modules are tested then integrated as they are completed. This current integration system can be delivered at this time or you can wait till the entire system is ready. In time boxing approach with incremental development the modularly integrated system deliveries are pre-scheduled in phased manner.

2.3.2.2 Iterative development

Iterative development is a rework scheduling strategy in this extra time is given to recheck and improve parts of the system. It does not presuppose incremental development, but works very well with it. Typically the difference is that the output from an increment is not subject to further refinement, and it's testing or user feedback is not used as input for revising the plans or specifications of the successive increments. On the contrary, the output from iteration is examined for modification, and especially for revising the targets of the successive iterations.[5]

The Unified Process authors have selected the term iterative development, and iterations mean any combination of incremental and iterative development. Iterative development means that they do both incremental and iterative development. Sometimes there might be problem if the project teams do only one and not the other without realizing it. Iterative enhancement means developing a software system incrementally, which allows the developer to learn from the development and use of the system of earlier, incremental, deliverable versions of the system. In this we can start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving sequence of versions until the full system is implemented. During iteration the design is modified and new functional capabilities are added.

The procedure consists of the Initialization step, the Iteration step, and the Project Control List. In initialization step base version of the system is created. The goal for this initial implementation is to create a product which offers a different option of the problem and give a solution that is simple to understand and easily implemented. In iteration process, a project control list is created which has a record of all tasks that need to be performed. It consists of

[5] Available from: http://en.wikipedia.org/wiki/Iterative_and_incremental_development

items as new features to be implemented and areas of redesign of the existing solution. The control list is constantly being revised as a result of the analysis phase.

The iteration involves the redesign and implementation of a task from project control list, and the analysis of the current version of the system. The goal for the design and implementation of any iteration is to be simple, straightforward, and modular, supporting redesign at that stage or as a task added to the project control list. In a light-weight iterative project the code may represent the major source of documentation of the system; however, in a mission-critical iterative project a formal Software Design Document may be used. The analysis of iteration is based upon user feedback, and the program analysis facilities available. It involves analysis of the structure, modularity, usability, reliability, efficiency, & achievement of goals. The project control list is modified in light of the analysis results.

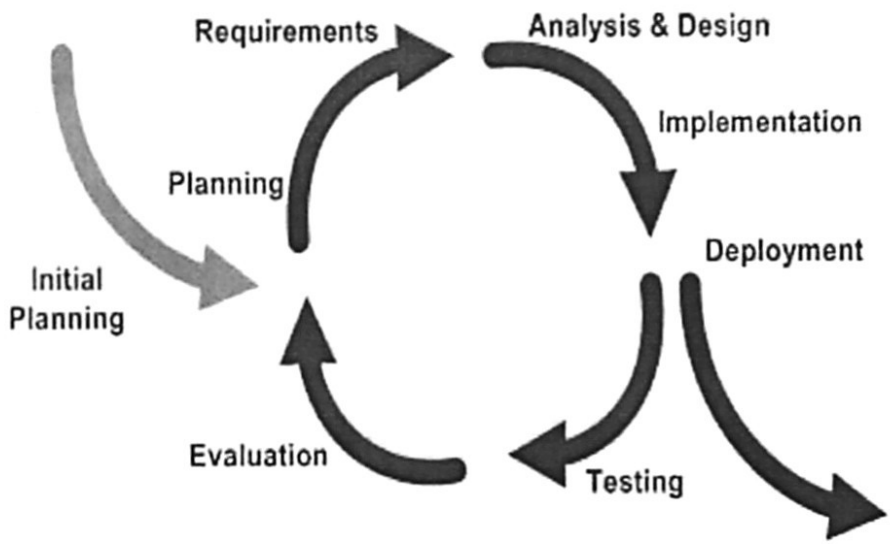

Figure 2.3 : Iteration Process [6]

Guidelines that drive the implementation and analysis include.[6]

- Any difficulty in design, coding and testing a modification should signal the need for redesign or re-coding.

- Modifications should fit easily into isolated and easy-to-find modules. If they do not, some redesign is needed.
- Modifications to tables should be especially easy to make. If any table modification is not quickly and easily done, redesign is indicated.
- Modifications should become easier to make as the iterations progress. If they are not, there is a basic problem such as a design flaw or a proliferation of patches.
- Patches should normally be allowed to exist for only one or two iterations. Patches may be necessary to avoid redesigning during an implementation phase.
- The existing implementation should be analyzed frequently to determine how well it measures up to project goals.
- Program analysis facilities should be used whenever available to aid in the analysis of partial implementations.
- User reaction should be solicited and analyzed for indications of deficiencies in the current implementation.

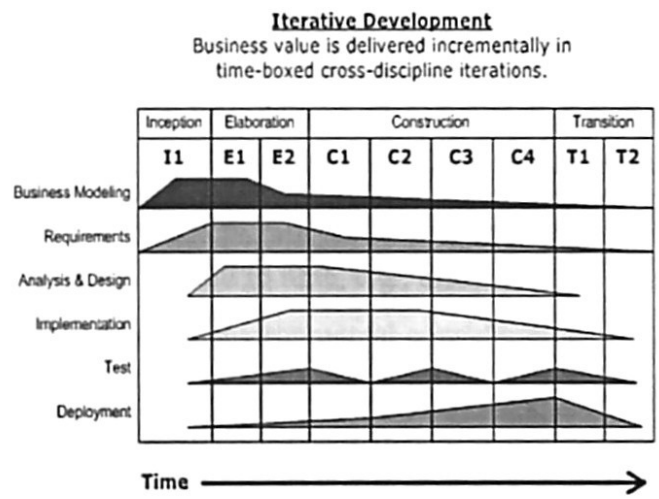

Figure 2.4 : Iterative Development [6]

Iterative development slices the deliverable business value (system functionality) into iterations. In an iteration a slice of functionality is delivered through cross-discipline work, starting from the model/requirements through to the testing/deployment. The unified process groups iterations into phases: inception, elaboration, construction, and transition. **Inception**

identifies project scope, risks, and requirements (functional and non-functional) at a high level but in enough detail that work can be estimated. **Elaboration** delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements. **Construction** incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements. **Transition** delivers the system into the production operating environment. Each of the phases may be divided into one or more iterations, which are usually time-boxed rather than feature-boxed. Architects and analysts work one iteration ahead of developers and testers to keep their work-product backlog full.

Using analysis and measurement as drivers of the enhancement process is one major difference between iterative enhancement and the current agile software development. It provides support for determining the effectiveness of the processes and the quality of product. It allows one to study, and therefore improve and tailor, the processes for the particular environment. This measurement and analysis activity can be added to existing agile development methods.

The relative changes measured over the evolution of the system can be very informative as they provide a basis for comparison, even if sometimes difficult to understand in the absolute. For example, a vector of measures, m1, m2, ... mn, can be defined to characterize various aspects of the product at some point in time, e.g., effort to date, changes, defects, logical, physical, and dynamic attributes, environmental considerations. Thus an observer can tell how product characteristics like size; complexity, coupling, and cohesion are increasing or decreasing over time. One can monitor the relative change of the various aspects of the product or can provide bounds for the measures to signal potential problems and anomalies.

Many utility software systems have been developed using this model, wherein the requirement is basically providing the customer with some working model at an early stage of the development cycle. As new features are added in, a new release is launched which has fewer bugs and more features than the previous release. Some of the typical examples of this kind of model are: Yahoo Messenger, Cyber Sitter, Net Meter, PC Security, P2P, etc.

Summary

Different approaches in design and software development used to manage risk. Top-down design emphasizes a complete design, often composed of many subsystems, before any coding begins. Bottom-up design, instead, requires early coding and testing of modules. Several software development methods were discussed. All these methods must include phases of planning and requirements, design and architecture, implementation, testing, and deployment. The waterfall model is direct and cost effective, but it is inflexible and does not gracefully handle changes to requirements after design has begun. It proceeds through the phases a single time, with no going back to previous phases. The iterative processes, in contrast, expect to revisit each phase, to incorporate changes and lessons learned. The spiral model uses several cycles that begin with requirements and design and end with evaluation of implementation during the cycles. The agile family of methods uses short iterations, or time boxes, that are each mini-development projects. They emphasize team communication and flexibility in requirements. The best-known agile method is extreme programming, which prescribes practices meant to encourage the values of communication, simplicity, feedback, courage, and respect. Rapid prototyping is a technique of quickly assembling a user interface for evaluation.

By using time boxing and ranking our outstanding tasks, we make ourselves consciously aware of how much time we have available. This allows us to focus our energies towards things that matter most. In this way, we get things that matter most done first.

In conclusion

Time boxing is an effective way for getting things done. By fixing the amount of time we spend on a given set of tasks, we can focus on doing the things that matter, give us motivation to start, prevent overruns and use our nul times effectively. In contrast, if we worked on things until completion in one sitting, we're less likely to start on complex tasks, more likely to overrun on open ended tasks and leaves us with less time and motivation to work on the next set of tasks.

Looking at the discussion so far we can state that Incremental and Iterative development (combined together and referred to as "Iterative Development Strategy") is a TIME BOXING implementation tool.

2.3.3 Implementation Guidelines

There are two major approaches of implementing the time boxing to a project.

2.3.3.1 Top-Down Time-boxing

It is used at project level by the clients who want to set a delivery date for a project of specified scope /quality /cost. In this the management takes decision of what, how, when and with what resource the product is to be produced. Top Down time boxing of projects fail due to poor staff morale. In this a project manager who decides to produce a project plan dictates project team work structure, their associated tasks /durations and effort required to complete a project without focusing on a task by task basis. Such an approach simply de motivates good project teams. This might lead to mismanagement of team hence resulting in delay of product delivery.

2.3.3.2 Bottom-up Time boxing

In this the time box is not set by the management but by the project team. This involves full participation by the project team in an iterative project planning process where every product, and its associated task breakdown, is estimated in terms of time, cost, ownership (who does what and who is responsible), effort and dependencies. All iterative estimates will detail, where applicable, team based risk /issue assessments and what assumptions had to be made in order to make the estimates possible.

Advantages of Bottom-up Time boxing

Compared to the Top Down approach mentioned earlier this approach has the following advantages:

- Promotes better staff motivation.
- Feasibility is at the heart of the Bottom-up Time-box.
- Greater control over the delivery as each Time-box is typically only 6 - 8 weeks in duration, so it is easy to understand what is going on in each product level Time-box.
- Limits scope creep and gold plating by the project team because of tight delivery deadlines.

- Puts brackets around chaos enabling people to focus sharply on meeting minimum requirements for success.

2.3.4 The Major steps involved in Time-boxing

You can create and use time boxes effectively by following the steps given below:

Step 1. Find the areas for Time-boxing.

Step 2. Identify what are your objectives.

Step 3. Identify the appropriate Time-box.

Step 4. Execute results within your Time-box.

Step 5. Evaluate and adapt.

2.3.5 Important points to be considered while Time-boxing

2.3.5.1 The Iron Triangle

On a given project, resources are usually fixed; and unless you believe in the Mongolian Horde Approach (hire a hundred people and hope some of them are good) the best team is a small one. Once you've put that team together, you've established their capability, at least in the short run. Now you have three aspects to manage, as shown below

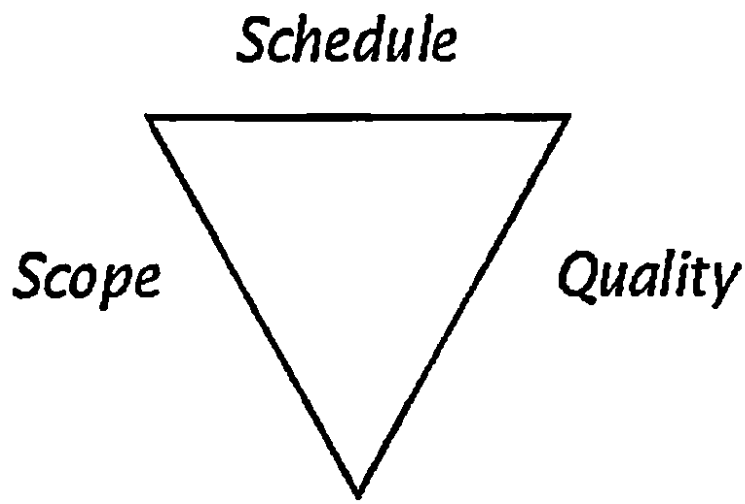

Figure 2.5 : The Iron Triangle[6]

[6] Available from: http://www.belizenorth.com/articles/TIMEBOX.htm

Schedule : The time when the software product is due.

Scope : The functions and features delivered in the software product.

Quality : The absence of defects in the product.

For example:

1. If we increase the scope, the schedule must grow, or the quality must decline.
2. If we shorten the schedule, we must decrease the scope or deliver with more defects.

The best time boxing strategy holds quality constant and reduces scope to meet a schedule.

2.3.6 Time box duration limits

Each product level time box should typically be no more than 6 to 8 weeks in duration. In case of any longer duration than this should be considered for breaking the product down into further lower level product deliverables and time box in turn. For example, a Use Case model could be one deliverable but if this turned out to be something that might take more than 8 weeks to deliver then it should be broken down into Use Case model components (key product features) each with separate product owners and independent quality checks and reviews.

2.3.7 The latest features

Those latest and greatest features cost more than you expect, and here's why. Remember the 90:90 rule :

The first 90% of a system takes 90% of the time:
The last 10% takes the other 90% of the time

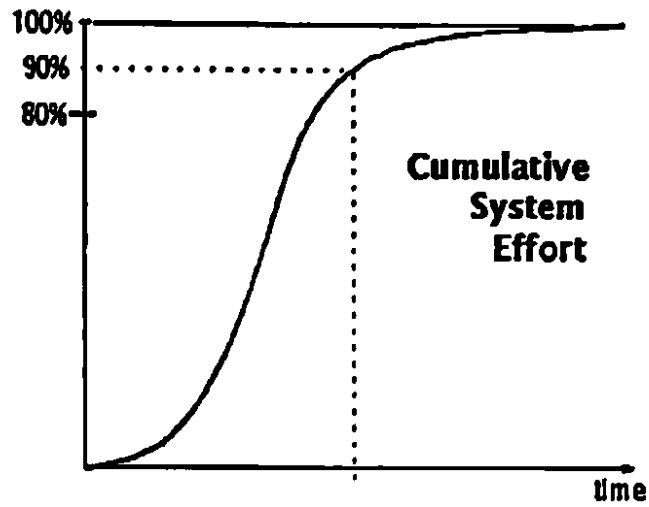

Figure 2.6 : The 90/90 Rule [6]

As we approach 100% complete, our progress slows down drastically. This is because we're making tough decisions in the face of uncertainty. Moreover, they're not very good decisions. We will probably have to make many of them over again, when we have more information. This last 10% also accounts for much of the arguing and fighting that goes on in a project. Time boxing forces us to forgo these last features, but it also lets us avoid most of the conflict that goes with them.

Pareto's Law -- the old "80:20 rule" -- gives us another justification for procrastinating on those last features. In the systems world, it predicts that:

20% of the system will give us 80% of the payback.

Now, in reality, this 20% only applies to a particular set of users. If we have a diverse set of users, we will have to give each group a different 20%, but it's reasonable to expect that we can please the vast majority of our users with 80-90% of the features. Sooner or later, we're going to deliver those last features.

2.3.8 The right features

Making Pareto's Law work for you may sound like magic, but there actually is a systematic way of finding out what features you should deliver first. Ask your customers to rank the features they want. You can do this most easily in a group meeting of customers and

developers. Write each feature on a Post-it, put these on a whiteboard, and have the group rank them (1 is high, 10 is low.). Then ask your developers to estimate how difficult each feature will be to implement (1 is easy, 10 is hard) and multiply these two to give you a priority weight for each feature. On the whiteboard, build a matrix like the one shown below. It will show you, the team and the customer features you should implement first and which you might postpone. (Quality practitioners will recognize this process as a part of QFD or "Quality Function Deployment.")

Table 2.1 : Feature Priority Matrix [6]

Feature	Customer Rank	Delivery Cost	Priority
Capture existing file	3	4	12
Create new records	1	1	1
Allocate new space interactively	5	9	45
Validate keys interactively	2	2	4
Validate all fields interactively	6	6	36
Recreate file from backup	3	4	12
Update file from journal	8	7	56
Modify existing records	1	3	3
Find record by primary key	1	2	2
Find record by secondary key	2	6	12
... etc...	...	...	...

2.3.9 Incremental Releases

To stage incremental releases of a software product the best method is to manage their features. Customer confidence can be build through a series of timely releases, which delivers a steady stream of new features. For doing this get it into a stable state, and be ready to ship at any time.

Here's a strategy for delivering that first release:

1. Define your features.

2. Prioritize them.

3. Define three subsets:

- Got to have
- Should have
- Nice to have

4. Build the 'got to have' subset as a prototype. Define a time box, start prototyping, and deliver what you have when you run out of time. (Since it's a prototype, you won't have trouble explaining why it looks incomplete.)

5. Use this early experience with the prototype to define time boxes for your first incremental release.

6. Stay within your time boxes, delivering the features you have ready, *on time.* [6]

2.3.10Maintaining Quality

If you're in a stable state, you have a much better chance of controlling quality. A couple of basic metrics will demonstrate stability and dramatically improve your ability to deliver a quality product as you reach the end of a time box. You need Defects Discovered and Defects Corrected for each time period (days or weeks.) Figure 2.7 is a graph of these two measures; you can also derive (and graph) other important measures such as Defects Remaining and Mean Time to Repair.

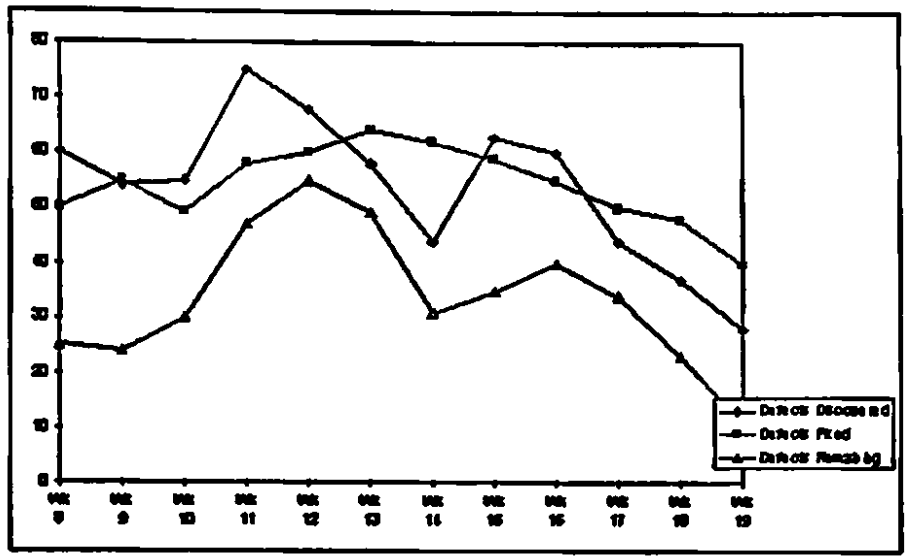

Figure 2.7 : Defects found vs. Defects fixed [6]

According to McCarthy, Microsoft has a ratio of one Quality Assurance (QA) person for every two developers on the team. This graph is a great way for QA to highlight the coordination between these two factions on your team.

You can Time-box anything

We found that you can time box any development activity, from requirements definition, to system design, to paper prototyping your screens. You define a time interval, and work within it. When you run out of time, you stop, and move on. Of course, you have to be reasonable; you can't do ten days of coding in a two-day time box; but you actually might able to build a prototype in three days. You won't know until you try.

2.3.11 Time-box Stage Planning

Plan the large projects in stages.

Plan >>> Design >>> Develop >>> Test >>> Release

During planning stage (requirements definition) which requires time boxing (bottom-up), put together all the time boxed and planned blueprint products i.e. use case model, test strategy, architectural design etc with connecting dependencies into an integrated stage plan. This usually happens not as a separate task but as part of the iterative project

planning process. Once complete we must obtain minuted agreement by the project team that the Stage Plan and its component products allowing for contingency (assigned in accordance with the projects risk, issue and assumption profile) can now be fixed in terms of the delivery dates. If the answer is YES then your stage plan is now time boxed

Controlling Slippages

Especially with projects which have a poor risk /issue and assumption profiles, time boxing often fails to get a project or stage in on time. If you feel this is likely then you must get agreement from the client as to the projects 4 key drivers and their relative order

e. g. Time > Scope> Cost >Quality

If the client /sponsor agree with these drivers and their respective order, then it is clear that you have authority to first reduce quality, then increase cost, and then start cutting out functionality in order to bring in a slippage in target delivery date and set a new Time-box.

It is very important before making these adjustments to the project plan that it is done with the consent of the project team and the sponsor /client. When informing the client /sponsor of a slippage and providing your suggested "What-if" changes to the project plan to bring in the target delivery date and set a new Time-box, it is very important that these driver tradeoffs are explained in terms of risk to the business case.

Normally all these details /procedures and definitions of quality would be explained in the Project Initiation Document (PID) which is normally signed off by the client /sponsor and governs the way the project is managed.

3 Literature Survey

3.1 Literature related to ECG

- Research based on ECG Application Featuring Data Transmission by Bluetooth (Daniel Marr, October 2001)[7] in which method has been shown to reduce the problem of heart disease in Australia. The prototype consists of reduced cables and reduced configuration. The data is stored in a computer file format and autonomous sensor communicates with a PC.
- In another application of ECG Feature Identification for Cardiologist Expert Diagnosis by Hong Liang (2005)[8] which proposed a reliable method for cardiologist expert diagnosis based on ECG elements identification. It analyses ECG key features (P wave, QRS complex, T wave) which includes noise purification, sample design for digital ECG, and the extended measurements report. This method can implement ECG report in real time and provide exact explanations for the diagnostic decision obtained. This method can offer mean standard values estimation for parameters and confident interval computing for predictive accuracy. This method solved the problems like: noise purification, incomplete and confusing ECG element's key features identification, decision-rule base, expert diagnosis model. A normal ECG makes heart failure less likely but further investigation of the patient is required in order to rule out this diagnosis. There is no consistent ECG abnormality that is specific enough to make the diagnosis of heart failure purely from this investigation.
- In the paper on Performance of exercise ECG testing by Frank G Yanowitz, MD (January 2009)[9] discusses about how the patient gender, age, coronary risk factors and the characteristics of the chest pain are also important determinants of the pretest probability of coronary heart disease and diagnostic accuracy of exercise ECG testing. There are two limitations to exercise ECG testing exclusion of subsets of patients from the test (e.g. Patients unable to exercise) and limited sensitivity and specificity compared to coronary angiography. In another paper on ECG signal de-

[7] Available from: http://innovexpo.itee.uq.edu.au/2001/projects/s369535/thesis.pdf
[8] Available from: http://www.ncbi.nlm.nih.gov/pubmed/17281069
[9] Available from: http://www.uptodate.com/patients/content/topic.do?topicKey=~d8x8vk/U6v

noising Brij N. Singh and Arvind Tiwari (January 2006) this paper presents a selection procedure of mother wavelet basis functions applied for de-noising of the ECG signal in wavelet domain while retaining the signal peaks close to their full amplitude. The obtained wavelet based de-noised ECG signals retain the necessary diagnostics information contained in the original ECG signal.

- Wavelet-Based ECG and PCG Signals Compression Technique for Mobile Telemedicine Manikandan, M.S. (Dandapat 2007. ADCOM 2007. I2007.29).[10] In this one of the emerging issues in tele-health care system is how effectively the limited and well established mobile technologies that are now almost globally usable are exploited. The main challenge is to develop a mobile telemedicine system to transmit bio-signals directly to a specialist in an emergency medical care unit for monitoring/diagnosis using an unmodified mobile telephone which provides the patient's information on the spot without unnecessary delays in seeking care, access to health facility and provision of adequate care.
- Literature related to Time boxing by Dave Cheong (July 26th, 2006)[11]

  a) Time boxes support query specification & expressiveness through direct manipulation

  b) Time Finder demonstrates the operation of this model on a small data set, while supporting data and query envelopes for displaying overview information

  c) Time box support queries aimed at finding events at arbitrary time sequences, or with given changes in attribute value.

  d) This utility will encourage improved algorithms and system design for integrating the interactive search facilities, usability of the interface

  e) Time boxes work with different data sets and application domains Possibilities include the general class of categorical data such as medical

- In Stock time series visualization based on data point importance by Tak-chung Fu, 2001. Time series visualization is a fundamental task in most financial applications.

[10] Available from: http://portal.acm.org/citation.cfm?id=1333727
[11] Available from: http://www.davecheong.com/2006/07/26/time-boxing-is-an-effective-getting-things-done-strategy/

a) Framework that can reduce the dimensionality of the time series data accurate
b) It is preferable that the representation framework can handle the multi-resolution problem rather than reduce the dimension to a fixed level only. In this paper, a framework that represents and visualizes time series data based on data point importance is proposed.

A method for discovering patterns across different resolutions is proposed. By converting the time series to symbol string based on data point importance, the potential patterns with different lengths can be encoded in the VizTree for visual pattern discovery while the important points and the overall shape of the time series patterns can be preserved even under a high compression ratio. Various experiments were conducted to evaluate the performance of the proposed framework. One may find it particularly attractive in financial applications like stock data analysis.

3.2 Application of Different approaches

Analysis of projects, approach for determination of appropriate approach for respective project. It discusses about S/W and H/W requirements, Time Box Planning, Deliverables Analysis, improve the on time deliveries of business values rather than focusing on complete deliveries at one go, which could lead to delays causing the loss of its business values.

The focus is on the EDS case study and study of the role of ECG Diagnostics in controlling the heart diseases. Considering the dependence on these proven ECG technologies there is even more need of further enhancements to reduce the delays in diagnosis and there after the required treatments for patients who are at the doorstep of a probable death. In this diagnostics one of such area is ECG Analytics. There are various software systems to take care of ECG Analytics. These systems need to respond in quick manner to the changing requirements from the doctor's community and there by deliver the business value to doctors as well as heart patients. EDS (ECG Diagnostic System) is one such system that caters to the needs of doctors.

The electrocardiogram (ECG) is the most commonly performed cardiac test. This is because the ECG is a useful screening tool for a variety of cardiac abnormalities. ECG indicates a heart attack or possible coronary artery disease. ECG machines are readily available in most medical facilities; and the test is simple to perform, risk-free and inexpensive.

ECG is performed as follows:

- The patient lies on an examination table, and 10 electrodes (or leads) are attached to the patient's legs, arms and chest.
- The electrical impulses generated by the heart are detected by electrodes which transmit them to the ECG machine.
- The ECG machine produces a graph (the ECG tracing) of those cardiac electrical impulses and then the electrodes are removed.

An electrical recording of the heart, ECG reflects the proper functioning of the heart muscles and their condition as a function of the blood supply and oxygen received by them.

3.3 Information gained from the ECG

The following information can be determined from the ECG tracing

- The heart rate
- The heart rhythm
- Whether there has been a prior heart attack
- Whether there are "conduction abnormalities"
- Whether there may be coronary artery disease
- Whether the heart muscle has become abnormally thickened

3.3.1 Fundamentals

The function of the ECG machine depends on the ability of the heart to produce electrical signals. The pumping action of the heart is regulated by the pacemaker region. Using an ECG allows doctors to measure the relative voltage of the impulses at various positions in

the heart. ECGs are possible because the body is a good conductor of electricity. When an electrical potential is generated in a section of the heart, an electrical current is conducted to the body surface in a specific area. Electrodes attached to the body in these areas enable the measurement of these currents.

3.4 Advancements in ECG

The most useful characteristics of the ECG is its ability to detect and describe arrhythmias (abnormal heartbeats)

3.4.1 Portability

The machine is portable since the size of the electromagnet is reduced.

3.4.2 Convenience

The electrodes were developed such that it could be attached directly to the skin. The original electrodes required the patient to submerge the arms and legs into glass electrode jars containing large volumes of sodium chloride solution.

3.4.3 Digitization

The incorporation of amplifiers, which improved the electronic signal, and direct writing instruments, which made the ECG data immediately available. The modern ECG machine is similar to these early models, but microelectronics and computer interfaces have been incorporated, making them more useful and powerful. While these newer machines are more convenient to use, they are not as accurate as the original ECG.

3.5 Categorizing ECG machines

ECGs can be categorized as Digital and Analog. Analog machine can detect the continuously variable electrical activity in the heart, amplify and record it while in progress. Digital ones convert the information into digits or numbers first, store them in a computer, analyze them and then retrieve and reproduce them.

Like all equipment, most ECG machines are digital, but there are many analog machines also exists. Cost is a prime factor. There are still analog machines in smaller clinics where ECG is not a prime requirement in their medical diagnostics. Even very old machines, which are properly maintained, can produce satisfactory tracings, but maintenance becomes a concern. But predictably, digitization is an eventuality. ECG comes in multiple forms such as single/multi-channel units and those with or without basic interpretation. Some ECG units come with interfaces for PCs.

3.5.1 Types of ECGs

- **Resting ECG:** The patient lies down for a few minutes while a record is made. In this type of ECG, disks are attached to the patient's arms and legs as well as to the chest.
- **Exercise ECG (Stress test):** The patient exercises either on a treadmill machine or bicycle while connected to the ECG machine. This test tells whether exercise causes arrhythmias (abnormalities) or makes them worse or whether there is evidence of inadequate blood flow to the heart muscle.
- **24-hour ECG (Holter) monitoring:** This is a sophisticated type of ECG. The patient goes about his or her usual daily activities while wearing a small, portable tape recorder that connects to the disks on the patient's chest. Over time, this test shows changes in rhythm (or ischemia) that may not be detected during a resting or exercise ECG.

The most commonly used ECG is the Resting ECG. For less frequent or hard to predict arrhythmias, loop recorders are used. These are worn by the patient for weeks to even as long as a month. When the patient feels an uncomfortable heart rhythm start up, he presses a button which records his / her ECG for a fixed duration. Electrocardiogram (ECG) supplies are used when monitoring a patient for a resting or diagnostic ECG analysis, stress testing and exercise monitoring, ambulatory monitoring (such as the telemetry and Holter monitoring) and cardio-monitoring and other imaging procedures.

3.5.2 The limitations of the ECG

- The ECG reveals the heart rate and rhythm only during the time that the ECG is taken. If intermittent cardiac rhythm abnormalities are present, the ECG is likely to miss them. Ambulatory monitoring is needed to record transient arrhythmias.
- The ECG can often be normal or nearly normal in patients with undiagnosed coronary artery disease or other forms of heart disease (false negative results.)
- Many "abnormalities" that appear on the ECG turn out to have no medical significance after a thorough evaluation is done (false positive results)

4 Case Studies

4.1 Case Study 1: EDS (ECG Diagnostic System)

4.1.1 Embedded Systems

Typically an Information System is the combined creation of a diverse group of people working under a software project. Further these information systems can be classified as Desktop Computing Systems and Embedded Systems. The Desktop computing systems are purely the Software systems where as the Embedded systems are integrations of Software as well as Electronic Devices those are interfaced with Desk Tops for Real time data acquisition & processing.

There is considerable difference in characteristics of these two types of systems. Many embedded systems have substantially different design constraints than desktop computing applications. No single characterization applies to the diverse spectrum of embedded systems. There is always an uncertainty in successful completion of development of embedded part of the entire system, due to varied applications, environmental and performance characteristics. Currently there is very little tool support available for expanding embedded computer design to the scope of holistic embedded system design. This adds up the uncertainty of embedded system development. Further some combination of cost pressure, long life-cycle, real-time requirements, reliability requirements, and design culture dysfunction can make it difficult to be successful applying traditional computer / software design methodologies and tools to embedded applications.

The typical stand alone Embedded System is shown below

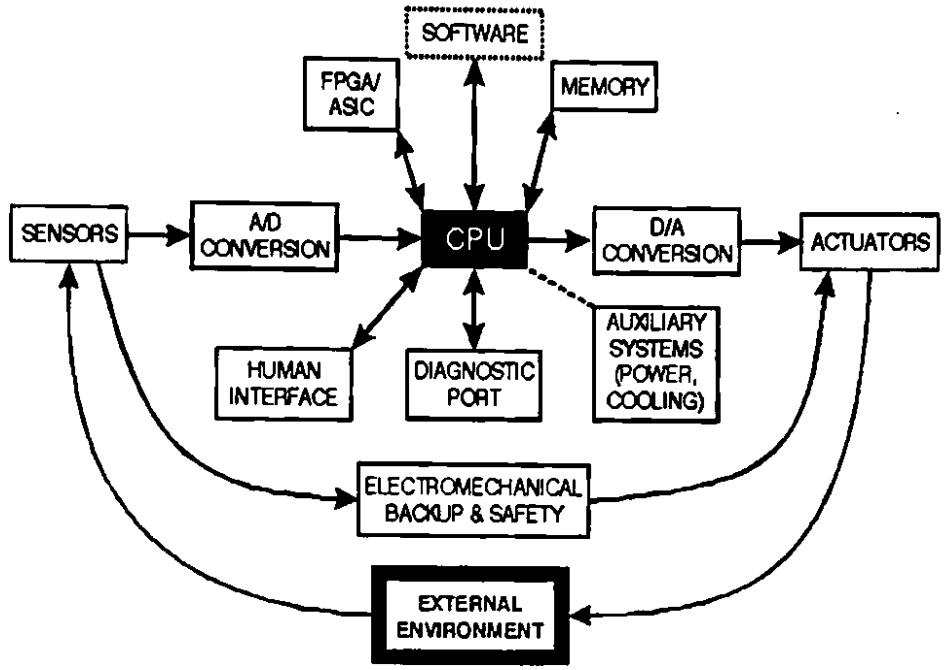

Figure 4.1 : Stand alone Embedded System

The design considerations stated above make great impact on embedded project management and timely delivery of the projects. However, knowing the strengths and weaknesses of available approaches can set expectations appropriately, identify risk areas and suggest ways in which the time bound projects can be delivered.

A time-bound project is a project that is constrained by hard deadlines. Hard deadlines are those in which the date of delivery is as important as the delivery itself. If the project delivers after the deadline, the delivery loses much of its value. Examples of hard deadlines are exhibition dates, government regulations, a competitor's announcement and the customer's own business plans.

Most of these projects start with more requirements that can realistically be handled within the imposed time constraints and consequently, midway through the development, they find necessary to start slashing some of them. These un-planned cuts result in customer frustration and wasted effort. A much better approach would be to define the

requirements' priority up-front, allocating their development to successive releases of the project in such way, that we could be almost sure that the project will deliver all the important requirements, that the second less important will still have a fair chance of being delivered, with the gold plated ones only to be done if there is any time left.

The lack of requirements prioritization is not the only reason for most of the project delays. The inability of traditional planning methods to deal with the uncertainty present on the estimates on which the plans are based, and the failure to recognize that development work do not progress in linear fashion, the infamous 90% complete syndrome, are also to blame. Critical path calculations involving uncertainty produce considerably shorter schedules than those that should be realistically expected. With a shorter schedule as starting point, being late is a tautology.

The second problem, assuming that a task progresses at a constant rate, prevents project managers from seeing the early signs of delay in tasks until it is too late to take any other action than trim down features, compromise on quality or re-schedule the project.

To overcome these problems and achieve timely deliveries of such complex systems the experts have suggested The Incremental and iterative development which is nothing but a Time Boxing technique tool. This method may not be the perfect solutions since it could have some drawbacks in itself. However we can consider the best optimized use of these approaches individually or in combination with some traditional approaches for successful delivery of the projects that is optimized for input requirements, outputs and timely delivery.

In the following sections we are going to study the Iterative Development approaches, its effectiveness and feasibility to apply them in Development of a typical Embedded Biomedical Systems taken as example case.

4.1.2 EDS Overview

Advanced EDS system is an embedded system that comprises of Embedded Electronics and Embedded Software units which are:

1. PC ECG -12 - ECG Acquisition unit
2. EDS Software – ECG Diagnostic System software

4.1.2.1 Embedded Electronic Device - PC ECG -12

This is a micro-controller based ECG device. It is used for ECG acquisition, amplification & transmission of the ECG data to computer in digital form for further use by EDS Software application.

This device is intended to use for capturing the patient's EGC signals through 10 ECG leads connected to patient with standard placements. These signals are filtered, amplified, measured & converted into digital data format. This device outputs the ECG data for 12 standard leads in 12 bit digital value formats. The digital data is transmitted to PC via USB port as input for the compatible PC software application, that clinically analyses, displays & gives output to other devices as required.

This device does not require (mains generated primary power / battery etc) power source. It uses the extra low (5 VDC) secondary power available from standard USB port. The power available from USB is isolated & used as supply source for patient side operation of this device. The data communication section of this device still continues to use the same un-isolated USB power.

Figure 4.2 : Embedded Electronic Device - PC-ECG 12

4.1.2.2 EDS Software

This is a biomedical information system. It consist the electronic database for diagnostic records providing paperless efficiency and connectivity and when combined with other diagnostic modules like Holter, Stress Test provides a complete solution to all diagnostic tests for each patient.

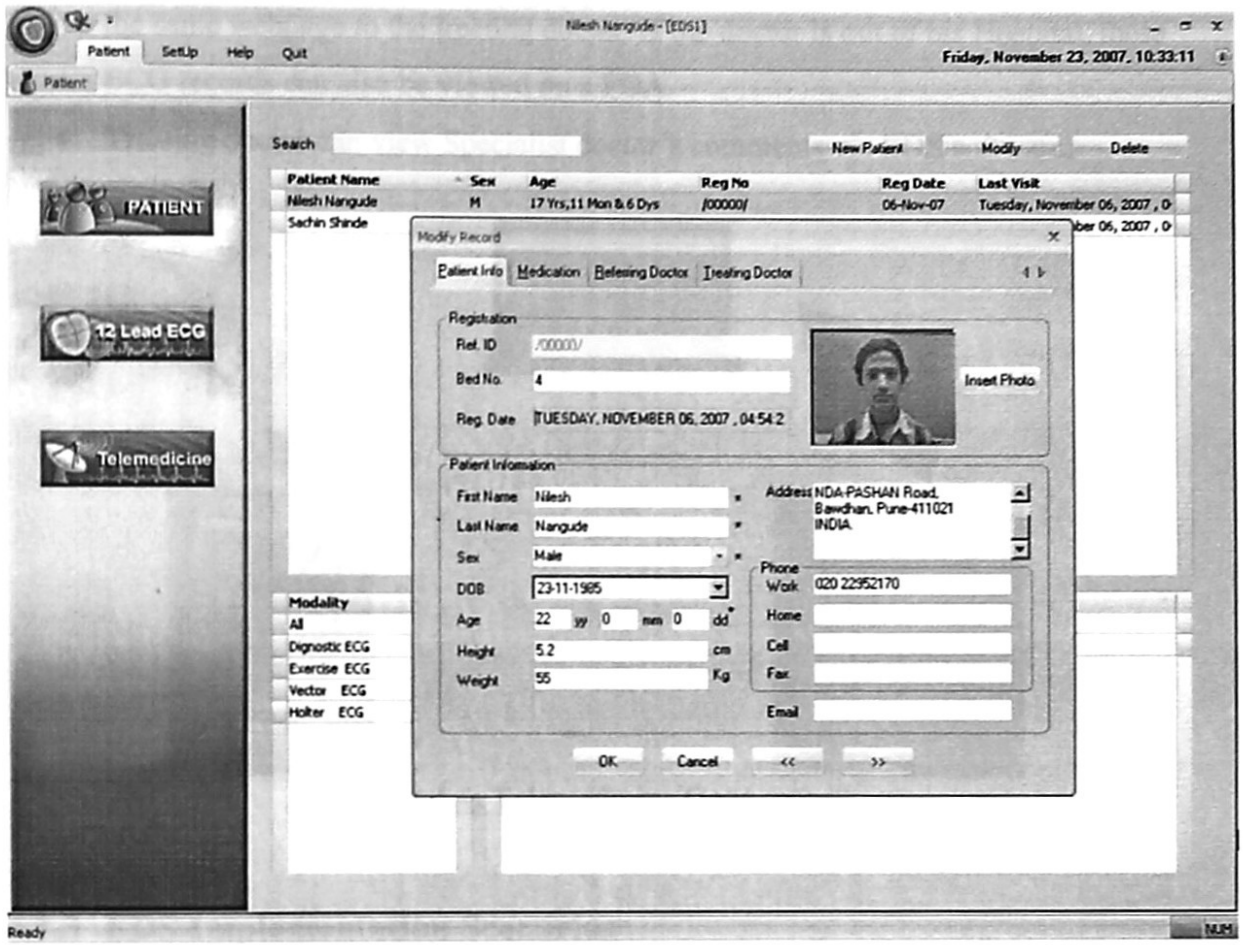

Figure 4.3 : EDS Sample Screen

It displays real-time ECG data and waveforms on-screen, instantly performs a full set of measurements and offers interpretation. It provides for on line as well as offline reports printing facility.

Telemedicine is one of the advanced features incorporated in this system that uses internet communication.

Telemedicine

- EDS software has telemedicine capability.
- It includes cohesive operation by treating and Specialist Doctor.
- In this the treating doctor with EDS software can directly upload patient ECG records on the website or a common website for group of DOCTORS or hospitals.
- Through the common website, specialist doctor can view the ECG records by logging onto the website and then asses the condition and give his comments on

the same webpage to the treating doctor. If the doctor is out of his Hospital, these ECG records can also be viewed on a PDA.

- Treating doctor can view Specialist doctor's comments via EDS software.

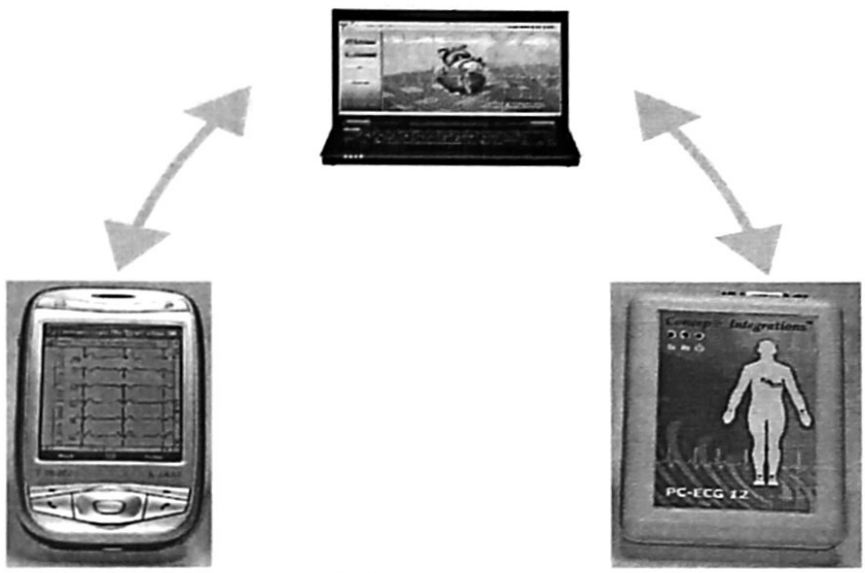

Figure 4.4: Telemedicine Connectivity

4.1.3 EDS Implementation Scenarios

This product system with PC ECG-12 which includes EDS software for the analysis and display and it also has the Telemedicine features.

PC ECG -12 will be available in two variants:

1. PC ECG-12
2. PC ECG-12 with Telemedicine

4.1.3.1 Scenario 1

If a Specialist cardiologist who has good number (minimum 10) of associated Doctor then we can provide a complete system for them.

In this, Specialist Cardiologist and all other associated Practitioner will have PC ECG -12 with telemedicine.

Each doctor will have a unique login id to a common website. On the website there will be two types of login screens:

1. for Specialist Doctor
2. for Treating Doctor

To send a file for analysis to Specialist Doctor, Treating doctor has to go via EDS software to Telemedicine and from there with upload button treating doctor will be able to upload it. After Doctor clicks on Uploaded Successfully, it will automatically direct doctor to the webpage, where doctor will login and then select the Specialist Doctor and will write his comments and then send it to Specialist Doctor.

Specialist doctor will login to the website and will look for new case. After that Specialist Doctor will check the ECG record and will add his comments and send it back to treating Doctor.

Treating doctor will go to "View Comments" on EDS software, which will direct him to website, where after login user will be able to view comments.

In the same way other treating doctor can also send ECG report to specialist doctor.

Comprehensive solution for this system includes:

1. PC ECG-12 units to all Doctors
2. EDS software to upload ECG reports
3. A common platform, website for particular group of doctors or they can use different websites for uploading ECG records.
4. Specialist Doctor can view on Computer.

In this if Specialist doctor can arrange 10 other treating doctors, and then Specialist doctor can be provided with a free of cost PDA.

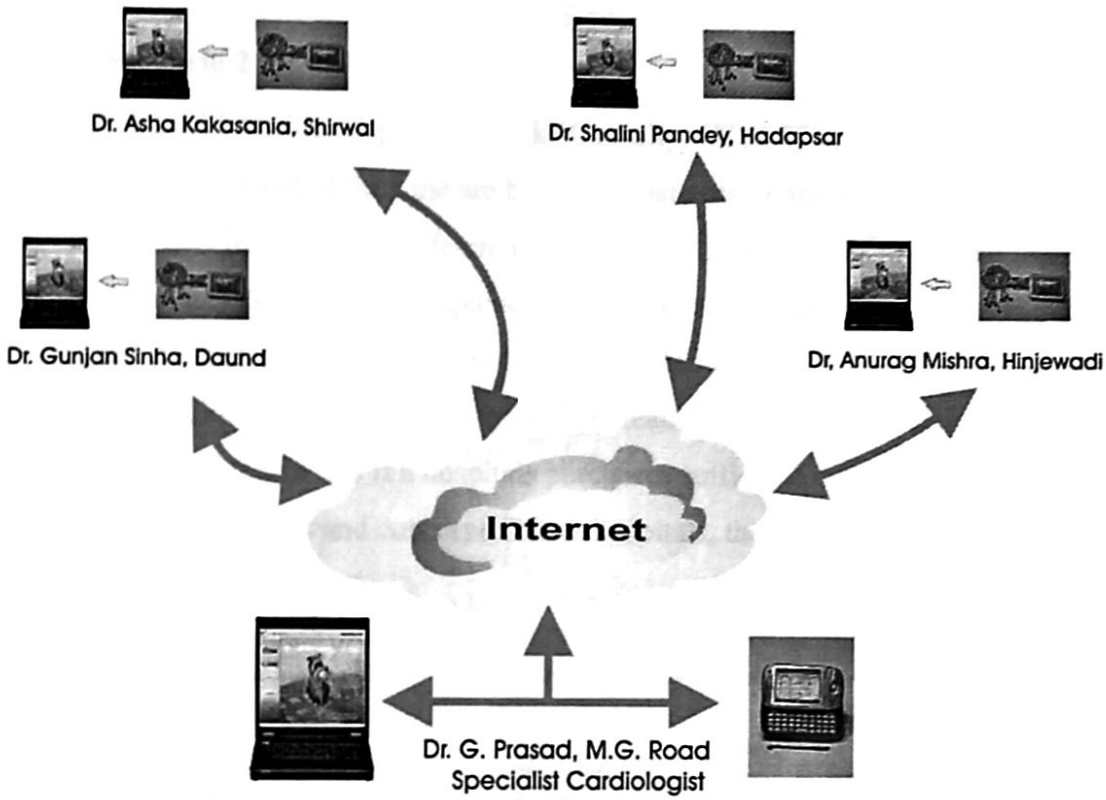

Figure 4.5: Diagrammatic Representation of Scenario-1

4.1.3.2 Scenario 2

In a Hospitals with various departments like Casualty, OT, OPD etc. and with Consulting cardiologist which most of the time are busy with patients or are on move from one Hospital to other. In a hospital different department will have PC ECG-12 and with the EDS software they will be able to upload the ECG records to the hospital website or they can also use Telemedicine specific website. The consulting doctor can view ECG records on website or PDA and sent his comments. The treating doctor can view these comments via the EDS software. In this if a hospitals purchases sufficient number of PC- ECG 12 units for it's departments and concerned sister hospitals, then they will be provided with a free of cost PDA

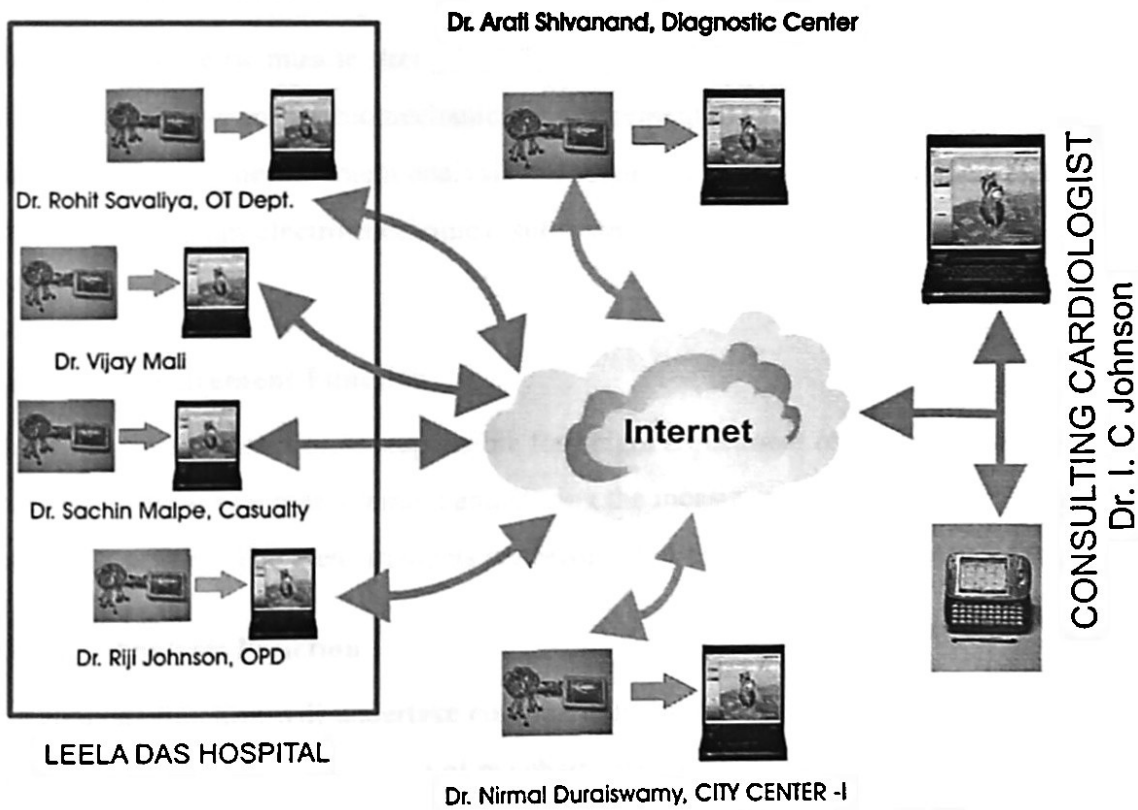

Figure 4.6: Diagrammatic Representation of Scenario-2

4.2 Case Study 2: EMG (Electro MyoGraph) System

This will describe the design concepts and architecture for S/W subsystem for EMG isometric muscle strength analysis system.

EMG consists of following functionalities:

- Measurements
- Analysis
- Configuration

4.2.1 EMG Overview

EMG is an isometric muscle strength analysis system. It consists of an electro-mechanical subsystem for biomechanical measurements and PC-based S/W subsystem for user interface, measurement analysis and reports. The biomechanical measurements are transferred from electro-mechanical subsystem to PC-based S/W subsystem through RS-232 interface.

4.2.1.1 Measurement Function

The measurement function is responsible for height adjustment of the measurement chair, conducting the actual measurement and storing the measurement data for further usage by analysis function. The measurements are recorded in the form of force versus time.

4.2.1.2 Analysis Function

The analysis function will undertake computations based on the recorded measurements and present the reports in the form of pie chart, graphs and tables. The reports have to be printed out on the printer. The analysis results and reports have to be stored for the future reference as well as for comparison with previous / future tests.

4.2.1.3 Configuration Function

The configuration function registers system settings and user's options and preferences. The configuration data is stored and provided to other functions for controlling the operation of the system. The configuration function consists of about 5 sets (screens) of settings and options.

4.2.2 Design considerations

- For each user control there will be a separate form.
- S/W should be such that it becomes easy to localize for different countries.
- Use of string table will serve the purpose of language independence.
- After completion of measurements some indication should e given to the main application that the measurements are done and saved.
- Display analysis result on pie-chart or graph.
- At the moment for the EMG professional six different measurements are possible.

4.2.3 System Architecture

There are different user controls for which different forms are to be implemented

There are total 5 user controls as shown in the block diagram:

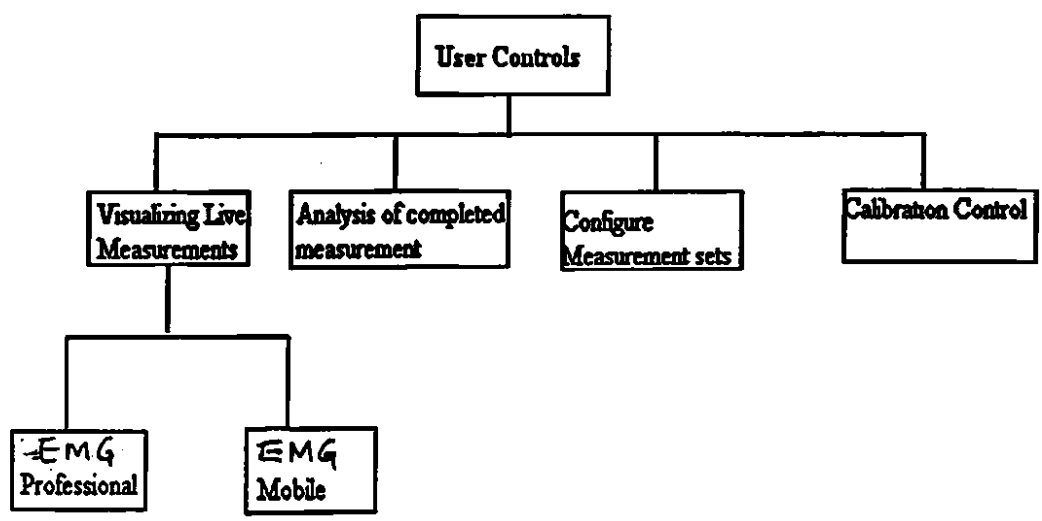

Figure 4.7: EMG User Controls

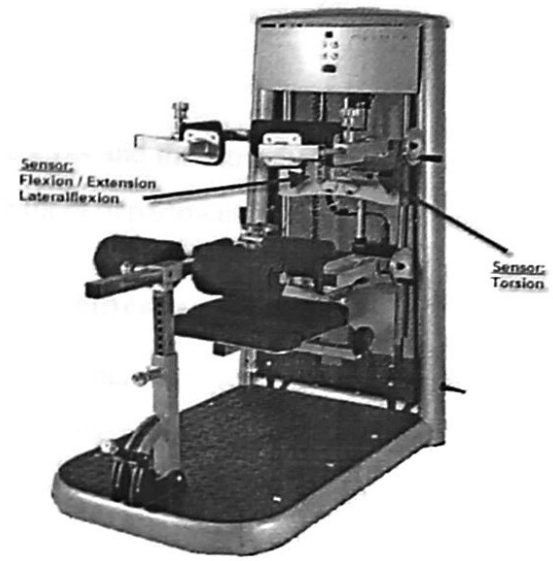

Figure 4.8 : Software EMG professional

5 Requirement Analysis

5.1 EDS Requirements

The requirements of the EDS and the attributes of each of the requirements as specified by the Business development department (Internal Customer) are listed below,

5.1.1 Software Requirements

In order to develop software there is requirement gathering which consists of Information requirement, storage requirement, protocol requirement, portable storage requirement, printing requirement etc. A detailed study is required to gather all these requirements. The following table illustrates the general requirements gathered for the case study in hand.

5.1.1.1 General Requirements

Table 5.1 : EDS General Requirement Table

No.	Requirements	Descriptions
01	Information requirement : Patient Information	Facility to enter/store patient information. a. Name, Birthdate, sex, b. Weight c. Height d. Address (single line and multiline, city, pin, country), e. Phone Number (cell Number, Home phone Number, Working Office Number) f. Doctors Name g. Doctors Email ID h. Qualification, i. Doctors Address, j. Doctors Phone Number (cell Number, Home phone

No.	Requirements	Descriptions
		Number, Working Office Number, FAX number) k. Email ID l. Reference ID m. Date & Time (When registered/test conducted) n. Medications (Pre defined list), o. Patient history, p. Clinical Information q. Relatives Cell/Phone No.
02	Data Storage Requirement : Data Storage Protocols	Facility to store patient data (Date & Patient wise) Protocols DICOM,HL7,HIPPA,TWIN compatible
03	Data Display	Facility to display wave form & parameters in following formats, a. Online b. Offline/review c. Zoom d. Annotations and Marker facility (user friendly and localized i.e. On toolbar) e. Color-coded display for different abnormalities.
04	LAN Connectivity & Telemedicine	Patient data should be available/ operable/review online on any computer within the LAN. System should have option of networking and transmission of data and final report
05	Printing Requirement	a. Grid selection facility. b. Print preview facility. c. Color selection facility. d. Report customization (add or remove parameters)

No.	Requirements	Descriptions
		e. System should have facility for i. editing ii. reviewing of final reports
06	Portable Data : Data on CD	Facility to store patient data on CD as and when required.
07	Tele medicine	Facility to send patient data across the net for consultancy. **Patient data should be available online on the net.
08	Security Requirement : Patient data security	Using security techniques for patient data safety.
09	Peripheral Requirements : Capture System/ PC Capabilities	Software should have capacity to capture system capabilities a. Monitor size b. Hard disk size, c. USB ports, d. Printer, e. Sound card Depending on all these information software should automatically configure for different modes /hardware.
10	User Interface; Gain Selection Facility	System should have facility to select the gain of each waveform. (5,10,20,40 mm/mv)
11	Key board interface for all operations	All operations in software should be possible by meaningful keyboard interface.
12	Warning Indication : Alarm facility	System should have audio and visual alarms. (in the system e.g. patient heart rate out of range, Lead fail)
13	Ports : Port selection facility	Facilities to select comport for data capture.

No.	Requirements	Descriptions
14	Doctors Conclusion: Impressions	System should have facility to provide physicians impression
15	System Upgrade: Automatic Upgrade	System should be upgraded at customer's end automatically by website when there is a Version change in S/w.
16	User Interface: Touch Screen	All software should have facility to operate by touch screen.
17	Voice Prompt	System should have voice prompts (tread mill instructions like speed increase/ decrease/ stopping treadmill etc)
18	Analysis Output: Vector Scale	Scaling vector cardiogram with the given scale say 1mv, 2mv, etc.
19	GUI: Vector Animation	Show vector cardio graphic display of one or all the three planes in a slow motion.
20	GUI: Vector Direction	System should show direction of the vector in GUI.

5.1.1.2 DECG (Diagnostic ECG) Requirements

DECG: Since it is a S/W designed for which ECG interface there are special requirements pertaining to ECG such as waveform display and related parameters etc. The requirement can be categorized as Resting ECG where the patient lies down for a few minutes while a record is made, Exercise (Stress) Test where the patient exercises either on a treadmill machine or bicycle while connected to the ECG machine and 24-hour Holter system requirements in this the patient goes about his or her usual daily activities while wearing a small, portable tape recorder that connects to the disks on the patient's chest. These requirements have been gathered in the following table summarized.

Table 5.2 : Diagnostic ECG Requirements

No.	Requirements	Descriptions
01	Waveforms display tiles	System should display all waveforms in following tiles, a. 12*1 b. 12*1+ Median c. 6*2 d. 6*2+ Median e. Rhythm Leads + Medians
02	Data storage/ retrieval	System should store and retrieve data for 10, 20, 30 and 40 seconds. System should provide the facility to store the ECG data for full test and it should be available for review as a full disclosure for the doctor as and when required.
03	GUI : Gain Selection	System should have facility to select the gain. (amplitude 5, 10, 20, 40 mm/mv)
04	Parameters	System should display all parameters on following tiles, a. Tile1 –HR/BP b. Tile2 -ST bar graph c. Tile3 – Axis d. Tile4 –ST map e. Tile5 – Amplitude chronograph f. Tile6 - Parameter table(All durations for all leads) g. Tile7-Durations (PD,QRS,PR,QT,QTC,VAT for selected lead.) h. ST trend Tile i. HR trend Tile j. BP trend Tile k. Abnormality list Tile l. Dynamic medians m. Median Comparison
05	ECG	List of Arrhythmia

	Analysis	a. Asystole b. V Tachy c. VE Run d. Bigeminy e. Trigeminy f. Wide QRS Complex Present. g. Atrial Ectopics Present. h. Pause. i. Ir-regular Rhytyhm Beats Present. j. Bradycardia. k. Tachycardia. Reanalysis after change in settings.
06	Port selection facility	Facilities to select port for data capture.(Com1 - COM255)

5.1.1.3 EECG (Exercise ECG) Requirements

Table 5.3 : Exercise ECG Requirements

No.	Requirements	Descriptions
01	Rhythm leads selection	System should provide rhythm lead selection facility (3, 4 or 6)
02	Protocols selection	System should provide the facility to select the protocol. User selectable should be up to 10.
03	Expanded medians	System should provide facility for expanded medians. Dynamic medians
04	Treadmill interface	System should provide treadmill interface through software only.
05	Report Generation	a. Phase wise and stage wise Linked median report b. HR/BP c. Median Summary d. ST bar graph e. ST trend

		f. Baseline comparison g. Analysis report h. ST max report i. Summary report System should provide the facility for previous selection memory. Check boxes for selection instead of dropdown boxes.
06	Termination criteria selection	System should provide facility to select termination criteria of the test.
07	Tiles	System should display all parameters on following tiles, 1. Tile1 –HR/BP 2. Tile2 -ST bar graph 3. Tile3 – Axis 4. Tile4 –ST map 5. Tile5 – Amplitude chronograph 6. Tile6- Parameter table(All durations for all leads) 7. Tile7-Durations (PD,QRS,PR,QT,QTC,VAT for selected lead.) 8. ST trend Tile 9. HR trend Tile 10. BP trend Tile 11. Abnormality list Tile 12. Dynamic medians 13. Median Comparison 14. Protocol display tile

5.1.1.4 Ambulatory ECG (Holter) Requirements

Table 5.4 : Ambulatory ECG Requirements

No.	Requirements	Descriptions
01	Data storage	System should provide the facility for storing the data for

		24/48/72/96 hours.
02	Reports Of Holter System	a. Summary of abnormality b. Patient Information c. Heart Rate Trend d. Arrhythmia Occurrences e. Heart Rate variability report f. ST Trends g. Maximum Heart Rate ECG report h. Minimum Heart Rate ECG report i. RR Histogram j. Reports in pdf, word and excel formats. k. Full Disclosure with selectable channels l. Full Disclosure with selectable Resolution (10min, 30min, 1Hr data on a page) m. Full disclosure for selectable time duration
03	Online viewing of ECG	System should have facility to view online ECG before/during patient is set for recording.
04	Full Disclosure	System should provide full disclosure of ECG for 24 hours. a. It should have different resolution. b. Selection facility for channels like 1,2, 3 or 12. c. Color codes for different abnormalities.
05	Holter analysis of ECG data from any other system	System should have facility to analyze the ECG Data independently as Holter analysis from a. Complete Care b. Cardio-ST **Known data formats (TODO)

5.1.2 Hardware Requirements

Certain hardware is required for real time signal capturing and data processing like A to D conversion, filters, patient cable, power supply etc.

5.1.2.1 General Requirements

Table 5.5 : EDS General Requirements

No.	Requirements	Descriptions
01	Sampling rate	System should provide sampling rate of 500 samples per channels.
02	A to D conversion speed	System should provide 10000 conversions per seconds.
03	ADC resolution	System should provide ADC resolution >= 12 bits.
04	Filters	System should provide low and high pass filters (0.5 to 150 Hzs) along with notch filter (50 Hzs) in software.
05	Pacemaker Detection	System should provide pacemaker detection.
06	Lead fail Information	System should provide lead fail information to PC.
07	Audio indication	Indication for lead fail or connection problem
08	Defib Protection	Defib protection should be provided in the system.
09	Communication Ports	System should provide connectivity through USB/BLUETOOTH/ZIGBEE/IR.
10	Physical Dimensions	System should be Compact in size. System should light weight. (Less than 300 Gms)
11	Isolation	System should provide patient isolation as per the IEC requirements.
12	Safety standards	System should comply for applicable IEC standards.

5.1.2.2 ECG Amplifier Requirements

Table 5.6 : ECG Amplifier Requirements

No.	Requirements	Descriptions
01	ECG acquisition	System should provide the facility to acquire 12 lead simultaneous ECG. System should have provision for orthogonal lead capture facility for VCG.
02	Number of channels	System should provide options for selecting 1/3/6 and 12 channels for ECG acquisition.
03	Patient cable	3,5,7 and 10 lead
04	Power Supply	Power for system should be derived from PC bus.
05	Defib Protection	Defib protection should be provided in the system.
06	Rhythm Leads	3 or 4?

5.1.2.3 Ambulatory ECG (Holter) Requirement

Table 5.7 : Ambulatory ECG Requirements

No.	Requirements	Descriptions
01	ECG acquisition	System should provide the facility to acquire 9(Lead I, II, III, aVL, aVR, aVF, C1, C2, C3) or 12(Lead I, II, III, aVL, aVR, aVF, C1, C2, C3, C4, C5, C6) leads.
02	Number of channels	System should provide options for selecting 3 channels with facility to select any 3 leads at a given time.
03	Patient cable	7/10 leads.
04	Power Supply	Power for system should be derived from PC bus and 1.5volts (AA size) battery.
05	Defib Protection	Defib protection should be provided in the system.
06	ECG acquisition	System should provide the facility to acquire 9(Lead I, II, III, aVL, aVR, aVF, C1, C2, C3) or

		12(Lead I, II, III, aVL, aVR, aVF, C1, C2, C3, C4, C5, C6) leads.

The new system should be downward compatible for interfacing the existing medical devices those are already in the market, namely CMS, CST and PCECG. Apart from these specified technical requirements there are some inherent quality and safety requirements imposed by quality management system and Medical Device Directive in view of Quality and Patient safety. Following are the typical requirements concerned with biomedical applications.

The system being developed is medical / clinical support system so it is directly concerned with the patient safety. Therefore Risk Management and Clinical evaluation are the major requirements of this project

On top of all these requirements there is one requirement that one cannot compromise is "On time Delivery". The system is being offered in competitive market competed by renowned manufacturers like Philips, L&T, GE etc. Hence to compete with such competitors the system demanded for use of advanced technologies, highest quality and timely delivery. In such scenario the time allowed for development and delivery was constrained to not to exceed 12 months for DECG and further 6 months for EECG applications.

As we have seen above, the system has two major distinct parts as hardware and software. The system also demands for varied in built functionalities and interfacing compatibilities with advanced IT systems such as telemedicine applications.

With all these considerations it was a real challenge to deliver the multifunctional quality product in time.

5.1.3 Implementation Analysis and Planning

5.1.3.1 The Problem background

Till now the project management in was based on conventional Waterfall development model. With this proven approach the quality and features of the system being developed were not compromised but the deliveries were not guaranteed in time. Some typical complex and R&D projects were getting delayed in the range of 25 – 50% overruns.

As a cumulative effect the overruns were forcing the feature cuttings, resulting in incomplete projects or incomplete deliveries, which in turn resulted in customer dis-satisfaction.

Another effect of inadequate time availability for clinical evaluation, due to overruns, caused the reworks or redesigns to great extent.

Therefore it was decided to find out the appropriate development solution so that the product can be delivered effectively in time without losing its value, features, quality and safety.

5.1.3.2 Feasibility

We have gone through different development approach the basics and their advantages and disadvantages. There we found that Incremental and iterative development through time boxing method is the most effective approach. Hence we focus on application of Time boxing for this project and check for the feasibility of application for development of EDS project.

A project to be a right candidate for Time boxing should possess following characteristics.

1. Should have appropriate development life cycle.

2. The requirements grouping and prioritization should be possible so as to plan the appropriate iteration.
3. The requirements / functionalities should have inherent incremental development characteristics in view of its business values.
4. The system should be flexible enough for the iteration / incremental development and reintegration.
5. Independent testing of particular feature / module / functionality shall be possible either the module as a whole or by the help of testing tools.

5.1.3.3 Planning

As suggested in earlier section project planning by incremental development would give more accurate results in scientific way for the large sized projects. But since we are experimenting the new techniques for the first time and in a small to medium sized project, we may not go into scientific methods. Rather we would try to develop a very simple methodology and try to get optimum results, leaving the room for further improvement in future. A real incremental approach isn't it?

5.1.3.4 Time Box Planning

With the view of Time Boxing the very first criteria was to divide the time available in appropriate sizes preferably of equal sizes. Hence the entire 18 months time available was divided in to three equal sized time boxes of 4 months. Usually smaller the time box, more it is effective. But since it is a R&D project and being an embedded system there are interdependencies for the modules to function, the time box selected is of 4 months leaving the buffer of 1 to 1.5 months buffer between each iteration for the clinical evaluation of the deliverables.

Now the next criterion is to get maximum output within the available time box so prioritization was done on the basis of customer priority and inheritability of the deliverables.

5.1.3.5 Prioritization of tasks

Since on time delivery of business values is the major requirement we calculate the priority by simple calculation

$$\text{Implementation Priority} = \frac{\text{Customer priority} \times \text{On time completion probability}}{\text{---------------------}}$$

Eqn 1

The customer priority is the actual priority specified by the customer (In our case Business Development department is the customer) on scale of 1 to 10

On time completion probability is the percentage of confidence of completion of the given task, estimated by the project manager with the help of assumptions like past experience, Complexity of the requirement, R&D requirements, effect of external factors like vendors, subcontractors, statutory and regulatory issues, readiness for adopting new techniques, manpower availability and other factors that affect the confidence level.

Once the requirements are prioritized as above, the requirements are further classified into three categories as

a. Got to Have
b. Should have
c. Nice to have

'Got to have' requirements cater with prototyping and initial acceptance by the customer. This is the foundation for further development that contains bare minimum requirements for the delivery of any project. Hence carry highest priority for delivery in initial deliveries.

Should have requirements are core requirements of the project, that the project cannot be completed until these requirements are fulfilled. These requirements have less priority in initial delivery as compared to got to have requirements but the priority goes on increasing as the project reaches near completion.

Nice to have features are the future provisions and value addition requirements, hence carry less importance in preliminary or core deliveries. These requirements can be delivered even after the core project has been delivered.

Thus we have got another parameter for prioritizing the requirements further in view of development process.

(Please Refer Appendix A: Priority Matrix for detailed priorities)

5.1.3.6 Iteration / Increment Planning

The iteration plans are prepared on the basis of Priorities determined in the Priority table.

(Please Refer Appendix B: Project plans for the detailed plans)

As discussed earlier the Business value delivery was the major criterion for iteration and time box planning.

In the general practice we know that business value of any particular activity is directly proportional to the deviation from expected delivery time. Because earlier the delivery lesser the development costs, lesser marketing efforts, more time available for stabilization, more business opportunities due to early entry in the competitive market. On the contrary more the delays more development costs, no time for stabilization, late entry causing the loss of business. From the customer point of view the priority set by him is actually a value priority of his requirement. As we go on adding the features of his priority requirements the value of deliverables increases with direct addition.

Thus if the delivery happens at expected time then value of that delivery is maximum. i.e.1. If delivery is earlier then there is no change in value since 1 is maximum value, but if it is delayed then priority comes into picture reducing the value by its product with delay unit. So we can state that

Business Value of individual deliverable

= 1 – (Priority index x Delay unit)

Where

PI = specified Priority / Max Priority number on the scale specified

---------- *Eqn 2.*

And therefore we have planned the time box that will give maximum number of required functionalities for which we are sure of On time delivery (On time probability).

Method used for determining the iteration components is as further

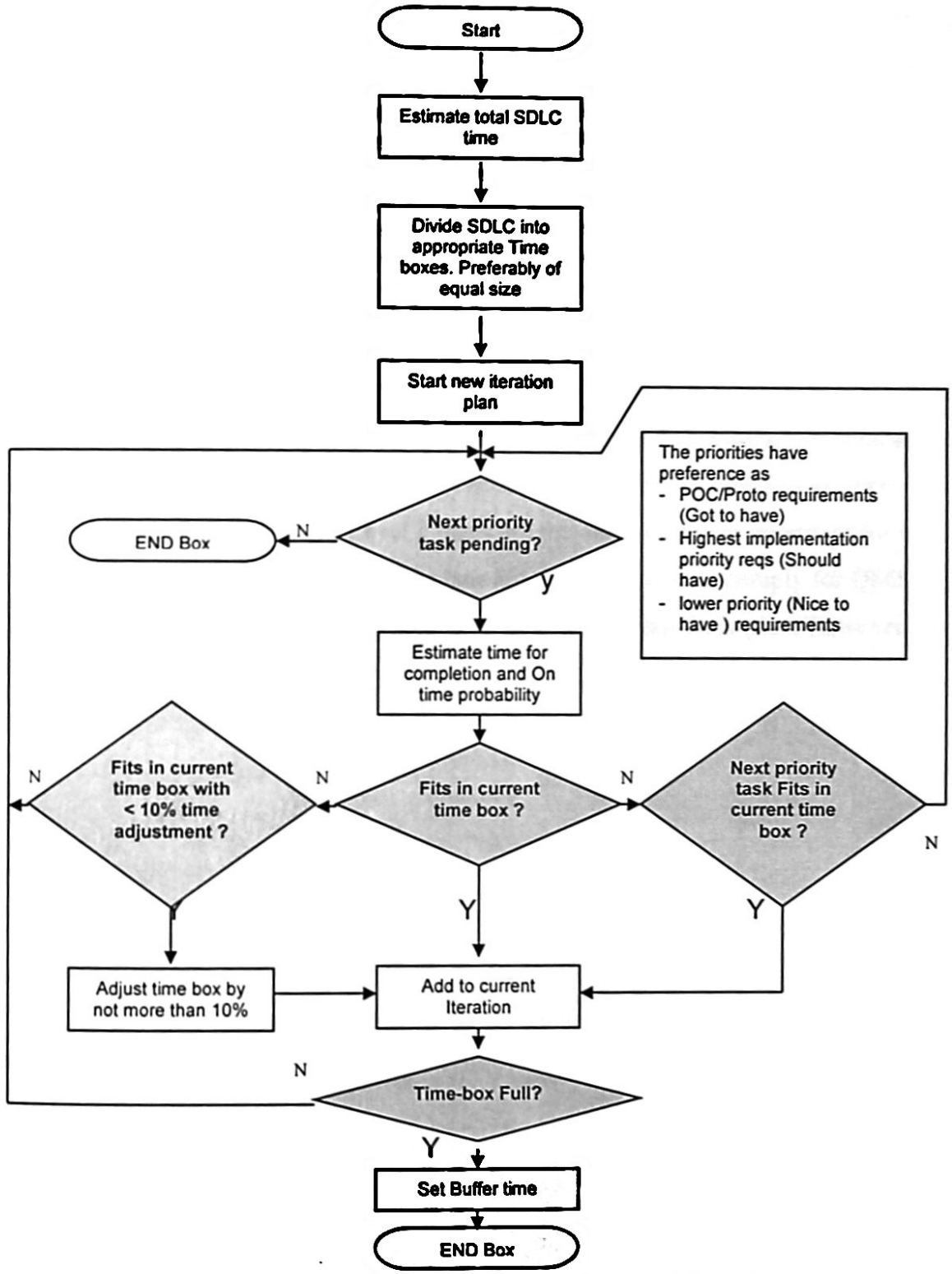

Figure 5.1 : Method used to find Iteration Components

The criteria for iteration plan are to construct a meaningful integrated deliverable. As we have seen the system has two parts. One is hardware and one is software. Being an embedded system both the parts or components of these parts are interdependent. They will not be functional independently until they have integrated to some functional extent. Thus even if the individual module carries some business value it will not be worthwhile even if ready for delivery independently. So the iteration was planned as to deliver the modular system that carries the right business value.

It is also important to mention here that the organization is ISO certified hence the activities planned should be within the framework of QMS. Similarly some specific requirements like MDD requirements impose some framework like clinical evaluation, compliance to the standards etc. These requirements are captured in Level 1 plan which is a Overall project Plan. Level 2 plans are the ones which are important for us. These are individual iteration plans. The plans are designed so as to comply for QMS requirements. The QMS requirements are handled inherently within these plans however those are not referred for the analysis.

5.1.3.7 Implementation Results

Following is the Deliverables analysis summary.

(Please refer Appendix C: Deliverables Analysis for more details)

Table 5.8 : EDS Deliverables Analysis

Deliverable	Total		Delivery1		Delivery2		Delivery3	
	Total	%	Total	%	Total	%	Total	%
Delivered Total	76	100	24	32	22	29	16	21
High priority (>7) deliverables	20	100	18	90	2	10	0	0
Medium priority deliverables (4 - 6)	10	100	4	40	4	40	2	20
Low priority deliverables (<4)	46	100	2	4.3	16	35	14	30
Delivery Value index with respect to On time delivery	1	-	0.8	-	1	-	1	-
Average Priority Delivered	-	-	7.63	-	2	-	2.25	-
Average Value index delivered	-	-	0.97	-	1	-	1	-

From above summary it is easily noted that though the deliverables were distributed equally over equal time boxes, the value delivered is at maximum in first delivery. Ninety percent of the high priority deliverables were delivered with 0.8 value index and average priority of 7.63. Left over 10 % high value deliverables were delivered in second iteration with 100 percent of its value.

customer to evaluate, demonstrate the key features and start selling its value, well ahead of actual entry into market with full system. In some cases the end user actually would not have required all the features in such cases the customer has got the product ready in

his hands to generate the business out of it. This also built the confidence of complete product getting delivered on expected final delivery time. Further it assured the deliveries at regular intervals with expected upgrades adding up to the value of product, It helped developers in getting enough time to focus on thorough evaluation of key deliverable and clinical evaluations as required by the QMS.[12]

On the contrary if waterfall method has been implemented customer would not have been assured of the on time delivery in advance. He would have been be able to see the deliverable only at the end of his expected final delivery date leaving very less time for evaluation of the key features delivered, before actually putting the product in market. The probable delay would have caused father value reduction and increasing the costs of development, cost of marketing etc. A very little delay in particular deliverable may be even of low priority deliverable, would have caused the delay in delivery of entire project with all its value accumulated at final delivery. Therefore this new approach has proven itself for maximum value delivery ahead of its actual requirements.

However the conventional waterfall method has been adapted for the tasks other than the technical deliverables due the typical requirements of ISO standards being followed in the organization. For example the design should not be approved without the risk management in inputs stage itself. The clinical investigation must be carried out prior to starting the actual development activities. The release cannot be possible until the verification and validation has been carried out and proper training has been extended to all other departments. For such activities which are mandatory, the waterfall method is the ideal one. Further during the iterative development also the waterfall method was used as embedded method for development of some typical deliverable since some of the deliverables are dependent on other deliverable to be complete. E.g. until the appropriate hardware is ready the software module cannot be delivered for its actual functionality.

[12] Available from: http://www.freewebtown.com/gaigoisaigon/

5.2 EMG Requirements

5.2.1 Software Requirements

Description of form of software delivery and contents

This is outsourced project for third party. It is extension module of existing Main System being supplied by them for muscle strength analysis.

The objective of this module is to provide facility to set the user controls for automated muscle test, calibrate the test setup and analyse the test results.

The requirement is as described further

The software has to provide different user controls either in one .NET dynamic link library (dll) for each control or in one .NET dll containing all user controls. There has to be one user control executing and visualizing live measurements in real time (it is developer's decision if developer delivers one user control for EMG professional and one for EMG mobile or one for both) and one user control visualizing the analysis of a completed measurement.

Additionally some user controls for calibration and the adjustment of options and measurement sets have to be provided. A detailed description of the required functionality of each user control will follow in the next paragraphs. An idea about the look and feel of the visualizations has already been given as screenshots.

Developer has to ensure that it is easy for (customer) Customer to localize the software for different countries. All texts in all controls could be saved in a file that can be edited without the change of software. In this way a translation of the software for different markets can be done by customer. This includes markets in East Asia with Chinese or similar languages.

5.2.1.1 User control for live measurement

This control has to contain all relevant functions necessary to do a measurement with a chosen set of examinations. There will be two different kind of electronic boards for EMG professional and EMG mobile. That means two different communications have to be implemented and the one that the user wants to use must be able to switch before the measurement through the options or one control for each EMG version has to be delivered by Developer.

EMG professional only

At first this control has to provide an automatic adjustment of the position of two motors in the EMG hardware based on the patient's body height (patient information as name, gender, body height, body weight, etc will be given through interface functions and should be saved with results in the result file). Computation for these motor positions will be provided by Customer. There should be a security notice to ensure nobody is sitting in the EMG while motors are in motion. If a correction of the automatic motor positions has to be done manually on the EMG hardware, new motor positions should be stored in a file in the patient's directory. These positions in the file (if existing) should then be used when adjusting motor positions before the next measurement of this patient.

After the correct adjustment of the motor positions the choice of a set of measurements has to be done. The possible sets of measurement have to be defined before, see 1.3. The control has to guide the user through the measurements, such that even an untrained user can do the measurements. This could be done e.g. through images and texts displayed which are defined in the user control to configure measurement sets.

All versions

The measurement has to start automatically after a threshold (given in the options for each measurement set) has been exceeded and last until the measurement values will go below a threshold (or a minimum measurement time has been fulfilled). Measurement values have to be displayed in real time during the measurement in a graph Biofeedback has to be implemented. This should be prepared in the following way:

Two given graph curves should be drawn over the time between which the patients current graph curve should lay. There has to be a possibility to change the definition of these curves over time. This could be done by defining them in a readable text file. After the selected sets of measurements are completed the data has to be saved in a directory given through an interface function plus a subdirectory created by the control (name created from the measurement date, e.g. 20080320_135919). The format of the file contents should be xml if that is fast enough to read and the size of the files will be reasonable. A message or something similar has to be sent to inform the main application that the measurements are done and saved.

5.2.1.2 User control for analysis

This control has to be able to open and read a completed measurement. The filename of the file to open will be given through an interface function. Normally each set of measurement will contain two corresponding measurement groups (e.g. Extension and Flexion, Lat Flex Right and Lat Flex Left, etc). In the analysis these two groups of each measurement set have to be visualized in a comparison, either with two graphs or with pie charts. Usage of standard values in the visualization for the specific patient group (depending on age, gender and weight or similar properties) and the specific measurement set should be prepared (as no real values are known at the moment and have to be found in a measurement series after the completion of the project, that means it would be good if they can be placed in a text file and will be read from this if existing, in this way no software changes are necessary to set or change these standard values).

In the graph visualization a part of the measurement time should be selectable in which the integral between the two corresponding group graphs is computed and the value is displayed. The graphs should automatically scale such that the axis intervals depend on the maximum and minimum values of the specific measurement. It should also be possible to zoom in specific regions of the graph. I recommend the open source "ZedGraph" library for this. The results should also be printable with both visualization methods (graph and pie charts), depending on what is currently displayed on the screen. These print functions should be available as interface functions. Patient information, date of examination, frames, impressum etc. will be printed by main application. Only result visualization has to be printed by this control.

5.2.1.3 User control to configure measurement sets

Usually there will be two force directions included in each measurement set. E.g. Flexion/Extension or Lat Flex Left/Lat Flex Right. All in all there are ten of such measurement sets available with the current electronic board. All ten possibilities have to be implemented, such that in a complete selection ten different sets of measurements could be saved as one examination. A control has to be delivered in which these sets can be defined. That includes a name for the set, the setup of the involved hardware parts and if necessary the length of the involved lever arm and an Image that will be displayed to explain the measurement. Also texts describing each set of measurements have to be defined (a choosable text file is recommendable to globalize the texts for different languages). The following image gives an idea about this control:

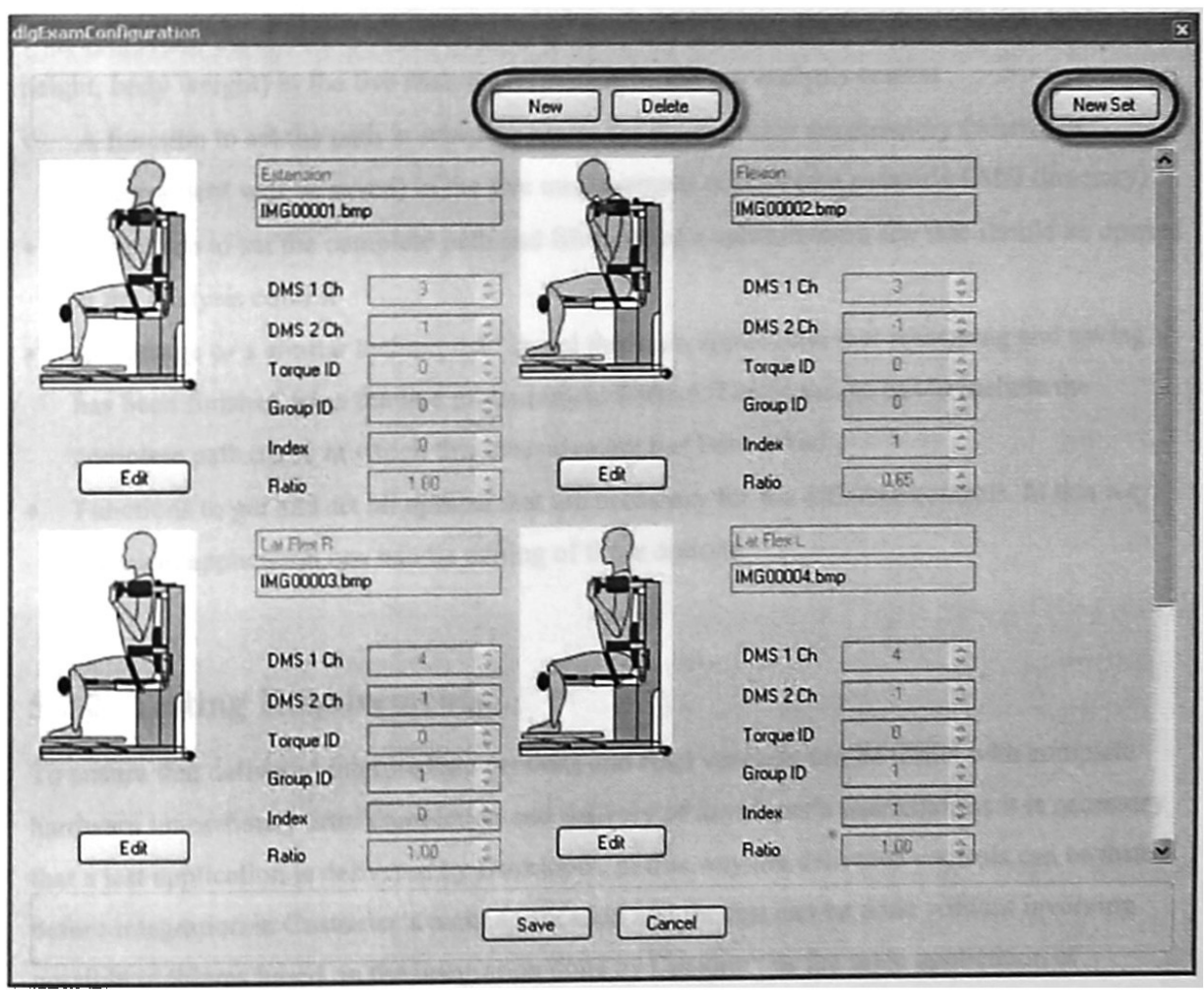

Figure 5.2 : EMG Sample Screen

5.2.1.4 User control for calibration and other hardware related options

In this project Developer has to deliver a graphical user interface to fulfil this task. A user control has to be created. The "user control for live measurement" can then read out the calibration values directly from the electronic board and use them to calculate from digits to kg. The calculation done in the table described in will be provided by Customer. Other options such as options for the serial port should be available over interface functions.

5.2.1.5 Interface functions

This part will describe functions of the controls that have to be made available in the user controls of Developer as public functions to be used by the main application.

Functions to set information about the patient (name, first name, gender, date of birth, body height, body weight) in the live measurement control and the analysis control

- A function to set the path in which to create the examination subdirectory (where the measurement will be saved) in the live measurement control (the patient's EMG directory)
- A function to set the complete path and filename of a measurement file that should be opened in the analysis control
- A message or a similar technique to signal the main application that measuring and saving has been finished from the live measurement control. This message has to include the complete path name in which this measurement has been saved.
- Functions to get and set all options that are necessary for the different controls. In this way the main application can handle editing of these options

5.2.2 Testing Requirements

To ensure that delivered intermediate versions and final versions can be tested with complete hardware immediately after completion and delivery of developer's user controls it is necessary that a test application is delivered by Developer. In this way the delivered controls can be tested before integration in Customer's main application and the test can be done without involving possible problems based on the integration done by Customer or the main application of Customer itself.

5.2.3 Requirement Table

Table 5.9 : EMG Summary Features

S. No.	Feature required
1	User control for live measurement
2	User control for Analysis (To access the different analysis report)
3	User control for configure measurement set (Configure the test setup)
4	User control for calibration and other hardware controls (Machine calibration Positions of measuring arms, measurements)
5	Interface with hardware set up (Comport)

5.2.4 Requirement Prioritization

The prioritization is done on the basis of requirements. This is a composite application with very tight integration. None of the module can run individually, neither has it had individual business value. So priority of all the modules is top most.

All the requirements are "must have" type because the system is not useful without any of these individual features listed in the requirements. There is no requirement of any "Nice to have" or "Should Have" features.

Table 5.10 : EMG Customer Priority Table

S. No.	Feature required	Priority	Comments
1	User control for live measurement	10	Must have
2	User control for Analysis	10	Must Have
3	User control for configure measurement set	10	Must Have
4	User control for calibration and other hardware controls	10	Must Have
5	Interface with hardware set up	10	Must Have

There is not much complexity in the functions/application hence there is no possibility for incremental development and hence no iterations are possible. The size of the project life cycle is too small; it can not be divided in small modules with respect to business values. Thus only the

completed project in all manners carries the business value. So this project needs to be developed in linear go and must be delivered in target time.

The Waterfall approach is one of the popular development approaches for such kind of project. But there is risk of schedule overrun because of its process oriented nature. The time line is too short and unanticipated changes and delays are expected. So a single step minor schedule overrun may lead to cumulative delays resulting into heavy resource overrun that can not be recovered in such a short time line.

To overcome these issues Agile approach would be an ideal option because of its flexible nature. It is less process oriented and more focused on execution rather than rules and documentation. The developer can decide his own priority within the given time line and execute the development so as to get the project delivered within specified time. Since the duration of project is small and the team size is small if during development there are changes the developer can make modifications after discussing it with the team members and deliver the project in time.

6 Conclusion and Recommendations

6.1 Conclusion

A well optimized mix of conventional waterfall approach , Time boxing and Agile techniques, with the help of incremental and iterative development, can be effectively used to improve the On time deliveries of business values rather than focusing on complete deliveries at one go, which could lead to delays causing the loss of its business values.

It helps both the developer as well as the customer in advance evaluation of the deliverable, building confidence of on time deliveries, and deliveries at regular intervals gradually upgrading the system to its completeness in definite stages. In some scenarios waterfall development approach will be optimum if the customer's requirements are clear, will not change and the delivery will be at the end. Agile methods allow for specification changes as per end-user's requirements which imply customer satisfaction. The project should adapt to changing circumstances, and even late changes in the requirements are welcome.

6.2 Recommendations

It is recommended that the suggested time boxing can be applied for almost all the projects in Embedded Biomedical Projects by small organizations where the timely delivery is the major challenge due to limited resources and typical characteristics of Biomedical systems e.g. requirement of rigorous clinical evaluations in view of patient safety.

However it is a challenge to determine the right candidate tasks for particular methods out of the specified mix of different approaches like time boxing and other conventional methods so as to get the optimum results. To achieve the best possible results, the requirements must be well defined, understood, and analyzed for selection of development approach for the particular task. The management techniques will help in prioritizing the tasks and breaking the projects in appropriate iterations.

While planning the iterative deliveries the project manager should not ignore the human psychology that will greatly affect the acceptance of half cooked project delivery. The customer, internal as well as external, should be made aware of this iterative delivery process and its significance in the timely delivery and completeness of the projects, prior to planning the iterations or increments. The iterations and increments should be planned so as to give some well defined business value output during the concerned delivery. Otherwise there is a risk of rejection of the delivery due to incomplete delivery.
In projects Agile will be useful if the co-ordination between the manager and team members is at regular intervals, if changes are there they are made and the system is delivered successfully in time. In Agile there is flexibility in specifications and dynamic changes are acceptable in any stage of development.

There is further scope for improvement in these suggested modules depending on the specific requirement and characteristics of the projects. Further there is scope for more accurate analysis by consideration of value of advance deliveries, value of deferred deliveries in next iteration etc.

Bibliography

Books:

1. Grady Booch, James Rumbaugh, Ivar Jacobson,2007 , *Unified Modeling Language User Guide* , New Delhi, Pearson Education
2. Waman Jawadekar, *Software Engineering*,2004, *Software Engineering Principles and Practice*, Delhi, Tata McGraw-Hill
3. Ivar Jacobson, Grady Booch, James Rumbaugh, 2003,*The Unified Software Development Process*, New Delhi, Addison Wesley Longman
4. Craig Larman , 2006, *Applying UML and Patterns* , New Delhi, Pearson Education
5. R.S.Khandpur, 2005, *Handbook of Biomedical Instrumentation*, New Delhi, Tata McGraw-Hill

Web Pages:

1. *The Waterfall Model* [online] Available from: http://en.wikipedia.org/wiki/Waterfall_model (Accessed 5th Jan 2008)
2. *The advantages and disadvantages of Waterfall Model* [online] Available from: http://www.blurtit.com/q533918.html (Accessed 5th Jan 2008)
3. Agilecollab, *The Waterfall method vs. Agile methodology* [online] Available from: http://agileintro.wordpress.com/2008/01/04/waterfall-vs-agile-methodology (Accessed 10th Jan 2008)
4. Pankaj Jalote , Aveejeet Palit, Priya Kurien, V. T. Peethamber ,*Time boxing : A Process Model for Iterative Software Development* [online] Available from: http://www.cse.iitk.ac.in/users/jalote/papers/Timeboxing.pdf (Accessed 22nd Mar 2008)
5. *Iterative and Incremental development* [online] Available from: http://en.wikipedia.org/wiki/Iterative_and_incremental_development (Accessed 17th Feb 2008)

6. Rick Zahniser, 2005,*Time boxing For Top Team Performance* [online] Available from: http://www.belizenorth.com/articles/TIMEBOX.htm (Accessed 17th Feb 2008)
7. Daniel Marr (October 2001), *ECG Application Featuring Data Transmission by Bluetooth* [online] Available from: http://innovexpo.itee.uq.edu.au/2001/projects/s369535/thesis.pdf (Accessed 17th Feb 2008)
8. Hong Liang (2005), *An application of ECG Feature Identification for Cardiologist Expert Diagnosis* [online] Available from: http://www.ncbi.nlm.nih.gov/pubmed/17281069 (Accessed 17th Feb 2008)
9. Frank G Yanowitz, MD, *Performance of exercise ECG Testing,* [online] Available from: http://www.uptodate.com/patients/content/topic.do?topicKey=~d8x8vk/U6v (Accessed 19th Feb 2008)
10. Manikandan MS, ADCOM Proceedings of the 15th International Conference on Advanced Computing and Communications, *Wavelet-Based ECG and PCG Signals Compression Technique for Mobile Telemedicine,* [online] Available from: http://portal.acm.org/citation.cfm?id=1333727 (Accessed 10th Mar 200)
11. Dave Cheong ,July 26th, 2006 , Time Boxing is an Effective Getting Things Done *Strategy* [online] [online] Available from: http://www.davecheong.com/2006/07/26/time-boxing-is-an-effective-getting-things-done-strategy/ (Accessed 21st Sep 2008)
12. *Visualizing Time-Series on Spirals , 2001* [online] Available from: http://www.freewebtown.com/gaigoisaigon/ (Accessed 19th Apr 2008)
13. *Time Boxing* [online], Available from : www.wikipedia.org (Accessed 2nd Jan 2008)
14. J. Meires, *Software Engineering. Project Management, Effectiveness How To Use Time Boxing for Getting Results Blog* [online], Available from : http://blogs.msdn.com/jmeires/default.aspx (Accessed 20th Feb 2008)
15. Dr Juliette Lee, 2006, *ECG Monitoring in Theatre* [online], Available from : http://mcu.motsps.com (Accessed 17th Mar 2008)
16. *Stress ECG* [online], Available from : http://indianheartjournal.com/20015/MayJune2003/StressEchocardiography (Accessed on 16th Jan 2008)

17. Craig Ferguson, Dec 2005, *Does a normal ECG rule out the diagnosis of heart failure in the breathless patient* [online], Available from : http://www.bestbets.org/bets/bet.php?id=865 (Accessed 25th May 2008)

18. A de kozakevicius,2005, *Adaptive ECG Filtering and QRS Detection using Orthogonal Wavelet Transform* [online], Available from : http://www.actapress.com/PaperInfo.aspx?PaperID=19476&reason=500 (Accessed 19th July 2008)

19. Harry Hochheiser, 2005,*Visual Specification of Queries for Finding Patterns in Time-Series Data* [online], Available from : http://hcil.cs.umd.edu/trs/2001-05/2001-05.html (Accessed 26th June 2008)

20. *ECG* [online], Available from : http://www.sciencedirect.com/science (Accessed 30th Aug 2008)

21. Dr. A.Kalia, *Basics of ECG*[online], Available from : http://www.slideworld.org/slideshow.aspx/ecg-basics-ppt-2842787 (Accessed 11th July 2008)

22. *ECG Basics* [online], Available from : http://en.ecgpedia.org/wiki/Basics (Accessed 15th June 2008)

23. Brij N. Singh and Arvind Tiwari (May 2006), *ECG signal de-noising* [online], Available from : http://www.citeulike.org/user/frednicolier/article/614968 (Accessed 23rd October 2008)

24. *Agile S/W Development* [online], Available from : http://en.wikipedia.org/wiki/Agile_software_development (Accessed 5th Apr 2009)

Appendix A: EDS Priority Matrix

Project : EDS

IMPLEMENTATION PRIORITY MATRIX

EDS PRIORITY MATRIX

S. No.	Requirements	Category	Customer Priority	On time - probability	Implement priority
A	Software Requirements				
A.1	General Requirements				
1	Patient Information	Should	8	1	8
2	Patient Data Storage - Facility to store patient data (Date & Patient wise)	Should	8	1	8
3	Patient Data Storage - DICOM,HL7,HIPPA,TWIN	Should	5	0.7	3.5
4	Data Display Set 1 - Online, Offline, review, Zoom	Should	10	1	10
5	Data Display Set 2	Nice	7	0.9	6.3
	Annotations and Marker facility(user friendly and localized ie. On toolbar)				
	Color-coded display for different abnormalities.				
6	LAN Connectivity	Should	6	1	6
7	Printing Set 1	Should	10	0.8	8
	Grid selection facility.				
	Print preview facility.				
8	Printing Set 2	Nice	6	0.8	4.8
	Color selection facility.				
	Report customization (add or remove parameters)				
	System should have facility for editing, reviewing of final reports				
9	Data on CD	Nice	3	0.9	2.7
10	Tele medicine 1 - Facility to send patient data across the net for consultancy	Should	3	0.5	1.5
11	Patient data should be available online on the net.	Nice	2	0.5	1
12	Patient data security	Should	2	0.5	1
13	Capture system / PC configuration Set 1 - Monitor size, Hard dosk size, USB port	Should	5	0.8	4
14	Capture system / PC configuration Set 2 - Sound Card	Nice	2	0.8	1.6
15	Gain Selection facility	Should	7	1	7
16	Key board interface for all operations	Nice	2	0.9	1.8
17	Alarm facility	Nice	2	0.9	1.8
18	Port selection facility	Should	8	1	8
19	Impressions	Nice	3	1	3
20	Automatic Upgrade - System should be upgraded at customer's end automatically by website when there is a Version change in s/w.	Nice	1	0.7	0.7
21	Touch Screen	Nice	1	0.7	0.7
22	Voice Prompt	Nice	1	0.5	0.5
23	Vector Scale	Nice	1	0.5	0.5
24	Vector Animation	Nice	1	0.5	0.5
25	Vector Direction	Nice	1	0.5	0.5
A.2	DECG Requirements				
1	Waveforms display tiles Set 1	Should	10	0.9	9
	12*1				
	12*1+ Median				
	6*2				
	6*2+ Median				
2	Waveforms display tiles Set 2	Should	7	0.8	5.6
	Rhythm Leads + Medians				
3	ECG Data storage / retrieval set 1	Should	9	0.8	7.2
	System should store and retrieve data for 10,20,30 and 40 seconds.				
4	ECG Data storage / retrieval set 1	Nice	4	0.8	3.2
	System should provide the facility to store the ECG data for full test and it should be available for review as a full disclosure for the doctor as and when required.				
5	Parameters Display Tiles - POC	Got to	8	0.9	7.2
	Tile1 –HR/BP				
	Tile2 -ST bar graph				
	Tile3 – Axis				
	Tile4 –ST map				
6	Parameters Display Tiles - Upgrade 1	Should	5	0.8	4
	Tile5 – Amplitude chronograph				
	Tile6- Parameter table(All durations for all leads)				
	Tile7-Durations(PD,QRS,PR,QT,QTC,VAT for selected lead.)				
7	Parameters Display Tiles - Upgrade 2	Should	3	0.8	2.4
	ST trend Tile				
	HR trend Tile				
	BP trend Tile				
	Abnormality list Tile				
	Dynamic medians				
	Median Comparison				
8	ECG Analysis POC	Got to	4	0.7	2.8

Project : EDS **IMPLEMENTATION PRIORITY MATRIX**

S. No.	Requirements	Category	Customer Priority	On time probability	Implement priority
9	ECG Analysis - Clinical Evaluation (All parameters above)	Got to	3	0.7	2.1
10	ECG Analysis - Release	Should	2	0.7	1.4
A.3	EECG Requirements				
1	Rhythm leads selection	Nice	2	0.7	1.4
2	Protocols selection Predefined set	Should	3	0.8	2.4
3	Protocol Selection User selectable	Nice	1	0.7	0.7
4	Expanded medians	Should	2	0.8	1.6
5	Dynamic Medians	Should	2	0.7	1.4
6	Treadmill interface	Should	3	0.9	2.7
7	Report Generation POC	Got to	3	0.9	2.7
8	Report Generation Demo	Should	2	0.9	1.8
9	Report Generation Release	Should	1	0.9	0.9
10	Previous Report selection memory	Nice	1	0.9	0.9
11	Check Box for selection	Nice	1	0.9	0.9
12	Termination criteria selection	Should	2	0.9	1.8
A.4	Ambulatory ECG (Holter) Requirements				
1	Data storage	Should	2	1	2
2	Reports for Holter System POC - Selective reportsAll reports specified	Got to	2	0.8	1.6
3	Reports for Holter System Upgrade 1	Should	1	0.8	0.8
4	Reports for Holter system Final Release	Should	1	0.7	0.7
5	Online viewing of ECG	Should	2	1	2
6	Full Disclosure POC	Got to	2	0.9	1.8
7	Full disclosure upgrade 1	Should	1	0.8	0.8
8	Full disclosure final Advanced	Nice	1	0.8	0.8
9	Holter analysis of ECG data from any other system	Nice	1	0.7	0.7
B	Hardware Requirements				
B.1	General Requirements				
1	Sampling rate - 500 samples/s/ch	Should	10	0.9	9
2	A to D conversion speed 10ksps	Should	10	1	10
3	ADC resolution 12 bit	Should	10	0.9	9
4	Filters software	Nice	8	0.8	3.6
5	Pacemaker Detection	Nice	2	0.6	1.2
6	Lead fail information	Should	3	0.7	2.1
7	Audio indication	Nice	1	0.8	0.8
8	Defib Protection	Should	4	0.8	3.2
9	Communication Ports - USB port	Should	8	0.9	7.2
10	Physical Dimensions - Pocket sized copmact device	Should	7	0.7	4.9
11	Isolation As per IEC requirements	Should	6	1	6
12	Safety standards As per ISO and MDD requirements	Nice	2	0.6	1.2
B.2	ECG Amplifier Requirements				
1	ECG acquisition 12 lead simultaneous	Should	9	0.9	8.1
2	Number of channels - 12	Should	8	0.9	7.2
3	Patient cable 10 lead	Should	8	1	8
4	Patient Cables options for 3,5 and 7 lead	Nice	2	0.8	1.6
5	Power Supply - To be derived from PC through USB port	Should	9	0.8	7.2
6	Rhythm Leads	Nice	1	0.5	0.5
B.3	Ambulatory ECG (Holter) Requirement				
1	ECG acquisition 9 lead	Should	3	0.9	2.7
2	ECG acquisition 12 lead	Nice	2	0.7	1.4
3	Number of channels 3 - Leads fixed	Should	3	1	3
4	Numer of channels 3 - Any 3 leads selection	Nice	2	0.8	1.6
5	Patient cable 7 / 10 leads	Should	2	1	2
6	Power for system should be derived from PC bus and 1.5volts(AA size) battery. - 24 Hour back up on battery	Should	3	0.8	2.4
7	72 Hour back up on battery	Nice	1	0.6	0.6
8	Defib Protection	Should	2	0.7	1.4

Appendix B: EDS Project Plans

DESIGN & DEVELOPMENT PLAN – LEVEL 1

PROJECT: EDS | Ref. No.: DPLN/EDSL1/Rev 1.0 | Date: 21/04/07

Plan Reviewed: SP	Plan Approved: AS

ASSIGNMENT No

Type of Project	NEW	DESIGN CHANGE	OTHER (Specify)-		
	Implementation	Lead / Control	Review / Approvals	Special Requirements	Proposed Dates
Interfaces / R & A	SMK,VAS,AnayK, Yogesh U,AnujK, Ketakee	SP/VAS	AS/SP		START:21/04/07 RELEASE: 20/10/08

TIME LINE:

Task / Functionality ID	R & A	Expected date	Review Date Status and remarks
Requirements Analysis & Review	SP/VAS	24/04/07	26/04/07 - The requirements are feasible
Standards / Statutory & Regulatory Requirement Analysis & Review	SP/VAS	28/04/07	26/04/07 -No statutory and regulatory requirements applicable
Risk Management	SP/VAS	30/04/07	05/05/07 – HID done, Risk – priority analysis done.
Clinical Evaluation Plan	SP/VAS	1/05/07	05/05/07 – Done
Feasibility Study & Review	SP/VAS	07/05/07	10/05/07 – Small size and Multiple device interface is challenge. To consider proper development approach
High Level Design and approval	SP/VAS	17/05/07	20/05/07 – Small size and Multiple device interface is challenge. To consider proper development approach
General Test & Quality Plan	SP/VAS	30/05/07	30/05/07 – Done
Resource Allocations	SP/VAS	30/05/07	30/05/07 – Done

S = Start, D = Delayed, P = Postponed, % = Progress %, NA = Not Applicable, CO = Carried Over, CF = Carried Forward, O = Completed, X = Stopped / Discarded

PROJECT: EDS | Ref. No.: DPLN/EDSL1/Rev 1.0 | Date: 21/04/07

Task / Functionality ID	R & A	Expected date	Review Date Status and remarks
Prototype Development	SP/VAS	01/07/07	02/07/07 – Demo of basic display screen, dummy enclosure satisfactory
Iterative Developments / Integrations	SP/VAS	01/06/07 to 30/08/08	30/09/07 – One Deliverable has overrun. Advance work has been done for 4-5 deliverables – Completed prototype 9Iteration 1 deliverable accepted with some change requirements please refer change request and feature upgrade requirements – 24 deliverables 15/10/07 – Delayed delivered completed, 30/03/08 – Upgraded iteration 2 – for DECG is ready for field evaluation, – Pending features to be reviewed and incorporated in next iteration – 22 deliverables 02/09/08 - Iteration OK. Some parameters are not planned to be delivered - 16 deliverables - Rhythm lead, pacemaker detection, holter hardware release halted since there functionalities are not required by customer currently.
System Verification	SP/VAS	30/09/07, 20/03/08, 30/08/08,	Refer test reports for bug reports and bug fixing
System Validations	SP/VAS	Oct 07, Apr-May08, Spet 08	05/05/08 - System validated iteration wise. Issues were reported and corrected accordingly
Design Change Review and Implementation	SP/VAS	Sept 08	Noise improvement in PCECG12 called for PCB iteration. Observed noise reduction Patient information parameters reduced to bare minimum requirements
Production	SP/VAS	Sept 08	27/09/08 – Trial production carried out under D&D supervision
Final Review & Approval	SP/VAS	05/10/08	04/10/08 – No issues. Design approved
Documentation	SP/VAS	05/10/08	Done
Training	SP/VAS	05/10/08	Done
System Release	SP/VAS	20/10/08	17/10/08 – Final system released for regular production

S = Start, D = Delayed, P = Postponed, % = Progress %, NA = Not Applicable, CO = Carried Over, CF = Carried Forward, O = Completed, X = Stopped / Discarded

DESIGN & DEVELOPMENT PLAN – LEVEL 2

PROJECT: EDS | Ref. No.: DPLN/EDSL2/Rev 1.0 | Date: 10/05/07

Plan Reviewed: SP	Plan Approved: As

ASSIGNMENT No: EDSPROTO1
Assignment: Deliver the prototype for general evaluation / demonstration
Prerequisits:
Post Processing:
Time Box Size: 16 weeks Number of Deliverables: 24

	Implementation	Lead / Control	Review / Approvals	Special Requirements	Proposed Dates
Interfaces / R & A	SMK,VAS,AnayK, YogeshAnujK, Ketakee	SP/VAS	APS/SP		START:01/06/07 RELEASE: 30/09/07

TIME LINE:

Task / Functionality ID	R & A	MDays	Time Line(week Number)																Remark
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Requirements Analysis & Review	SP/VAS	2	S/O																
Standards / Statutory & Regulatory Requirement Analysis & Review	SP/VAS	1/2	S/O																
Risk Management	SP/VAS	1/2	S/O																
Clinical Evaluation Plan	SP/VAS	1/2	S/O																
Feasibility Study & Review	SP/VAS	1/2	S/O																
Designs	SP/VAS	8	S	O															
Test & Quality Plan	AnujK	8	S	O															
Resource Allocations	SP/VAS	1/2	S																

S = Start, D = Delayed, P = Postponed, % = Progress %, NA = Not Applicable, CO = Carried Over, CF = Carried Forward, O = Completed, X = Stopped / Discarded

PROJECT: EDS | Ref. No.: DPLN/EDSL2/Rev 1.0 | Date: 10/05/07

Task / Functionality ID	R & A	MDays	Time Line(week Number)															Remark
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Prototype Development	NA	NA																	
Iterative Developments / Integrations																			
A1..1	Yogesh	15								S	O								
A1..2	Yogesh	3										S/O							
A.1.4	VAS	30			S				O										
A.1.7	Anay K	15							S	O									
A.1.9	Anay K	8																	
A.1.10	Anay K	30											S						Advance work
A.1.13	VAS	8											S/O						
A.1.15	Yogesh	4										S		O					
A.1.18	Yogesh	2			S/O														
A.1.19	Yogesh	2												S/O					
A.2..1	Yogesh Anay K	25							S			O							
A.2.3	VAS	6								S/O									
A.2.5	Yogesh	35										S						D	Overrun
A.2.7	Yogesh	30											S						Advance Work
A.2.8	Anay K VAS	30											S						Advance work

S = Start, D = Delayed, P = Postponed, % = Progress %, NA = Not Applicable, CO = Carried Over, CF = Carried Forward, O = Completed, X = Stopped / Discarded

PROJECT: EDS | Ref. No.: DPLN/EDSL2/Rev 1.0 | Date: 10/05/07

Task / Functionality ID	R & A	MDays	Time Line(week Number)																Remark
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
A.4.1	SMK	8			S	O													
A.4.2	SMK	30					S												Advance work
A.4.3	SMK	15					S												Advance work
A.4.4	SMK	15					S												Advance work
A.4.5	SMK	30					S												Advance work
A.4.6	SMK	45					S												Advance work
A.4.7	SMK	20					S												Advance work
A.4.8	SMK	60					S												Advance work
B.1..1	KET	30			S	10 %		90 %	O										
B.1.2	KET	-			S				O										
B.1.3	KET	-			S				O										
B.1.4	AnayK	8														S			
B.1.8	KET	2								S									
B.1.9	SP/KET	15		S		O													
B.1.10	SP	8								S		0							
B.1.11	KET	1			S/O														
B.1.12	KET	15												S					Advance Work
B.2..1	SP/KET	15				S				O									
B.2.2	SP/KET	-			S/O					S			O						
B.2.3	SP/KET	2			S/O														
B.2.5	KET	1			S/O														
B.3..1	SP	30																	Prototype ready but not required
B.3.2	TBD	TBD																	Prototype ready but not required
B.3.3	SP	30																	Prototype ready but not required
B.3.4	SP	15																	Prototype ready but not required
B.3.5	SP	2																	Prototype ready but not required

S = Start, D = Delayed, P = Postponed, % = Progress %, NA = Not Applicable, CO = Carried Over, CF = Carried Forward, O = Completed, X = Stopped / Discarded

PROJECT: EDS | Ref. No.: DPLN/EDSL2/Rev 1.0 | Date: 10/05/07

Task / Functionality ID	R & A	MDays	Time Line(week Number)															Remark
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
B.3.6	SP	8																	Prototype ready but not required
B.3.7	SP	15																	Prototype ready but not required
B.3.8	SP	8																	
Deliverable Verification	SP/VAS/ AnujL	15										S				O			
Deliverable Validations	AnujK	30													S			O	
Design Change Review and Implementation	SP/VAS	30													S	O			
Review & Approval	SP/VAS/ APS	1																S/ O	
Documentation	AnayK/ KET/An ujK	10								S							O		
Training	AnayK/ KET	8														S	O		
Release	SP/VAS	1																S/ O	
																		S/ O	

S = Start, D = Delayed, P = Postponed, % = Progress %, NA = Not Applicable, CO = Carried Over, CF = Carried Forward, O = Completed, X = Stopped / Discarded

DESIGN & DEVELOPMENT PLAN – LEVEL 2

PROJECT: EDS | Ref. No.: DPLN/EDSL2/Rev 1.0 | Date: 10/11/07

Plan Reviewed: SP	Plan Approved: AS

ASSIGNMENT No: EDSPROTO2
Assignment: Deliver the Beta system
Prerequisits:
Post Processing:
Time Box Size: 16 weeks No of Deliverables: 22

Interfaces / R & A	Implementation	Lead / Control	Review / Approvals	Special Requirements	Proposed Dates
	SMK,VAS,AnayK, YogeshAnujK, Ketakee	SP/VAS	APS/SP		START:21//11/07
					RELEASE: 20/03/08

TIME LINE:

Task / Functionality ID	R & A	MDays	Time Line(week Number)															Remark
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Risk Management	SP/VAS	1/2	S/O																
Designs	SP/VAS	8	S	O															
ResourceAllocations	SP/VAS	½	S/O																
Iterative Developments / Integrations																			
A.1.3	Yogesh	8												S/O					
A.1.5	VAS	10		S	O														
A.1.6	Anay K	8		S/O															
A.1.10	Anay K	30				S				O									
A.1.17	Anay K	2									S/O								
A.1.23	VAS	15				S	O												
A.1.24	VAS	15						S	O										
A.2.2	Anay K VAS	10							S		O								
A.2.5	Yogesh	35	D			O													Complete

S = Start, D = Delayed, P = Postponed, % = Progress %, NA = Not Applicable, CO = Carried Over, CF = Carried Forward, O = Completed, X = Stopped / Discarded

PROJECT: EDS | Ref. No.: DPLN/EDSL2/Rev 1.0 | Date: 10/11/07

Task / Functionality ID	R & A	MDays	Time Line(week Number)																Remark
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
A.2.6	Yogesh	25				S			O										
A.2.7	Yogesh	10	S	O															Advance activity has done in prev. iteration
A.2.8	Anay K VAS	10									S	O							
A.2.9	VAS										S	O							
A.3.7	Anay K	30											S				O		
A.3.8	Anay K	15															S	O	
A.4.2	SMK	8	S/ O																
A.4.3	SMK	8		S/ O															
A.4.4	SMK	8			s/o														
A.4.5	SMK	10				S	O												
A.4.6	SMK	15						S	O										
A.4.7	SMK	8								S	O								
A.4.8	SMK	10									S	O							
B.2.4	KET	4			S/ O														
B.2.6	TBD	TBD																	
Deliverable Verification	SP/VAS/ AnujK	15												S			O		
Deliverable Validations	AnujK	30													S			O	
Design Change Review and Implementation	SP/VAS	10													S	O			
Review & Approval	SP/VAS/ APS	1																S	
Documentation	AnayK/ KET/An ujK	5														S	O		

S = Start, D = Delayed, P = Postponed, % = Progress %, NA = Not Applicable, CO = Carried Over, CF = Carried Forward, O = Completed, X = Stopped / Discarded

PROJECT: EDS Ref. No.: DPLN/EDSL2/Rev 1.0 Date: 10/11/07

Task / Functionality ID	R & A	MDays	Time Line(week Number)															Remark
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Training	AnayK/ KET	8														S	O		
Release	SP/VAS	1																S	

S = Start, D = Delayed, P = Postponed, % = Progress %, NA = Not Applicable, CO = Carried Over, CF = Carried Forward, O = Completed, X = Stopped / Discarded

DESIGN & DEVELOPMENT PLAN – LEVEL 2

PROJECT: EDS | Ref. No.: DPLN/EDSL2/Rev 1.0 | Date: 08/04/08

Plan Reviewed: SP	Plan Approved: AS

ASSIGNMENT No: EDSPROTO3
Assignment: Deliver the Beta system
Prerequisits:
Post Processing:
Time Box Size: 16 weeks No of Deliverables: 16

Interfaces / R & A	Implementation	Lead / Control	Review / Approvals	Special Requirements	Proposed Dates
	VAS,AnayK, Manas	SP/VAS	APS/SP		START:01/05/08
					RELEASE: 30/08/08

TIME LINE:

Task / Functionality ID	R & A	MDays	Time Line(week Number)															Remark
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Risk Management	SP/VAS	1/2	S/O																
Designs	SP/VAS	8	S	O															
Resource Allocations	SP/VAS	1/2	S/O																
Iterative Developments / Integrations																			
A.1.8	Anay K	15			S	O													
A.1.11	Anay K	8					S	O											
A.1.12	TBD	TBD																	
A.1.14	VAS	8			S/O														
A.1.16	Manas	3			S/O														
A.1.20	TBD	TBD																	
A.1.21	VAS	10				S/O													
A.1.22	TBD	TBD																	
A.2.4	VAS	7					S/O												
A.2.10	VAS	8												S	O				

S = Start, D = Delayed, P = Postponed, % = Progress %, NA = Not Applicable, CO = Carried Over, CF = Carried Forward, O = Completed, X = Stopped / Discarded

PROJECT: EDS | Ref. No.: DPLN/EDSL2/Rev 1.0 | Date: 08/04/08

Task / Functionality ID	R & A	MDays	Time Line(week Number)															Remark
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
A.3..1	VAS	4						S/O											
A.3.2	Anay K	15							S	O									
A.3.3	Anay K	10								S	O								
A.3.4	Anay K	8										S	O						
A.3.5	Anay K	10											S	O					
A.3.6	Anay K	20				S								O					
A.3.10	Anay K	15												S		O			
A.3.11	Anay K	3												S		O			
A.3.12	Anay K	3												S		O			
B.1.5	TBD	TBD																	
B.1.6	TBD	TBD																	
B.1.7	TBD	TBD																	
B.1.12	TBD	TBD																	
B.2.6	TBD	TBD																	
Deliverable Verification	SP/VAS/ AnujK	15												S			O		
Deliverable Validations	AnujK	30													S			O	
Design Change Review and Implementation	SP/VAS	10													S	O			
Review & Approval	SP/VAS/ APS	1																S/O	
Documentation	AnayK/ KET/An ujK	5														S	O		
Training	AnayK/ KET	8														S	O		
Release	SP/VAS	1																S	

S = Start, D = Delayed, P = Postponed, % = Progress %, NA = Not Applicable, CO = Carried Over, CF = Carried Forward, O = Completed, X = Stopped / Discarded

Appendix C: EDS Deliverables Analysis

EDS

Deliverables Value calculation

S. No.	Requirements	Customer Priority	Delay	Value
A	Iteration 1			
1	Patient Information	8	0	1
2	Patient Data Storage - Facility to store patient data (Date & Patient wise)	8	0	1
4	Data Display Set 1 - Online, Offline, review, Zoom	10	0	1
7	Printing Set 1	10	0	1
9	Data on CD	3	0	1
13	Capture system / PC configuration Set 1 - Monitor size, Hard dosk size, USB port	5	0	1
15	Gain Selection facility	7	0	1
18	Port Selection Facility	8	0	1
1	Waveforms display tiles Set 1	10	0	1
3	ECG Data storage / retrieval set 1	9	0	1
5	Parameters Display Tiles - POC	8	1	0.2
1	Rhythm leads selection	2	0	1
1	Sampling rate - 500 samples/s/ch	10	0	1
2	A to D conversion speed 10ksps	10	0	1
3	ADC resolution 12 bit	10	0	1
4	Filters software	6	0	1
8	Defib Protection	4	0	1
9	Communication Ports - USB port	8	0	1
10	Physical Dimensions - Pocket sized copmact device	7	0	1
11	Isolation As per IEC requirements	6	0	1
1	ECG acquisition 12 lead simulteneous	9	0	1
2	Number of channels - 12	8	0	1
3	Patient cable 10 lead	8	0	1
5	Power Supply - To be derived from PC through USB port	9	0	1
	Average	7.63		0.97
	Iteration total	24		23.2
	Iteration value = Number of Deliverables - sum of deliverable values	0.8		

S. No.	Requirements	Customer Priority	Delay	Value
	Iteration 2			
3	Patient Data Storage - DICOM,HL7,HIPPA,TWIN	5	0	1
5	Data Display Set 2	7	0	1
6	LAN Connectivity	6	0	1
10	Tele medicine 1 - Facility to send patient data across the net for consultancy	3	0	1
17	Alarm facility	2	0	1
23	Vector Scale	1	0	1
24	Vector Animation	1	0	1
2	Waveforms display tiles Set 2	7	0	1
6	Parameters Display Tiles - Upgrade 1	5	0	1
7	Parameters Display Tiles - Upgrade 2	3	0	1
8	ECG Analysis POC	4	0	1
9	ECG Analysis - Clinical Evaluation (All parameters above)	3	0	1
7	Report Generation POC	3	0	1
8	Report Generation Demo	2	0	1
2	Reports for Holter System POC - Selective reportsAll reports specified	2	0	1
3	Reports for Holter System Upgrade 1	1	0	1
4	Reports for Holter system Final Release	1	0	1
5	Online viewing of ECG	2	0	1
6	Full Disclosure POC	2	0	1
7	Full disclosure upgrade 1	1	0	1
8	Full disclosure final Advanced	1	0	1
4	Patient Cables options for 3,5 and 7 lead	2	0	1
	Average	2.00		1.00
	Iteration total	22		22
	Iteration value = Number of Deliverables - sum of deliverable values	1		

S. No.	Requirements	Customer Priority	Delay	Value
A	Iteration 3			
8	Printing Set 2	6	0	1
11	Patient data should be available online on the net.	2	0	1
14	Capture system / PC configuration Set 2 - Sound Card	2	0	1
16	Key board interface for all operations	2	0	1
21	Touch Screen	1	0	1
4	ECG Data storage / retrieval set 1	4	0	1
10	ECG Analysis - Release	2	0	1
1	Rhythm leads selection	2	0	1
2	Protocols selection Predefined set	3	0	1
3	Protocol Selection User selectable	1	0	1
4	Expanded medians	2	0	1
5	Dynamic Medians	2	0	1
6	Treadmill interface	3	0	1
10	Previous Report selection memory	1	0	1
11	Check Box for selection	1	0	1
12	Termination criteria selection	2	0	1
	Average	2.25		1.00
	Iteration total	16		16
	Iteration value = Number of Deliverables - sum of deliverable values	1		

Appendix D: EMG Project Plan

EMG Project Plan

Plan Reviewed: SP	Plan Approved: AS

Task / Functionality ID	R & A	Start date	End date	Status as On date	Status as On date	Status as On date	Status as On date	Status as On date
Requirements Specification	SP/VAS	22/04/08	23/04/08	29/4/08	25/5/08	25/6/08	23/7/08	28/8/08
Specification Review	SP/VAS	23/04/08	24/04/08					
Preliminary Specifications	SP/VAS	23/04/08	24/04/08					
Statutory & Regulatory Requirements	SP/VAS	24/04/08	25/04/08					
Design Details	VAS/AB	24/04/08	02/05/08	30%	Activity Completed			
Resource Plan	SP/VAS	28/04/08	28/04/08	Completed				
General Test & Quality Plan	SP/VAS	05/05/08	12/05/08		Completed			
Design Approval	SP/VAS	12/05/08	12/05/08		Completed			
Risk Management	SP/VAS	12/05/08	19/08/08	10%	Completed			
Development		**30/04/08**	**28/08/08**					
Features development ***Agile Approach***	VAS/AB	30/04/08	20/08/08					
User control for live measurement	VAS/AB			10%	20%	70%	Completed	
User control for Analysis	VAS/AB			20%	20%	60%	80%	Completed
User control for configure measurement set	VAS/AB			25%	50%	Completed		
User control for calibration and other hardware controls	VAS/AB			30%	80%	80%	Completed	
Interface with hardware set up	VAS/AB			30%	80%	80%	Completed	
Verification	VAS/AB	26/05/08	11/08/08		10%	50%	90%	Verified
Validation	VAS/AB	16/06/08	25/08/08				20%	Validated
Main System Integration	VAS/AB	07/10/08	01/08/08			30%	95%	Completed
Release	VAS/AB	28/08/08	28/08/08					Released