**M S Ramaiah Institute of Technology**

(An Autonomous Institute, Affiliated to VTU)

MSR nagar, MSRIT post, Bangalore-54

**Smart Health Care Monitoring System using RaspberryPi2**

Under the guidance of

**Mrs. S. Rajarajeswari**

Submitted by

Himanshu Kumar 1MS12CS039

Suhail T N 1MS12CS115

Manoj more S 1MS13CS412

Suresh V 1MS13CS421

*In partial fulfillment for the award of the degree of*

# *Bachelor of Engineering in Computer Science & Engineering*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**M.S.RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute, Affiliated to VTU)**

**BANGALORE-560054**

[www.msrit.edu](http://www.msrit.edu)

**PROCESS MODEL: AGILE METHOD**

**Abstract:** Software development which can be delivered fast, quick adaptation to requirements and collecting feedback on required information. The agile software methods and development is practices based approach empowered with values, principles and practices which make the software development process easier and in faster time. Agile methods which encompass individual methods like Extreme programming, Feature Driven Development, Scrum, etc are more coming into the commercial and academic worlds.

**Introduction**

Agility refers to the quality of being agile. Internet software industry and Mobile and wireless application development industry are looking for a very good approach of software development. Conventional software development methods have completely closed the requirements process before analysis and design process. As this approach is not always feasible and compatible with all other projects. In contrast to the conventional approaches, agile methods allow developers to make late changes in the requirement specification document.

The focus of the agile software development as given by “Agile Software Development Manifesto” is presented in the following:

* Individuals and interactions over processes and tools
* Working software over comprehensive documentation.
* Customer collaboration over contract negotiation
* Responding to change over following a plan

1. There is vital importance of communication between the individual who are in development team, since development centers are located at different places. The necessity of interaction between Individuals over different tools and different versions and processes is very vital.
2. The only objective of software development team is to continuously deliver the working software for the customers. New releases must be produced for frequent intervals. The developers try to keep the code simple, straight forward and technically as advanced as possible and will try to lessen the documentation.
3. The relationship between developers and the stakeholders is most important as the pace and the size of the project grows. The cooperation and negotiation between clients and the developers is the key for the relationship. Agile methods are using in maintaining good relationship with clients.
4. The development team should be well-informed and authorized to consider the possible adjustments and enhancements emerging during the development process.

**Agile methods**

Agile methods are designed to produce the first delivery in weeks, to achieve and early win and rapid feedback. These methods invent simple answers so that change can be less. These also improve design issues and quality as they are based on iteratively incremental method.

What makes a method an Agile?

When the process is:

1. Incremental: Small releases with rapid iterations
2. Cooperative: Customer and developer relationships
3. Straight: The method which is easy to learn and modify with documentation
4. Adaptive: Able to embrace changes instantly

Different **Agile Software Development methods**

1. Extreme programming
2. Scrum
3. Feature driven development

**Extreme programming**

The life cycle of Extreme programming consists of five phases

1. Exploration
2. planning
3. iteration on releases
4. Product ionizing
5. Maintenance
6. Death

Extreme programming is a light weight software methodology for a small to medium sized teams developing software in the rapidly changing requirements situation. XP matches the behavior of successful programmers in the wild

* + Tests
  + Refactoring
  + Evolutionary delivery
  + Incremental planning
  + Less overhead

**Basic principles:**

* Embracing change: change can come at any time. Model should be ready to take it.
* Rapid feedback: Time between action and feedback is less
* Assume simplicity : Initially assuming simplicity of any task and then proceeding according to the solution
* Incremental change
* Quality work

**SCRUM**

The term ‘SCRUM’ originally derives from a strategy in the game of rugby where it denotes “getting an out of play ball back into the game” with teamwork.

Scrum concentrates on how the team members should function in order to produce the system flexibly in a constantly changing environment.

Scrum is extremely simple model, used by different software companies from long time, which works with existing engineering practices and is scalable and work with common sense which is to say it is very easy.

Scrum process includes three phases

* Pre-game
* Development
* Post-game

**Pre-game phase** includes two sub-phases

Planning and Architecture design

*Planning* includes the development of the required system. A Backlog list is created, which contains all the requirements that are known at that moment. In every iteration the Back log list is updated by scrum team to gain commitment for the next iteration.

*Architecture phase*: In this phase an abstract view of the model is designed by viewing Backlog list.

**The Development phase**: This phase takes care of the different variable like time farm, quality, requirements, recourses, technologies and tools. The system is developed in Sprints. Sprints are the iterative cycles where functionality is developed or enhanced to produce new increments. Each Sprint includes the traditional phases of software engineering

1. Requirements
2. Analysis
3. Design
4. Evolution and delivery

**The Post-game phase** close to release.

* + Roles in Scrum
  + SCRUM master
  + Product owner
  + Scrum Team
  + Customer
  + User
  + Management

**Practices**

**Product Backlog** defines that is needed in the final product based on current know-how.

**Effort estimation** is the iterative process which is concentrated on Backlog list

**Sprint:** It is the procedure of adapting to the changing environmental variables.

**Feature Driven Development**

FDD is agile approach for software development systems. It provides enough structure for large items, emphasis on quality and working software, delivers frequent, tangible results.

Five phases of FDD

* Develop a overall model
* Build a features list
* Plan by feature
* Design by feature
* Build by feature

**Dynamic systems development method**

Dynamic systems development method is the framework of controls for the software development. While developing DSDM systems designers are interested in “edge case” rather than a mere project “ingredient”.

The principles involved in DSDM are

* Active user involvement is imperative.
* Teams must be empowered to make decisions
* Focus on frequent delivery
* Fitness for deliverable is criterion for accepted deliverables
* Iterative and incremental delivery is mandatory
* All changes during development must be reversible
* Requirements must be baseline at high-level
* Testing is integrated throughout the lifecycle
* Collaboration and co-operative approach

**Conclusions**

As we came to know that traditional software development approaches are more mechanistic which concentrate more on Processes, tools, contracts and plans. In contrast to traditional methods, agile methods keep emphasis on interaction, working software, embracing change at any moment of the project, customer relationships.

The method can be agile if it is:

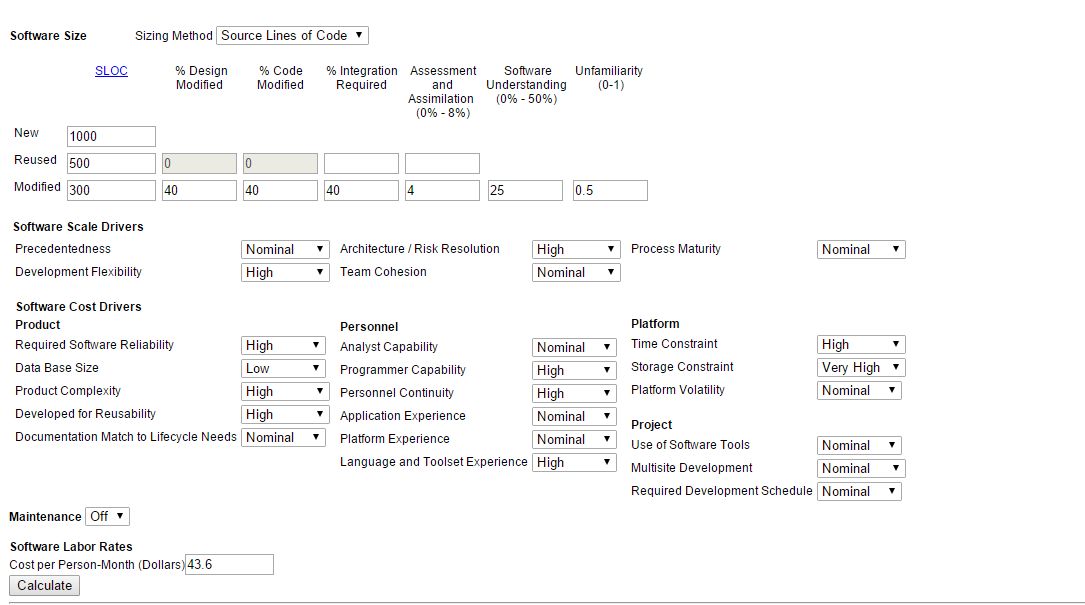
* Incremental
* Cooperative
* Straightforward
* Adaptive

“Agile view is more people centric rather than plan-centric.” Agile methods are not defined by a small set of principles, practices and techniques. It creates a strategic capability which has capability of responding to change, capability to balance the structure and flexibility, capability of innovation and creations through development team and uncertainty.

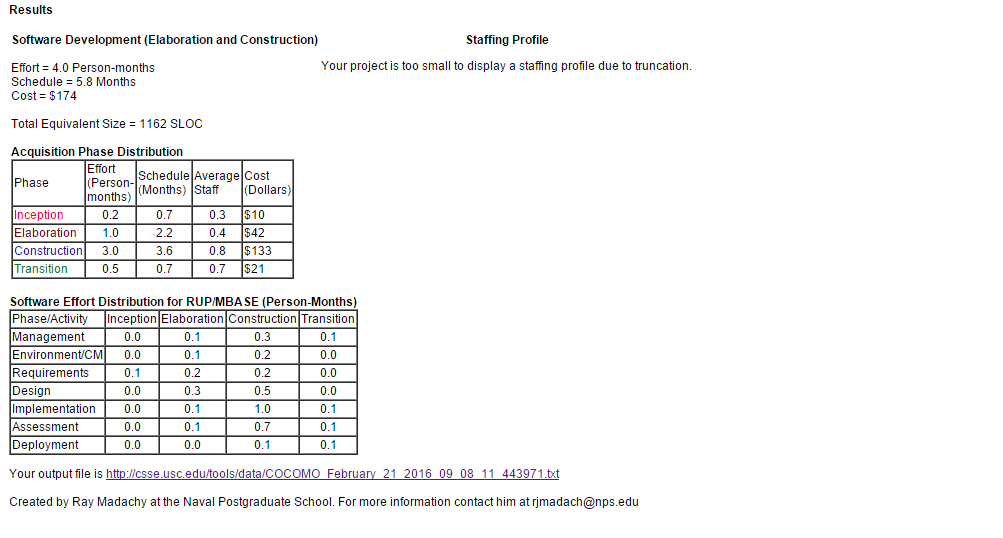
We have also discussed about different Agile Software development models such as

* XP (Extreme programming)
* Scrum
* FDD (Feature driven development)
* DSDM (Dynamic systems development method)

**EFFORT ESTIMATION USING COCOMO MODEL**



**Figure 3. Online COCOMO Model Page1**



**Figure 4. Online COCOMO Model Page2**

**RISK IDENTIFICATION AND MITIGATION**

**INTRODUCTION**

Risk is denned as an exposure to the chance of injury or loss. That is, risk implies that there is a possibility that something negative may happen. In the context of software projects, negative implies that there is an adverse eject on cost, quality, or schedule. Risk management is the area that tries to ensure that the impact of risks on cost, quality, and schedule is minimal. Risk Mitigation, within the context of a project, can be defined as a measure or set of measures taken by a project manager to reduce or eliminate the risks associated with a project. Risks can be of various types such as technical risks, monetary risks and scheduling-based risks. The project manager takes complete authority of reducing the probability of occurrence of risks while executing a project.

|  |  |
| --- | --- |
| **Possible Risk Factors That May Arise during Any Project’s Evolution** | |
| Life Cycle Stage | Possible Risk Factors |
| All | You or your team spends insufficient time on one or more stages. |
|  | Key information isn’t in writing. |
|  | You or your team moves to a subsequent stage without completing one or more of the earlier stages. |
| Starting the project | Some background information and/or plans aren’t in writing. |
|  | No formal benefit-cost analysis has been done. |
|  | No formal feasibility study has been done. |
|  | You don’t know who the originator of the project idea is. |
| Organizing and preparing | People unfamiliar with similar projects prepare your project plan. |
|  | Your plan isn’t in writing. |
|  | Parts of the plan are missing. |
|  | Some or all aspects of the plan aren’t approved by all key audiences. |
| Carrying out the work | People on the project team didn’t prepare the plan. |
|  | Team members who didn’t participate in the development of the project plan don’t review it. |
|  | You haven’t made an effort to establish team identity and focus. |
|  | You haven’t developed any team procedures to resolve conflicts, reach decisions, or maintain communication. |
|  | Needs of your primary clients change. |
|  | You have incomplete or incorrect information regarding schedule performance and resource expenditures. |
|  | Project-progress reporting is inconsistent. |
|  | One or more key project supporters are reassigned. |
|  |  |
| Team members are replaced. |  |
|  |  |
| Marketplace characteristics or demands change. |  |
|  |  |
| Changes are handled informally, with no consistent analysis of their effect on the overall project. |  |
| Closing the project | Project results aren’t formally approved by one or more project drivers. |
|  | Project team members are assigned to new projects before the current project is completed. |

**TOP 5 RISK ITEMS**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **RISK ITEM** | **RISK MITIGATION TECHNIQUES** |
| 1. | Developing the wrong software functions. | Early user manuals, User surveys and developing the software in iterations (prototyping). |
| 2. | Developing the wrong user interface. | Task Analysis, User Characterization, Scenarios and Prototyping. |
| 3. | Updating objectives. | Releasing the software in small increments. |
| 4. | Lack of technical and management skills. | Adequate training and also allocating tasks to people based on their capabilities and strengths, enhanced communication between team members |
| 5. | Use of non effective components. | Clearly understanding the requirements and analyzing the scenarios, requirements scrubbing, cost benefit analysis. |

|  |  |
| --- | --- |
|  |  |