A survey on software process models

Nandini AV and Nilita Anil Kumar

***Abstract*—A software process (also known as software methodology) is a set of related activities that helps in development of the software. These activities may involve the development of the software from the scratch, or, modifying an existing system. Any software development involves certain steps like requirement specification, designing and implementation, verification and validation and evolution. The difference between each process model is how these steps are executed. These process models also vary on basis on the sequence of execution of these steps and the iteration. The main objective of this paper compare the advantages and disadvantages of different software development models to help user select the model of choice according to the demand of situation**

***Index Terms*—Software Engineering, process Models, Software Development, Software Process Model, SDLC, software life cycle model**

I. Introduction

Software process is a complex process and it requires strategic planning and decision-making There are no ideal process, most organizations have developed their own software process. Any software process must include the following four activities:

Software specification (or requirements engineering):Define the main functionalities of the software and the constraints which needs to be taken care of while developing a software

Software design and implementation:This involves the designing and planning, and then implementing that design.

Software verification and validation: At this stage we check if the developed software matches the requirement criteria.

Software evolution (software maintenance):The timely modification to meet the user expanding user requirement.

In practice, they include sub-activities such as requirements validation, architectural design, unit testing, integration and many more. There are also supporting activities such as configuration and system management, quality assurance, project management, user experience feedback etc. When we talk about a process, we usually talk about the steps in it. A software process model is simple strategic representation of software development process. Each process is a representation of these activities in a different perspective without losing the essence.

II. Common Software Development Life Cycle Models

A. Waterfall Model

The waterfall model [1] is the classical model of software engineering.It is a linear sequential flow model. In this model the progress is seen as flowing steadily downwards through the phases of software implementation, hence its name waterfall model. A phase would start only if the previous phases has been completed.it does not follow the iterative pattern, The waterfall approach does not define the process to go back to the previous phase to handle changes in requirement, it's a one way flow. Each phase, a review takes place to keep a track if the project is on the right path . There is no phase overlap in this model.

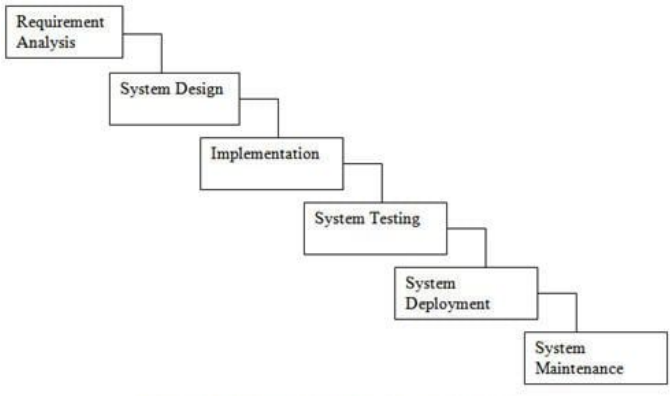

Fig. 1. Waterfall model

1) Usage of waterfall model: Waterfall mode is used when requirements are very well-known, clear and fixed. To use this model product definition should be stable. Technologies used should be well understood. There are no ambiguous requirements, there should be sufficient knowledge about domain and technologists model is used for small scale software development projects.

2) Advantages of waterfall model: Waterfall model is easy to explain to the users. The Stages and activities are well-defined hence less confusion. This model require clear planning and scheduling. Easy detection of errors/misunderstanding as there is verification at the end of each phase. Each phase has specific targets/deliverable(s).

3) Disadvantages of waterfall model: Waterfall model does not allow much revision. After the testing stage, it is very difficult to go back change something that was not well-documented or thought upon in the concept stage. So , it's not a good model for complex or object oriented objects and projects with changing requirements.

B. V-shape Model

Like waterfall model, the V-Shaped life cycle [2] is a sequential path of execution of processes. The V-model is a kind of SDLC model where process executes in a sequential manner in V-shape. It is less flexible. It requires a clear picture with few changes in beginning which is occasionally possible

Nandini AV, Nilita Anil Kumar are with Information Technology, National Institute Of Technology (e-mail: nandinivineeth0112@gmail.com). (e-mail: nilita.aniill@gmail.com).

in real life scenarios. Once an application is in the final stage that is the testing stage t, it is very difficult to go back and change something that was not well-thought-out in the concept stage. No working software is produced until late during the life cycle so it is tough to incorporate the changes in case customer wants any modification.. High amounts of risk and uncertainty ,hence it is not good for long, complex and object-oriented projects.

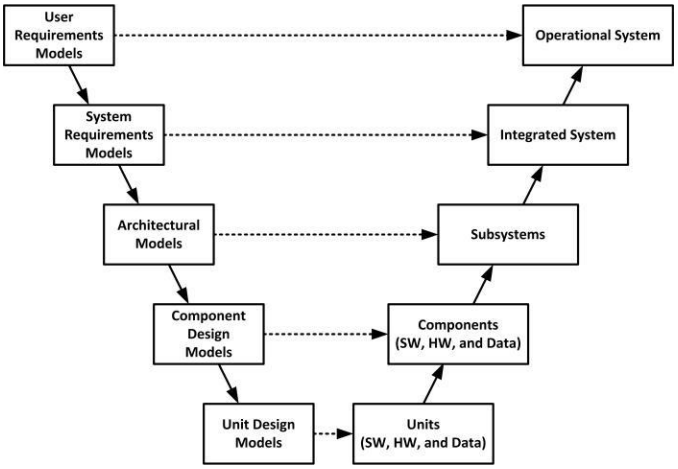

Fig. 2. V-shape model

V-shape model is also known as Verification and Validation model. This is a variation of waterfall model, it is based on the association of a testing phase for each corresponding development stage. Development of each step is associated with the testing phase. Like in waterfall model, the next phase starts only after completion of the previous phase i.e. for each development phase, there is a testing activity corresponding to it. But this model emphasizes the verification and validation of the product, testing of the product is planned in parallel with a corresponding phase of development.

Verification: it is the evaluation of product done to find whether specified requirements meet. It involves static analysis techniques done without executing code.

Validation: It involves dynamic analysis technique (functional, non-functional analysis). This testing done by executing code. It is the process to evaluate the software to determine whether software meets the customer expectations and requirements. The V-Model contains Verification phases on one side and the Validation phases on the other side. These two, Verification and Validation phases are joined by coding phase in V-shape.

1) Usage of V-shape model: The V-shaped model can be used for small or medium-sized projects. This can can be used when requirements are clearly defined prior to the commencement of project. This is suitable when ample technical resources are available with needed technical expertise.

2) Advantages of V-shape model: Non Complex structure hence easy to use.

Testing activities like planning, test designing happens well before coding and implementation. Hence, it is considered better than waterfall model. Proactive defect tracking — that is defects are found at early stage hence avoids the downward flow of the defects.

3) Disadvantages of V-shape model: The V-shaped model is very rigid and least flexible. Software is developed during the implementation phase, so no early prototypes of the software are produced hence working model is produced towards the end of the cycle.

C. Iterative Model

An iterative life cycle model [3] does not attempt to start with a full specification of requirements. This is a particular implementation of a software development life cycle that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete. The main feature of this is the modularity , this is a way of breaking down the software development process of a large application into smaller processes.

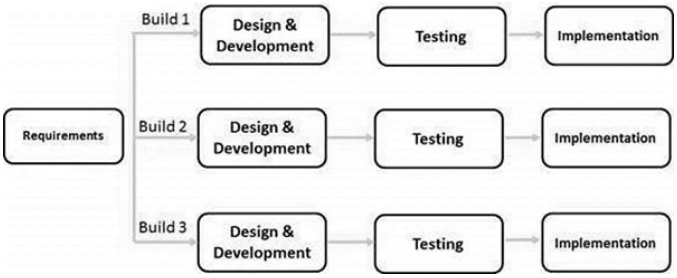

Fig. 3. Iterative Model

1) Usage of iterative Model : The requirements of the systems are defined and understood clearly. New technology is being used and is being learned by the development team. There are some high risk features and goals which might change in the future. There resources with needs skill sets are unavailable and are planned to be used on a contract basis for specific interactions.

2) Advantages of Iterative model: Implements during the earlier stages of the development process. Allows the team to find functional or design related flaws as early as possible. Easily adaptable to the ever-changing needs of the projects as well as the client. It is best suited for agile organizations. Parallel development can be planned.

3) Disadvantages of Iterative model: Implementation of iterative model require huge amount of resources.Progress of process is dependent on risk analysis. Defineing the increments require knowledge of complete systemNot suitable for small projects and where requirement changes .

D. Spiral Model

The spiral model [4] is similar to the incremental model, with more emphasis placed on risk analysis. In this model begins with a design goal and ends with the customer review and feedback. The development team starts with small modules and it goes through development phase. The functionalities are added in every-increasing spirals until the application is ready for the production phase.

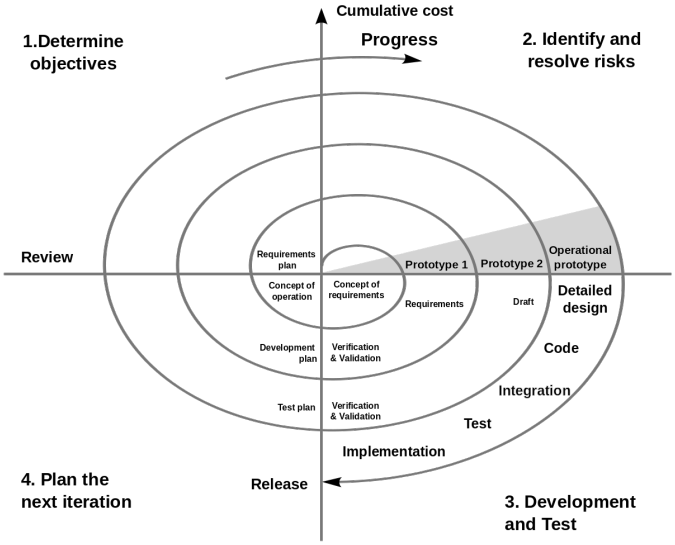

Fig. 4. Spiral model

1) Usage of Spiral model: Spiral model can be used when project is large and requirements are unclear and complex, when product releases are required to be frequent. When project is sensitive to risk and cost evaluation.in projects where prototyping can be done.

2) Advantages of spiral model: The main feature of Spiral model is that functionality or changes can be done at later stages. There is continuous development which helps risk management. Customer feedback can be integrated easily.

3) Disadvantages of Spiral model: Spiral model works best for large projects only, also demands risk assessment skills. This involves heavy documentation. Spiral model protocols need to be followed strictly.

E. Incremental Model

The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system [5]. Incremental Model is a process model where requirements are broken down into multiple small modules of complete software development cycle. These modules are standalone module. This model done in steps from analysis design, implementation, testing/verification, feedback. The Highest priority requirement is developed first. These modules are successively built to produce a final total system. Once the module is developed, requirement for that increment are frozen.

1) Usage of Incremental Model:: Incremental Model can be used when requirements of the system are clearly understood.When early release of a product is demanded.When development team is not very well skilled or trained. When projects involves high-risk features and goals.

2) Advantages of Incremental model:: The software development time is less . The changes in the requirement can be integrated easily. The system being flexible the Customer feedback is taken for each module and integrate

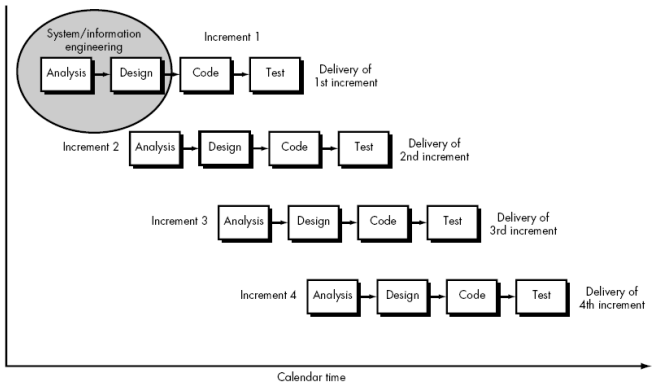

Fig. 5. Incremental model

3) Disadvantages of Incremental model:: Incremental model require good planning. Each iteration should be rigid and should not overlap with each other. Rectifying a problem in one unit requires correction in all the units and consumes a lot of time if the modules are not standalone.

F. Rapid prototyping model

A rapid prototype [6] is a working model that is functionally equivalent to a subset of the product. The major essence of this model is prototyping. Instead of freezing the requirement, a prototype is presented to the customer so that the customer can get the feel of how the product will look. A feedback is taken from the customer and integrated with the software requirement, which is used for development.

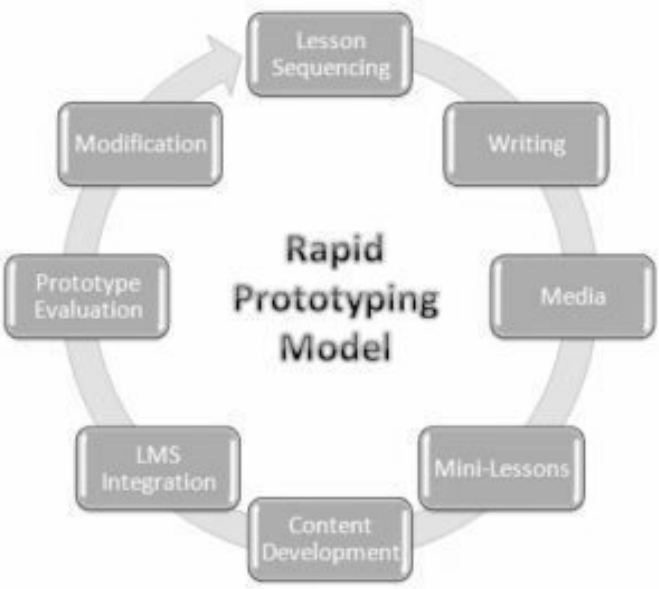

Fig. 6. Rapid prototyping model

1) Usage of Rapid prototyping model: Prototype model is used the product is highly user centric.It ensures that the user feedback is incorporated in the prototype to result in a useable system.It It is time consuming task ,for a system to be built

that allows ease of use and needs minimal training for the end user.

2) Advantages of Rapid prototyping model: Users are actively involved in the development ad gives feedback the users get a better understanding of the system being developed.risks can be detected much earlier. Missing functionality/requirement can be identified easily.

3) Disadvantages of Rapid prototyping model: Rapid prototyping model follows implementing and then repairing way of building systems.Hence, this might increase the complexity of system beyond the original plans. This is time-consuming as constant feedback are to be taken.

G. Agile Software Process(ASP)

The Agile Software Process (ASP) was first proposed at 1998 [7] unlike traditional software process models based on volume, the ASP is time-based and quickly delivers software products. It is an iterative and incremental model where the project is built in smaller chunks, not all at once and delivering them in short two weeks cycle called iterations. The self-organizing teams and customers or the end users take collaborative efforts to provide solutions to changing requirements.

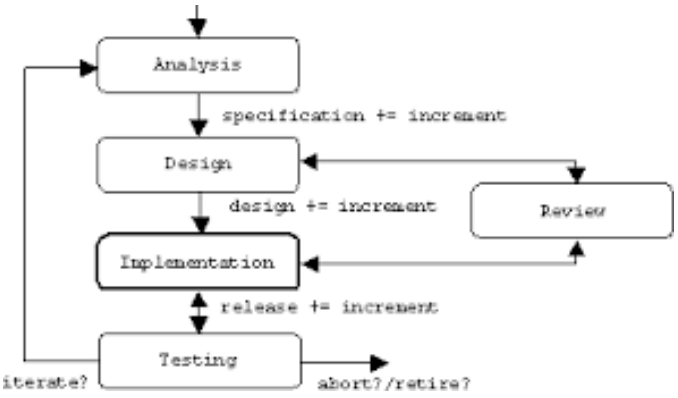

Fig. 7.

1) Usage of Agile Software Process: Agile Software Process is useful when new changes have to be implemented. They can be implemented at a very low cost because of the frequency of increments that are produced.

2) Advantages: The continuous involvement of customers and end users is more focused rather than the process and tools. Rapid satisfaction of customers by the delivery of the working software frequently(weeks rather than months. There is a regular adaptation to changing requirements, even the late ones. Face to face communication and daily cooperation among customers, developers and testers take place.

3) Disadvantages: It is difficult to assess the amount of effort that is required at the beginning if the software deliveries are large. Design and documentation are given less emphasis. If the customer is not clear of the final outcome they want, the project can be taken off track or the project can fail due to unrealistic expectations. Only senior programmers are allowed to take the decisions and not the newbie programmers.

4) Methodologies used in Agile Development Process:

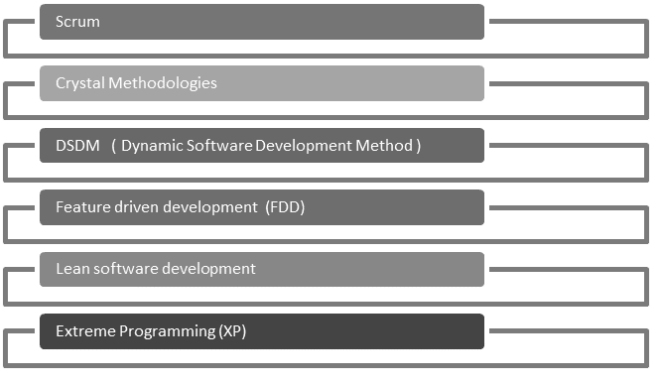

Fig. 8. Methodologies of Agile Development Process

a) Scrum: Scrum in software development means working with a small team and on a project management framework to detect the strengths and weaknesses of the project. A SCRUM is a Rugby team of eight individuals [8]. The team acts together as a pack to move the ball down the field. Teams work as tight, integrated units with a single goal in mind.
It consists of three roles, which are the product owner, scrum master and team. The product owner is the voice of the client. He/She is responsible for setting a goal for a sprint, designing and maintaining a flexible plan to build on it and deciding a release date. Scrum master helps in building a winning team, resolves problems during the project and ensures team productivity. The team includes the people who design, implement and fix the bug of the product.
A Particular set of features for each sprint is set by the product owner in the product backlog. The team then selects the top feature from the product backlog which they will deliver at the end of the sprint. After agreeing to do a particular feature, the task is broken down into smaller tasks that are put in the sprint backlog by the product owner. Then the sprint starts, for each sprint there is a daily meeting where the team member provides their task status and the problems related to the task and the scrum master is responsible for solving the problem. After the sprint is completed, the feature is presented and approval is taken from the product owner and reviews regarding the feature is given in the sprint review. The team, scrum master and the product owner discuss regarding the improvement in the overall process at the retrospective meeting.

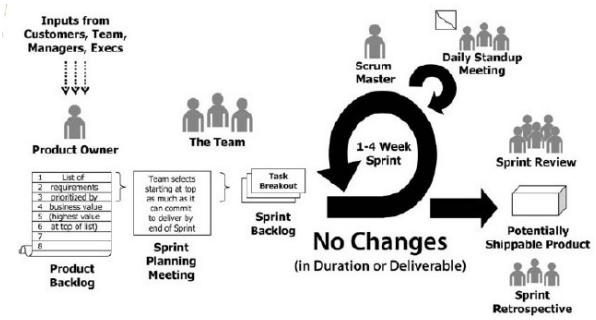

Fig. 9. Scrum Methodology

b) Crystal Methodologies: Crystal methodology focuses more on people and their interactions when working on a project rather than on the process and the tools. The Crystal family of lightweight SDLC methodologies is the creation of Alistair Cockburn [9]. People's skills and talents and the way they communicate has the greatest impact on the outcome of the project.

The core team includes the executive sponsor, lead designer, Ambassador user and a number of the system analyst. The first step includes chartering where the initial project plan is built, which includes the project map showing the project completion dates and iteration completion dates for delivery cycles. In cyclic delivery, each iteration lasts from one week to three months. It is then delivered to the user for feedbacks which are used for the improvement of the system and revising the plan or requirement. In the final step, the acceptance training is performed and the final product is prepared for the user. the final reflection is performed and the lessons learned are recorded.

c) Dynamic Software Development Method(DSDM): The Dynamic Systems Development Method (DSDM) is a framework [10] used to control software development projects with short timelines. It was created after project managers started using RAD which has an iterative way of working. The principles of DSSM focus on on-time delivery of real benefits to the business, aligned to clearly defined strategic goals, never compromise quality, built incrementally from firm foundations, active involvement of the users and decision-making power given to the team.

The techniques used for DSDM are time boxing, MoSCoW, and prototyping. Timeboxing technique makes sure that a certain is achieved at a certain interval but not more than 2,4 or 6 weeks. The MoSCoW rules are used to weight the importance of requirements so that the focus will be on to finish the most important features for the end product and completing the project within the budget. The rules followed are must have - all the features in this group must be implemented , if they are not , the system would simply not work , should have — features in this group are important but can be omitted due to time constraints, could have - features that enhance the system with functional items which can be assigned to a later timebox, want to have - features that serve a limited group of users or have little value. Prototyping is used to achieve two principles that are frequent delivery and incremental development.DSDM differentiates prototypes as a business prototype for assessment of the evolving system, usability prototype to check the user interface, performance prototype to ensure solution will deliver performance, capability prototype to evaluate possible options.

d) Feature Driven Development(FDD): Feature Driven Development (FDD) is a model-driven short-iteration software development process [11]. FDD is used when you are working in big cooperation on a large scale software project. This methodology relies heavily on chief developers and has a top-down approach. It is a five-step development project built

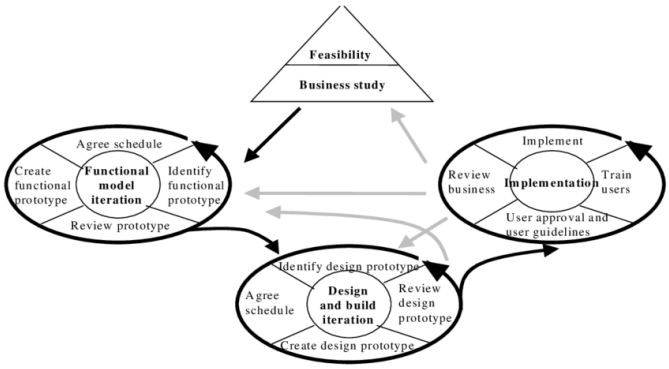

Fig. 10. Dynamic Software Development Method

around the discrete "feature" project. The project life cycle includes developing an overall model, building a feature list, planning, designing and building by feature.

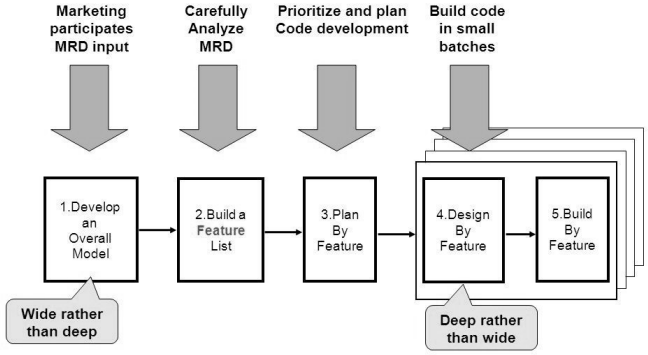

Fig. 11. Feature Driven Development

e) Lean Software Development: Lean Software Development is an iterative model developed by Mary and Tom Poppendieck, adapted from the Toyota manufacturing methodology. Its emphasis on optimizing efficiency and minimizing waste in the development of software. A lean team is a tool that refines on what the product actually should be.

The principles of lean software development include eliminating waste i.s anything that doesn't add value to the product, focus on learning, decide on features as late as possible to eliminate the need to redo the work as the market changes, delivery as early as possible, empowering the team, building integrity and optimizing the whole.

f) Extreme Programming(XP): Extreme programming is used to improve software quality. It is used when the customer is not sure about the functionalities of the system. It is used when the product has to be released in short cycles of 14 days called iterations.

The first phase of extreme programming is planning where the user creates stories with the development team. The user stories are converted into iterations. The combination of iterations provides the user with the final product. The second phase is analysis to prioritize the stories, define the budget

Fig. 12. Lean Software Development

and iteration span time. The third phase is design to break down the task and test scenario preparation for each task. The fourth is the execution phase which includes coding, testing, mid iteration review and end of iteration review. The fifth phase is wrapping for smaller releases and demos and reviews. The last phase is closure which includes production launch and production support.

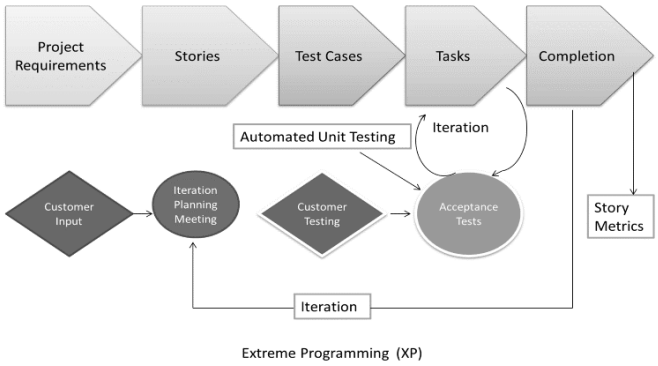

Fig. 13. Extreme Programmming

III. Conclusion

In this paper, different Software Development Life Cycles(SDLC) are studied and compared. Each of them have their own advantages and disadvantages. Selecting the correct life cycle model is extremely important as the software has to be delivered with the timeline with the desired quality. This study will make the process of selecting the correct SDLC easy and will be very effective for the software industry.

IV. References

[1] Royce, Winston, "Managing the Development of Large Software Systems", Proceedings of IEEE WESCON 26, 1970.

[2] Kevin and Harold , "The Relationship of System Engineering to the Project Cycle," in Proceedings of the First Annual Symposium of National Council on System Engineering, October 1991: 57–65.

[3] Craig and Victor R. ,"Iterative and Incremental Development: A Brief History". IEEE Computer (IEEE Computer Society) 36 (6): 47–56. :10.1109/MC.2003.1204375. ISSN 0018-9162. Retrieved 2009-01-10.], June 2003.

[4] B, "A Spiral Model of Software Development and Enhancement", ACM SIGSOFT Software Engineering Notes", "ACM", 11(4):14-24, August 1986.

[5] Craig and Victor , Iterative and Incremental Development: A Brief History, IEEE Computer, June 2003.

[6] C. Melissa McClendon, Larry Ragout, Gerri Akers: The Analysis and Prototyping of Effective Graphical User Interfaces. October 1996.

[7] International Conference on Software Engineering in Kyoto Japan (, 1998a).

[8] Linda Rising and Norman S., AG Communication Systems, "The Scrum Software Development Process for Small Teams, IEEE Software July/August 2000.

[9] Ernest , "About Software Engineering Frameworks and Methodologies", IEEE AFRICON 2009.

[10] : DSDM in a bird's eye view, DSDM Consortium, p. 3-8 (2001)

[11] , P., , E. & De Luca, J. (1999). Java Modeling In Color With UML: Enterprise Components and Process. Prentice Hall International. (ISBN 0-13-011510-X).