Software Project Models

Abhimanyu Chopra, Abhinav Prashar, Chandresh Saini

CSE Department, Dronacharya College of Engineering ,Gurgaon ,India
CSE Department, Dronacharya College of Engineering ,Gurgaon ,India
CSE Department, Dronacharya College of Engineering ,Gurgaon ,India
Email-abhinav.prashar@gmail.com, abbhimanyuchopra@yahoo.com, chandresh.saini@gmail.com

ABSTRACT: This research papers deals with significant issue in computer technology world. It is concerned with the software management processes that deals with the area of software development through the different development models, which are known as software project life cycle. There are mainly four development models i.e, waterfall, Iteration, V-shaped, spiral. These models have their own advantages and disadvantages. Therefore, the main aim of this research is to represent different software project models of software development and make a comparison between them to show the features and defects of each model.

1. INTRODUCTION

We all know the importance of computers in our life. In fact, computer has become indispensible in present time as it is used in different important fields such as industry, medicine, commerce, education and even agriculture. It has become an important element in the growth developing countries. Now a days, every organizations become more dependent on computer in their works as a result of computer technology. Computer is considered a time- saving device and its progress helps in executing complex, long, repeated processes in a very short time with a high speed. Computers are also used by people for fun and entertainment. The number of companies that produce software programs for the purpose of facilitating works of offices, administrations, banks, etc, has increased recently which results in the difficulty of enumerating such companies. From past four decades we have seen that software has been developed from a tool used for analyzing information or solving a problem to a product in itself. However, the early programming steps have created problems turning software into an obstruction to software development particularly those relying on computers. Software consists of documents and programs that contain a collection that has been established to be a part of software engineering procedures. The main aim of software engineering is to develop a appropriate work that constructs programs of high quality.

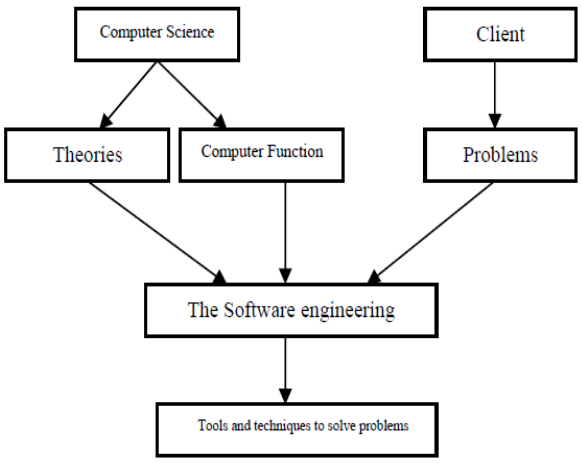


Fig. 1 Explanation of software engineering conception.

[1]

2. SOFTWARE PROCESS MODELS

Software process model is a description of the sequence of activities carried out in an SE project, and the relative order of these activities. A software process model is an abstract representation of a process. It provides a fixed generic framework that can be tailored to a specific project. It presents a description of a process from some particular perspective as:

1. Specification.
2. Design.
3. Validation.
4. Evolution. [2]
5. Different models

In software Engineering we have 8 project lifecycle models:-

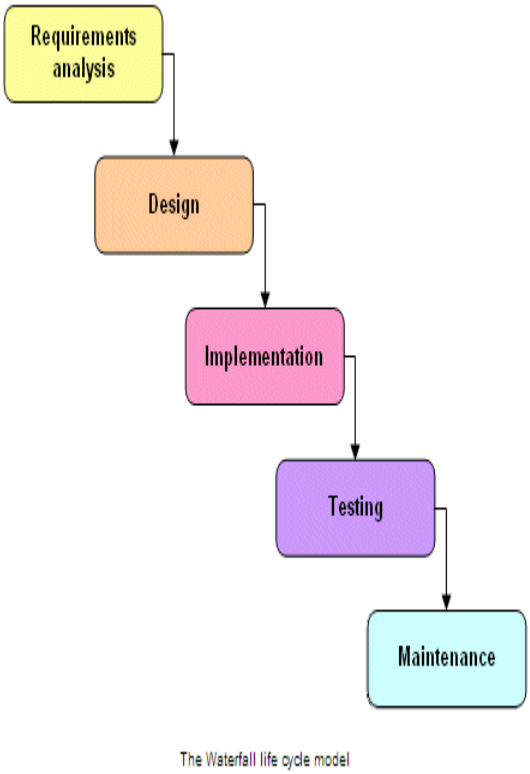
1. Waterfall model: Separate and distinct phases of specification and development.
2. Prototype model.
3. Rapid application development model (RAD).
4. Evolutionary development: Specification, development and validation are interleaved.
5. Incremental model.
6. Iterative model.
7. Spiral model.
8. Component-based software engineering: The system is assembled from existing components.

In his research will review the following four models:

1. Waterfall model.
2. Iteration model.
3. V-shaped model.
4. Spiral model.

4. WATERFALL MODEL

The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Analysis, Design, Production / Implementation, Testing, and Maintenance. The waterfall model is the classic process model – it is widely known, understood and used. In some respect, waterfall is the ”common sense” approach. [3] The Waterfall Model is used in the manufacturing and construction industries; highly structured physical environments in which after-the-fact changes are prohibitively costly, if not impossible.



The Waterfall life cycle model

[4]

WATERFALL MODEL PHASES:

1) Requirement analysis: Every possible requirements of the system is analysed and captured in this phase. The requirements are recorded in a document named SRS.

2) System Design: The requirements recorded in the first phase are studied in this phase and design of the system is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

3) Implementation: Output from the previous phase is considered as input for this phase, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

4) Testing: All the small program units developed in the implementation phase are joined to form a system, after testing of each unit. Post integration the entire system is tested for any faults and failures.

5) Maintenance: The last phase of waterfall Model is maintenance. There may be some issues

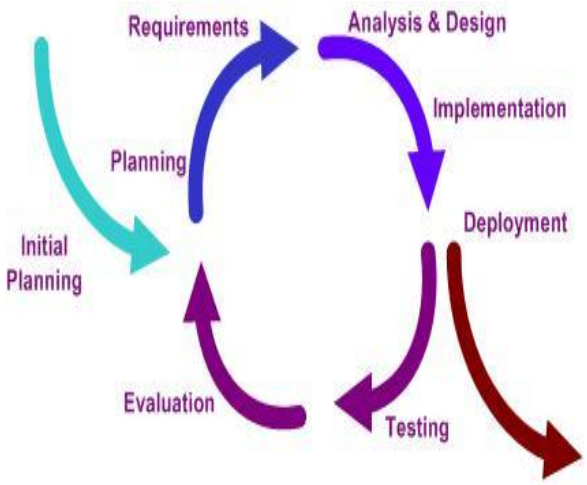
Which come up later in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Advantages	Disadvantages
1). Easy to understand and implement.	**1)** Doesn't reflect iterative nature of exploratory development.
2) Widely used and known	**2)** Sometimes unrealistic to expect accurate requirements early in a project
3) Fits other engineering process models: civil, mech etc.	**3)** Software is delivered late, delays discovery of serious errors.
4) Identifies deliverables and milestones	**4)** No inherent risk management
5) Easy to arrange tasks	**5)** Difficult and expensive to change decisions.
6) Process and results are well documented	**6)** It is difficult to measure progress within stages**.**
7) Works well for smaller projects Where requirements are well understood.	**7)** Significant administrative overhead, costly for small teams and projects.

5. ITERATION MODEL

Iterative Model is a combination of both iterative design or iterative method and incremental build model for development. The combination is of long standing and has been widely suggested for large development efforts. The relationship between iterations and increments is determined by the overall software development methodology and software development process. The exact number and nature of the particular incremental builds and what is iterated will be specific to each individual development effort.[5]

The Iteration Model



[6]

An iterative lifecycle model consists of repeating the following four phases in sequence:

1) **Requirements phase:** In this phase the requirements for developing the software are collected and analyzed. Iteration should eventually result in a requirements phase that produces a complete and final specification of requirements.
2) **Design phase:** In the design phase the software solution to fulfill the requirements is designed. This can be a new design, or a new version of an earlier design.
3) **An Implementation and Test phase:** In this phase the software code is generated, integrated and tested.
4) **A Review phase:** This is the last phase in which the software is evaluated, the current requirements are reviewed, and changes and additions to requirements proposed.

Advantages	Disadvantages
1) Generates working software quickly and early during the software life cycle.	**1)** Each phase of an iteration is rigid and do not overlap each other.
2) More flexible – less costly to change scope and requirements.	**2)** Problems may arise pertaining to system architecture because not all requirements are gathered up front for the entire software life cycle.
3) Easier to test and debug during a smaller iteration.	
4) Easier to manage risk because risky pieces are identified and handled during its iteration.	
5) Each iteration is an easily managed milestone.	

6. V-Shaped Model

The V**-** Shaped software development model is considered an extension of the waterfall model. Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape. The V-Model describes the relation between each phase of the development life cycle. The high-level design phase focuses on system architecture and design. An integration test plan is created in this phase in order to test the pieces of the software systems ability to work together. However, the low-level design phase lies where the actual software components are designed, and unit tests are created in this phase as well. [7]

The V-Shaped Model

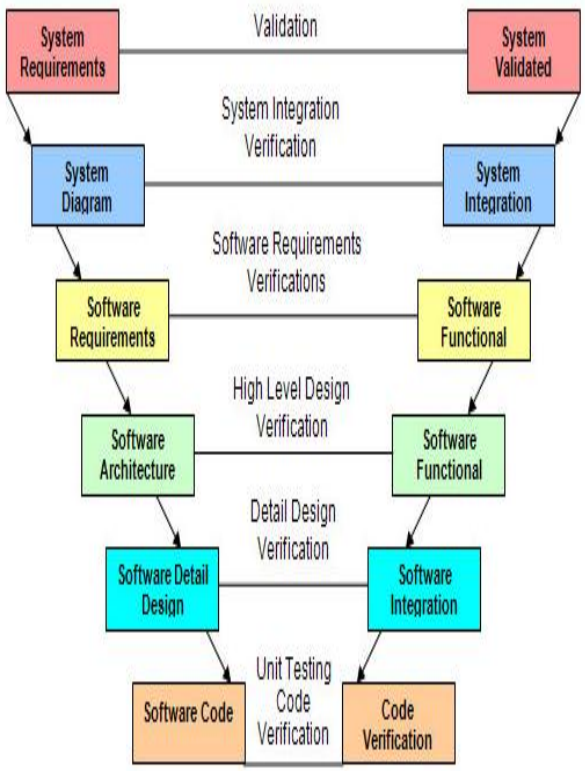


Figure 4 - V-Shaped Model

[8]

Following are the important phases of the v-shaped model:

1) **Project and requirements planning**: In this phase we determine the system requirements and how different available resources of the organization will be allocated to meet them.
2) **Product requirements and specification analysis**: This phase is used to do analysis of the software problem and conclude with a complete specification of the expected external behaviour of the software system to be built.
3) **Architecture or high-level design:** This phase defines how the software functions are to implement the design.
4) **Detailed design:** This phase defines and records algorithms for each component that was defined during the architecture phase. These algorithms will be translated into code.
5) **Coding:** Transforms the algorithms defined during the detailed design phase into software.
6) **Integration and testing:** This phase acts as an interconnection the sets of previously unit-tested modules to ensure that the sets behave as well as the independently tested modules did during the unit-testing phase.
7) **System and acceptance testing:** This phase is used to check whether the fully integrated software system rooted in its actual hardware environment behaves according to the software requirements specification.
8) **Acceptance testing:** This last phase of the V-shaped model allows the user to test the

functionality of the system against the real requirements. After this final testing, the software and its surrounding hardware become operational. Maintenance of the system follows.

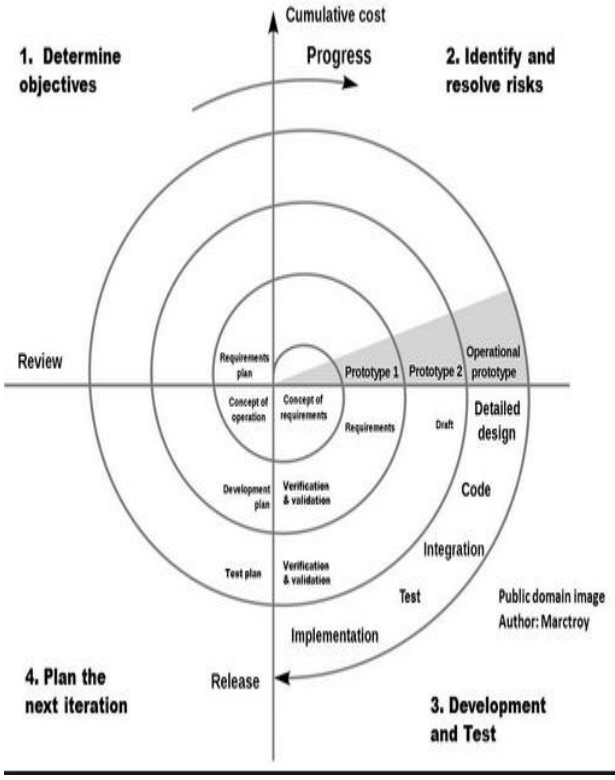
Advantages	Disadvantages
1) The model encourages verification and validation of all internal and external deliverables, not just the software product.	**1)** It does not easily handle concurrent events.
2) The V-shaped model encourages definition of the requirements before designing the system, and it encourages designing the software before building the components.	**2)** It does not handle iterations of phases.
3) It defines the products that the development process should generate; each deliverable must be testable.	**3)** The model is not equipped to handle dynamic changes in requirements throughout the life cycle.
4) It enables project management to track progress accurately; the progress of the project follows a timeline, and the completion of each phase is a milestone	**4)** The requirements are tested too late in the cycle to make changes without affecting the schedule for the project.
5) It is easy to use (when applied to a project for which it is suited).	**5)** The model does not contain risk analysis activities.

7. SPIRAL MODEL

The spiral model is a risk-driven process model generator for software projects. Based on the unique risk patterns of a given project, the spiral model guides a team to adopt elements of one or more process models, such as incremental, waterfall, or evolutionary prototyping. Extends waterfall model by adding iteration to explore/manage risk Project risk is a moving target. Natural to progress a project cyclically in four step phases

1. Consider alternative scenarios, constraints
2. Identify and resolve risks
3. Execute the phase
4. Plan next phase: e.g. user req, software req, architecture then goto 1. [9]

The Spiral Model



10]

The spiral model has following important phases:

1. **Determine objectives, alternatives and constraints:**
   - System requirements are defined in detail and include artefacts for functionality, performance, hardware/software interfaces, key success metrics, etc.
   - Constraints such as cost, schedule and interfaces are addressed

2. **Identify and resolve risks, evaluate alternatives:**
   - All possible and available alternatives for developing a cost effective project are evaluated and strategies developed to determinate their use
   - Identify and resolve all the possible risks in the project such as lack of experience, new technology, tight schedules, poor process, etc.
   - Resolve any found risks and uncertainties. Subset reviews may be commissioned to investigate other process models until all high risk situations are resolved.

3. **Development and test:** Prototype the system from the preliminary design. Follow the usual pattern of create and review design, code, inspect code and test.[11]

4. **Plan the next iteration:**
   - Review the first prototype for strengths, weaknesses, and risks
   - Elicit the requirements for the second prototype
   - Plan and design the second prototype;
     (i) Create the project plan

(ii) Document the configuration management plan
(iii) Construct a test plan
(iv) Devise an installation plan. [12]

Advantages	Disadvantages
1) High amount of risk analysis	**1)** Can be a costly model to use.
2) Good for large and mission-critical projects.	**2)** Project's success is highly dependent on the risk analysis phase.
3) Software is produced early in the software life cycle.	**3)** Doesn't work well for smaller projects.
4) Development can be divided into smaller parts and more risky parts can be developed earlier which help in better risk management.	**4)** Risk analysis requires highly specific expertise.

[13]

8. CONCLUSION

After finishing this research work we conclude that:

1. Many existing project models for developing systems are used based on the sizes of projects and requirements.
2. Most of these models were established between 1970 and 1999.
3. The two most commonly used models for project development are Waterfall model and spiral model.
4. Each and every model has its own advantages and disadvantages for the development of systems, so each model tries to eliminate the disadvantages of the previous model.

9. REFRENCES

[1]. www.ijcsi.org/papers/7-5-94-101.pdf

[2]. www.nada.kth.se/~karlm/mvk/mvk08_lec2.pdf

[3]. www.nada.kth.se/~karlm/mvk/mvk08_lec2.pdf

[4]. http://www.technologyuk.net/computing/sad/waterfall_model.shtml

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

[6]. http://en.wikipedia.org/wiki/Iterative_and_incremental_development

[7]. www.ijcsi.org/papers/7-5-94-101.pdf

[8]. http://www.antrix.com/services/service_list/11_validation_computersystem

[9]. www.nada.kth.se/~karlm/mvk/mvk08_lec2.pdf

[10]. http://ultimatesdlc.com/spiral-model/

[11]. http://ultimatesdlc.com/spiral-model/

[12]. http://ultimatesdlc.com/spiral-model/

[13]. http://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm