A Review of Software Process Models and Methodologies

Mr. Anand K Pandey (737)
Asst Prof, MCA Dept, ITM Gwalior
anand_pandey23@rediffmail.com

Mrs. Rashmi Pandey (1216)
Lecturer in Comp. Application Dept,
IASCA, Gwalior reshu_tanu@yahoo.com

ABSTRACT
This paper focuses on describing the most representative software process models, which are analyzed and compared with respect to their strengths and weaknesses. Now a days software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software products. Software process models come and go through a series of passages that account for their inception, initial development, productive operation, and retirement from one generation to another. A systematic study of the software process models will require a vocabulary in which various process models, methods and approaches can be described, compared and contrasted. These models help to manage the various tasks distributed in different phases of development. Different software process networks can be viewed as representing multiple interconnected task chains. This work reviews the concept of software development methodology which integrates the organizational model as a component with static and dynamic elements of the software process.

KEYWORDS
Framework, software process, software project, Software products, Process models, environment, lifecycle.

1. INTRODUCTION
A software process model is an abstract representation of process. It present a description of a process from some particular perspective. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process [13]. Extending this to the specific case of software, we can say that a software process is a method of developing and producing or reproducing the software. Again, this is nothing new, every process produces something. A process model provides a harness into which the technical activities and the supporting tools can fit. Software process models, range from simple traditional models to advanced software process models suited for safety and mission critical systems. A software process model is either a descriptive or prescriptive characterization of how software is or should be developed. A descriptive model describes the history of how a particular software system was developed [2]. Descriptive models may be used as the basis for understanding and improving software development processes or for building empirically grounded prescriptive models. A prescriptive model prescribes how a new software system should be developed. Prescriptive models are used as guidelines or frameworks to organize and structure how software development activities should be performed, and in what order. Mostly all the software processes focuses on answering three basic questions: Why, how and what can be done to engineer or reengineer the solutions [8]. In this paper we have introduced the essential concept of software process model with the help of different need and role of software process model, its process framework, concept of software methodology and comparative study of different process models.

In contrast to software life cycle models, software process models often represent a networked sequence of activities. A process model is a general process specification which has been found useful in many circumstances [4]. Software process networks can be viewed as representing multiple interconnected task chains. Task chains represent a non-linear sequence of actions that structure and transform available computational objects (resources) into intermediate or finished products [7]. The process models are more descriptive and prescriptive than life cycle model. Our main object and purpose to write this paper is to discuss a window view involved in software process models and familiarize the computer science students, software professionals and users with concept of software development methodology, different need and role of process models, strength and weakness of the different models.

2. NEED AND ROLE
Software is everywhere, from cell phones to large military systems.
According to the National Academy of Science, the software process models often represent a networked sequence of activities, objects, transformations, and events that embody strategies for accomplishing software evolution. Such models can be used to develop more precise and formalized descriptions of software process activities. Their power emerges from their utilization of a sufficiently rich notation, syntax, or semantics, often suitable for computational processing [10]. System development with components is focused on the identification of reusable entities and relations between them, beginning from the system requirements and from the availability of components already existing.

Software processes models play an important role to provide efficiency, accuracy and discipline nature of software products. A role makes it possible to adapt the vision of an object in an process step i.e, which specify types of the product model are able to be adapted to the process execution context using the define role. Thus a process becomes a list of roles where each role is customized in order to satisfy the process requirements. That is then properties and behavior of an object are characterized in the process execution context [14]. Each
role has methods which are used to adjust the behavior and mechanism of the original object to the execution context. That is, a role can redefine the original methods or define new ones in order to customized object behavior and process model for the context where the object is used or model is applied. Users are assigned to roles and roles are associated with permissions that determine what operations a user can perform on information objects acting as a role member. Additionally, various kinds of constraints, hierarchical structures, and other attributes of roles can be specified. Although process models are important in providing overall guidance, they are seldom followed exactly. Developers often have different, personal views on what roles they are supposed to play.

A software process model that can be adapted to your organization's specific project needs is known as Adaptable Process Model (APM). The APM is intended as a basis from which a customized software process can be developed. The complete Adaptable Process Model (APM) is provided for informational purposes and for assessment by potential users [12]. The software process model is used to provide you with different software process that you can customize and adapt to system or project needs.

We can categorize the different role of Software Process Models as follows:

(i) Descriptive Role:
- Track what actually happens during a process.
- Takes the point of view of an external observer who looks at the way a process has been performed and determines the improvements that have to be made to make it perform more effectively or efficiently.

(ii) Prescriptive Role:
- Defines the desired processes and how they should/could/might be performed.
- Lays down rules, guidelines, and behavior patterns which, if followed, would lead to the desired process performance. They can range from strict enforcement to flexible guidance.

(iii) Explanatory Role:
- Provides explanations about the rationale of processes.
- Explore and evaluate the several possible courses of action based on rational arguments.
- Establish an explicit link between processes and the requirements that the model needs to fulfill.
- Pre-defines points at which data can be extracted for reporting purposes.

3. SOFTWARE PROCESS VERSUS LIFE CYCLE
The software process has assumed its own identity and has been the subject of investigation and research. Software process modeling describes the creation of software development process models using a specific life cycle. Software process evaluation judges and decide on the quality of the software process of a given organization, and may propose a process improvement strategy. A process model can be analyzed, validated and simulated, if executable. As shown in figures.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Criteria</th>
<th>Life Cycle</th>
<th>Software Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Focused on</td>
<td>Product Output in Each Phase</td>
<td>Activity performed during software development</td>
</tr>
<tr>
<td>2.</td>
<td>Relationships</td>
<td>The state in transformed by the activate</td>
<td>The activity works on a state</td>
</tr>
<tr>
<td>3.</td>
<td>Generality</td>
<td>Multiple life Cycle models</td>
<td>Single model for the developer organization</td>
</tr>
<tr>
<td>4.</td>
<td>Domain Scope</td>
<td>Specific, project dependent</td>
<td>General Project independent, organization dependent.</td>
</tr>
<tr>
<td>5.</td>
<td>Process Evaluation</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 1.1 Difference Between Life Cycle and Software Processes

4. CONCEPT OF SOFTWARE DEVELOPMENT METHODOLOGY
A methodology is composed of one of the software development models used in conjunction with one or more techniques, i.e., methodology = model + technique(s). The techniques of processes are ways to implement the models [3].

A software development methodology or system development methodology in software engineering is a framework that is used to structure, plan, and control the process of developing an information system.

Software engineers practices for using selected process techniques to improve the quality of a software development effort by a specified methodology. This is based on the assumption, subject to endless debate and supported by patient experience, that a methodical approach to software development results in fewer defects and, therefore, ultimately provides shorter delivery times and better value. The documented collection of policies, processes and procedures used by a development team or organization to practice software engineering is called its software development methodology (SDM) or system development life cycle (SDLC) [4]. The original movement to try to change this introduced the notion of methodology. These methodologies impose a disciplined process upon software development with the aim of making software development more predictable and more efficient. They do this by developing a detailed process with a strong emphasis on planning inspired by other engineering disciplines - which is why I like to refer to them as Engineering Methodologies. Engineering methodologies have been around for a long time. They've not been noticeable for being terribly successful. They are even less noted for being popular. The most frequent criticism of these methodologies is that they are bureaucratic. There's so much stuff to do to follow the methodology that the whole pace of development slows down.

Moving through the specified methodology, we might encounter the following major steps:
• Project charter and business case
• Definition of the business process and business requirements
• Documentation of user, functional and system requirements
• Top level architecture, technical approach, and system design
• System decomposition into component and unit specifications and design
• Coding, unit test planning, and unit test
• Generation of test data for unit testing and system testing
• System integration and testing
• Implementation, delivery and cut-over
• Training and user support
• System upgrades and routine software maintenance

5. COMPARISON OF DIFFERENT SOFTWARE PROCESS MODEL
We know that each process model is suitable for some context, and the main reason for studying different models is to develop the ability to choose the proper model for a given project. Using a model as the basis, the actual process for the project can be decided, which hopefully is the optimal process for the project [2]. Here we summarize the strengths and weaknesses of the different process models as follows.

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Task</th>
<th>Waterfall</th>
<th>V Model</th>
<th>Incremental</th>
<th>Spiral</th>
<th>Prototype</th>
<th>RAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Well defined Required</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Domain Knowledge of Team Member</td>
<td>Adequate</td>
<td>Adequate</td>
<td>Adequate</td>
<td>Very</td>
<td>Less</td>
<td>Adequate</td>
</tr>
<tr>
<td>3</td>
<td>Expertise of User in Problem Domain</td>
<td>Very Low</td>
<td>Very Low</td>
<td>Adequate</td>
<td>Very</td>
<td>Less</td>
<td>Adequate</td>
</tr>
<tr>
<td>4</td>
<td>Availability of reusable Components</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Users involvement in all phases of SDLC</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Complexity of System</td>
<td>Simple</td>
<td>Complex</td>
<td>Complex</td>
<td>Complex</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Higher Cost</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

Table 1.2 Comparison of Software Process Model

6. STRUCTURE ORIENTED VS OBJECT ORIENTED METHODOLOGY
A large number of methodologies have been proposed in software engineering to represent the requirement of application domain. The models developed using these methodologies help the software engineers and all other persons involved in a project to understand different aspects of the software development and hence often are called the software blueprints [3]. There are two well known methodologies, These are Structure Oriented Methodologies and Object Oriented methodologies.

6.1 STRUCTURE ORIENTED METHODOLOGIES
Models proposed under this category focus on the information structures and the relationship between the structures and semantic data models. These models capture the static view of the problem domain, and it emerged as a result of two properties data independence and data abstraction. Data independence says that model should be independent of implementation detail where as abstraction provides techniques for representing the system details in a organized manner. Structure Oriented software model illustrate the organization of procedural method of programs or projects. Each thread of execution begins with a root module at the top of an inverted tree of called modules. When project become large, functional branches of a tree get pushed down to a child diagram creating a linked stack of diagrams that highlight the project’s structure. Such as ER model, Waterfall model, Spiral model, V model, Incremental model, Prototype model, RAD model.

6.2 OBJECT ORIENTED METHODOLOGIES
Object Oriented software technology is likely the most active area of research today. An Object Oriented software model shows object instances, their operations and messages between objects and their building blocks to document the mechanisms within an object-oriented design [5]. This mechanism illustrates the design with a collection of communicating objects. Such as component based model.

7. EARLIER RESEARCH IN SOFTWARE ENGINEERING
Many sciences have good explanations of their research strategies. These earlier research explanations include summary for simplified views for the public and other observers [9]. In addition to the ongoing interest, debate, and assessment of process-centered or process-driven software engineering environments that rely on process models to configure or control their operation, there are a number of promising avenues for further research and development with software process models. These opportunities areas and sample direction for further exploration include.

• Research on Software Reliability: The Software Reliability Research group studies how program analysis, program verification and software measurement techniques can be used to improve the quality of software.
• Software process simulation: This model efforts which seek to determine or experimentally evaluate the performance of classic or operational process models using a sample of alternative parameter configurations or empirically derived process data. Simulation of empirically derived models of software evolution or
evolutionary processes also offer new avenues for exploration. This research describe an approach and experiences in developing and applying simulation and modeling technologies to software processes. The focus of this research is organized around three topics for software process simulation and modeling.

- **Software process and business process reengineering:** The business process reengineering (BPR) method is described by M. Hammer and Champy as the fundamental reconsideration and the radical redesign of organizational processes, in order to achieve drastic improvements of current performance in cost services and speed. BPR research give an idea about the factor that limits organization’s performance is the ineffectiveness of its processes [6]. BPR research derives its existence from different disciplines, and four major areas can be identified as being subjected to change in BPR - organization, technology, strategy, and people.

- **Web-based software process models:** A web based process model is described by Internet society in 1989 for use of software engineering and its model world wide. The web based process models uses extreme programming and Web-based virtual software development enterprises or workspaces. Web based engineering actively promotes systematic, disciplined and quantifiable approaches towards successful development of high-quality, ubiquitously usable Web-based systems and applications.

- **Understanding, capturing, and operationalizing process models:** This model is invented by J. Beck in 1999 for understanding, capturing and operationalizing the different requirements of processes. This model characterize the practices and patterns of globally distributed software development associated with open source software, as well as other emerging software development process.

**CONCLUSION**
Software Process models are processes of the same nature that are classified together into a model. These models have general approaches for organizing a project into activities. The main work of this paper is to discuss the issues which is involved in software development process model which , must account for software the interrelationships between software products and production processes as well as for the roles played by tools, need of people and their workplaces. These models are used to improve the quality and productivity at low cost in less time of software products. These models are also describes the range of expected results that can be achieved by following the processes. The comparative study of different models help us to choose best process model according to our requirements.

**FUTURE SCOPE**
The process model will cover all functions of the organization, both operational/customer-facing functions and support/back-office functions. It describes a lifecycle that can be used for successful software development. The main function of a process model is to establish the order in which a project specifies, implements, tests, and performs its activities. The next generation of geographic information systems (GIS) will be driven by process models. These are usually composed of algorithms and heuristics that will act on users requests. In future process models will represent an evolution from these existing technologies, catalyzed by artificial intelligence that takes traditional GIS operations into the world of dynamic, proactive computing on a semantic Web of interconnected data and intelligent software agents.

**REFERENCES**


[14] http://www.springerlink.com/content/5157ht6w70876352

Proceedings of the 4th National Conference; INDIACOM-2010
Write your paper title here in title case