Chapter 5

Investigating and Analyzing the Desired Characteristics of Software Development Lifecycle (SDLC) Models

5.1 Introduction

Several Software Development Lifecycle Models (SDLCs) are in existence (18, 19, 20). Over the time different people have proposed different models to meet the industrial demands. Any SDLC process model should be a repeatable, clearly documented, highly effective and must be based on the best industrial practices. However, the traditional SDLC process models provide very insightful theory and helpful best practices, but do not provide the practical details for daily application. As a result, statistics (62) shows that SDLC process models are rarely used by organizations for the purpose they are designed and developed for. Another primary reason for not using these models is due to lack of their suitability for real life projects - which led to software crisis. While investigating the reasons for unsuitability of these models, it is identified that we lack well defined characteristic parameters for any SDLC model. Without applying the process model in real project, we do not have adequate metric to analyze the suitability and goodness of such models. Davis et al (105) has proposed a strategy long back in 1988 for comparing alternative SDLCs only based on the ability to satisfy user needs and reduced life cycle cost. In this work, we have investigated, identified and analyzed the features of any

Based on the publication in the "*International Journal of Advanced Research in Computer Science and Software Engineering(IJARCSSE)*", ISSN: 2277 128X, 2(3), 311–315, 2012.

SDLC process model in general which may further be used for characterizing any SDLC process model for its suitability.

5.2 Objectives and Goal

The objective of this work is to identify the different characteristics of a good software development lifecycle model. Given these characteristics, one can judge, evaluate, predict and select the best SDLC model suitable for real projects. The outcome of this work then can be used while designing and developing new process models too. Using such metric, one may evaluate a model for its suitability, applicability and predictability of success for any project. Moreover, these can be used to design the quality, suitability and predictability metric of any process model. The outcome of this research may further be used to develop new SDLC models according to the need of the industry and even may be used while designing any new process model. Hence, the goal of this work is to develop the foundations for SDLC metric.
We shall use the term software development lifecycle process model and process model over the paper interchangeably.

5.3 SDLC Process Models and Objectives

There are numerous examples of disasters that had been caused by software failures. As the computerization of the society continues, the public risks of poor quality software will become untenable unless orderly steps are taken to improve the software processes (106). Any SDLC process model has three primary business objectives (107):

- Ensure the delivery of high quality systems,
- Provide strong management controls over the projects, and
- Maximize the productivity of the systems development team.

Further, these objectives can be broadly categorized from the following two perspectives:

1. **The Technical Perspectives :** While building a system, there remain many technical activities and issues including system definition (analysis, design, coding), testing, system installation (e.g., training, data conversion), production support (e.g., problem management), defining releases, evaluating alternatives, reconciling information across phases and to a global view and defining the project's technical strategy etc. to be resolved.

2. **The Management Perspectives :** When we plan to develop, acquire or revise a system, we must be absolutely clear with the objectives of that system. The objectives must be stated in terms of the expected benefits that the business expects from investing in that system. The objectives should exhibit the expected return on investments. To achieve the project objectives and goal many management related issues have to addressed and resolved. The primary management activities includes setting priorities, defining objectives, project tracking and status reporting, change control, risk assessment, stepwise commitment, cost/benefit analysis, user interaction, managing vendors, post implementation reviews, and quality assurance reviews etc.

All the above objectives irrespective of technical or managerial, has to be achieved through some SDLC process model if possible. But, unfortunately not all the available process model does address these issues efficiently.

5.4 Desired Characteristics of DLC Models

In order to meet the project objectives and goal, SDLC have to satisfy many specific requirements i.e. being able to support different types of projects and systems of varying scopes, supporting both the technical and management activities, being highly usable, and providing guidance on how to execute and install it for solving real life problems and many more. In the following section we are going to identify, enlist and discuss briefly some primary characteristics that are expected from any SDLC process model. As the degree of importance of these characteristics does vary from project to project, here we just enlist these characteristics alphabetically.

- **Change Management:**
  Requirement changes are often necessary, frequent and inevitable. The drivers of requirement changes may be customer demand, technical demand, competitive demand or even governmental or business policy demand. While occasional changes are essential, historical evidence demonstrates that the vast bulk of changes can be deferred and phased in at a subsequent point. To develop quality software on a predictable schedule, the requirements must be established and maintained with reasonable stability throughout the development cycle. Changes will have to be made, but they must be managed and introduced in an orderly way. Hence, change management is a critical part of any SDLC model. As requirements are

changed frequently, there is a need of streamlined flexible approach to manage these requirement changes within the SDLC model. Although requirement change management and SLDC are not mutually exclusive but the change management activities occurs throughout the development process. Further, cost of adopting changes is higher after the completion of the development. Hence, the objective should be to limit the change management activities within the initial development period as much as possible. If change is not controlled, orderly testing is impossible and no quality plan can be effective.

- **Concurrent and Parallel Development:**
  If high cohesion and low coupling modular system design is possible, then concurrent, distributed and parallel system development activities can be employed. This can improve productivity, timely system delivery while reducing total development cost and optimal usage of available resources.

- **Coordination among project stake holders:**
  A software project is not an individuals' job, but a collective effort towards the common goal. The success of software development projects depends on carefully coordinating the effort of many individuals across the multiple stages of the development process. Coordination —- long recognized as one of the fundamental problems of software engineering has become ever more challenging. This has led to a growing body of work on coordination in software development (108). With the rapid advancement of Information and Communication Technology (ICT), the location or center specific project development barrier are being diminishing. For optimal resource utilization, often project development activities are distributed over different development centers. Further, more the people are involved in one job, more the chances of misunderstanding and communication gap. Hence, if large numbers of people are involved and scattered over different development centers, the process model must provide mechanism for better coordination among the project stakeholders.

- **Cost of Life Cycle Implementation:**
  Additional costs and overheads are the primary barrier in process model implementation. For this reasons, many organizations do not implement or follow any process model. An ideal process model implementation should be economic, easy and justifiable. It should not require additional, special application or software purchase to effectively perform the process implementation or ongoing process management. In

addition, it should be easily automated utilizing any internal process management software tools currently being used within an organization.

- **Customer Involvement and Interaction:**
  In most of the common process model, there is no direct communication among customer, development team or project management team throughout the development process. In traditional models, management plays the vital role of bipartite body who works as the communication channel and messenger in the communication between customer and development team. As a result always there remains some communication gap and some missing or hidden information yet to convey to the development team but with the management. As a result, often proper requirements remain unspoken or hidden to the development team. Even conveyance of information might cause a loss of knowledge, as great amount of data remains with its carrier and never get handed off to others (109). Some interviewees suggested that the lack of direct contact between the development team and the customers could encumber the process of specifying requirements for the future. In turn handoffs among functions can cause delays and increasing risks of information being misunderstood (110). According to interviewees, level of details is varying depending on representatives between customer and developer. As a result, the developed system is frequently not satisfactory or even lead to project failure. Allowing direct communication of development team with the customer during the entire development process could eliminate project completion time and recourses consuming non value-added-action in form of handoffs, therefore waste (111). Hence, user or customer involvement during all phases of the project development is very important for project success and must be supported by any process model.

- **Proper and Sufficient Documentation:**
  Documentation has two ways of influencing the development process. Firstly, it makes the development process easier to understand. Secondly, documentation enables easy system maintenance, which can be linked to one of the principles in lean philosophy – knowledge sharing. But, unnecessary documentation can be addressed as waste. Any process model must enforce to develop the necessary document concurrently with the development process, while must avoid producing unnecessary documents to prevent miss utilization or waste of resources as in the case of agile development philosophy.

- **Early Defect Removal:**
  In case of any error or defect, if possible, it is always better to remove or rectify them in the earliest phases of the SDLC process model. Hence, the process model must focus on identifying the errors in the same or closest phase of the SDLC process to avoid or reduce the redo-work and cost. The best way to identify errors is to perform a close and effective verification after each and every phase and to set specific predefined phase entry and phase exit criteria effectively.

- **Easy to Execute:**
  Not all process models are easy to execute. Some process execution may require additional focus than the other. But often degree of easiness in execution may affect the other evaluation criteria of the process model. Always neither all easy process models are bad nor are all complex process models good. Hence, the tradeoff must be resolved depending on the suitability project demand.

- **Effective Management and Control:**
  Most of the existing traditional SDLC process models don't involves management team directly with the development team. Hence, the project management team does not have direct communication with the development and associated members. The management just remains as a silent intermediate communication body. Thus, proper management observation and control is hidden in the development process. As a result, the development process lacks proper management supervision and controls. In addition, the project has to suffer from resource shortage, risk handling, coordination and many other conflicts and problems. To overcome these problems, a direct involvement of project management team with the development team is necessary and important. The software development process must be under statistical control of the project management team to produce consistent and better result through process improvement.

- **Focus Towards Goal:**
  The project objectives and goal must be well defined and specific. The process model should view software development within the context of the larger system level definition, design, and development. Further, it should recognize both the potential value of opportunity and the potential impact of adverse effects, such as cost overrun, time delay or project risks according to the system and project specifications. Hence, any process model must reflect the project scope, objectives and goal consistently during the development process.

- **Incremental:**
  Software development project may be divided into distinguishable cascading phases. Before starting a phase, it may require a defined set of inputs from its immediate previous phase. The incremental methodology maintains a series of such phases. However, in the design phase development is broken into a series of increments that can be implemented sequentially or in parallel. The subsequent phases do not change the requirements rather build upon them towards the project completion. The methodology continues focusing only on achieving the subset of requirements for that development increment and continues all the way through implementation. Increments can be discrete components, functionalities, or even integration activities. Hence, through the incremental methodology, quantifiable partial solutions may be given to the customer without waiting for the entire project to be completed. The subsequent partial system may be developed in parallel to the already developed and operational partial system that may be further integrated when the second incremental development is available. The incremental process increases the degree of customer satisfaction and product quality as the defect of the delivered partial system may be identified while at operation and necessary changes may be incorporated immediately and deliver with the next increment.

- **Iterative:**
  In iterative process, the development begins by specifying and implementing the partial software requirements available at the moment without waiting for the complete or full software requirements specification. Further, these partial requirements can be reviewed in order to identify additional requirements and necessary modifications are made. This process is then repeated to implement the newly identified and specified requirements producing a new version of the software at each such iteration of the process model. This allows the project team to take advantage of what was being learned during the development of earlier, incremental, deliverable versions of the software in use. Iteration enhances the ability of the project management team to efficiently address the requirements of stakeholders and to complete, review, and revises phase activities until they produce satisfactory results. The product is defined as completed when it satisfies all of its requirements.

- **Distributed or Multi-Site Software Development:**
  Recent advances in information technology have made Internet based collaboration much easier. It is now possible for a software team to draw on talented developers

from around the world without the need to gather them together physically. To solve the problems like team relocation and project delay, developing software at multiple sites has been considered these days. Besides the obvious advantage of being able to tap into a much larger pool of human resources, experts working together from two or more locations can actually yield better outcomes (112). In such cases, software managers have to be able to manage these distributed teams. They need to define sharper processes, tracked, overseen and ensure that they are followed.

- **Quick Implementation:**
  A predefined solution that does not require organizations to start from the very initial level would allow organizations to exponentially cut down the time required to fully complete a process implementation effort. A process blueprint solution would drastically reduce the time and cost associated with traditional process improvement by laying an effective foundation for SDLC organizations to build upon.

- **Phase Length and Cycle Duration:**
  Phase length and Cycle duration should be optimal. It is well known that long cycle and phase duration is one of the primary reasons for project failure. Further, long phase and cycle duration makes the performance measurement task more difficult. If the cycle and phase duration is long, there may be problem with resource scheduling and may promote unoptimized utilization of resources which may affect the project cost, quality and schedule. Another source of risk resides in the relatively long stages, which makes it difficult to estimate time, cost and other required resources for project completion. Further, if the cycle duration is more, the delivery of incremental and partial system delivery will get delayed that may decrease customer satisfaction. Hence, the process model must be designed in such a way that the duration of each cycle and length of each phase must be small.

- **Predictability:**
  Any process model should be predictable. That is, cost estimates, schedule and quality commitments would be met with reasonable consistency, and the quality of the resulting products would generally meet the users' needs (109). As money, time, people and many other resources are involved, and at the same time quality and customer satisfaction are prime concern, before starting the project we need to predict the future outcome from it. If the outcome is not favorable, carrying out the project is nothing but waste of resources and gaining loss! In addition, the

process model should ensure that we can produce desired functions with higher quality using optimal resources in lesser time in a predictable manner. Thus, the process should have a predefined level of precision to facilitate a complete, correct and predictable solution.

- **Process Tailoring:**
  There may be different kind of projects and situations when no standard process is applicable. In such cases, the process model should provide the flexibility to adopt itself with the project and situational demand maintaining the integrity and consistency of the process by permitting tailoring of the standard process. Hence, tailoring is the process of adjusting the standard process to obtain a process that is suitable for the project need and situation (113). Thus, process tailoring facility makes the process model more flexible and adaptable.

- **Progress Measurement:**
  To evaluate the project performance and management control progress measurement of the project work is very important. For this purpose, milestones need to be set at regular intervals. Otherwise the project may suffer from 99% complete syndrome (19, 20). Effective and proper project measurement is only possible when the development process is under statistical control (114).

- **Prototyping:**
  Prototyping is necessary when it is very difficult to obtain exact requirements from the customer at the beginning of the project. Given the prototype of the system, user keeps giving feedbacks from time-to-time and according to the feedback necessary modifications are incorporated in the proposed or to be developed system. By doing these, the hidden, unidentified user requirements may be discovered during the initial phases of the system development process. By doing so, project failure risks may br reduces while improving the degree of user satisfaction and system quality.

- **Quality Control:**
  Lack of quality assurance during the different phases of the development process is often a potential source of risk. Validating the product is restricted to a single testing phase lately in the development process. Hence, the testing phase is the highest risky phase, since it is the last stage wherein the system is put as a subject for testing. Thus, all problems, bugs, and risks are discovered too late when the

recovering from these problems requires large rework which consumes time, cost, and effort. Milestones and deliverables can be setup or specified for each step of the project. Each module of the project is thoroughly tested before the beginning of another module. Project requirements are measured against the actual results.

- **Reliability:**
  Any process model must ensure development of quality reliable systems. A potential source of risk resides in the relatively long stages of any process model, which makes it difficult to estimate, time, cost, and other resources required to complete each stage successfully. In general, if incremental model is followed, the partial working systems are delivered periodically. It increases the reliability of the process model as reliability of the system does increase gradually during each partial product delivery to the customer.

- **Repeatable:**
  A SDLC process model should be repeatable i.e. the process should be repeat in case of projects which are similar in type or belongs to similar domain. A repeatable process reduces the cost of process model implementation as it is well known, learned and experienced to the development team. Hence, repeat process will reduce the project risks and cost. The quality of the system will be better as the outcomes of the phases are predictable.

- **Reuse:**
  The advantages of reuse in developments are now well established. Studies show that reuse had great impact on productivity, cost, quality, time-to-market and customer satisfaction (115). But very few process models like BDIDGE (62) are designed keeping the view of reusability in mind. Hence, as reuse has potential advantages, support of reusability is an important desired characteristic for process model in recent days.

- **Risk Management:**
  Risk is commonly defined as a measure of the probability and severity of adverse effects (116). Risks are an inherent part of any project which must be managed in advanced (if possible) or during the development process. As all projects inherit some risks, it is desired that the process model should provide adequate scope for risk management. Project risk may be related to project staffing, resources, schedules/budgets, technical, requirements changes etc. The SDLC model must

focus on the risk associated with the project continuously so that the management can take necessary control measure to prevent project failure. To manage risks, any process model must provide direct control over the project by project management. But, it has been seen that a very few model i.e. spiral and BRIDGE (62) provides such direct management support to the development process.

- **Scope:**
  Client demands never ends! The more you provide, the more they demand! Hence, if the scope of the project is undefined, satisfying customer is just a dream. The process should clearly mention what is desired to be produced and the developed product should be comparable to the defined requirements. Thus, the process model should have its specific scope as well as must limit the scope of the project. Scope of the process model bounds the suitability of the process model for different types of systems and projects.

- **Security Assurance:**
  Effective security is incorporated at the onset of a project. If it is included as a requirement early in the system development and/or acquisition process, it typically results in less expensive and more cost effective security. Waiting to integrate security until later in the process usually results in interoperability issues and increased cost. Integrating security into the SDLC begins with being able to articulate the security properties desired within the system. This process is typically cyclical in refinement beginning at the top level and drilling down into what will eventually be security specifications. There are many ways to express the high level security requirements i.e. ISO 15408 and others.

- **Separation of Concern in Different Phases:**
  As the development process is divided into different phases with distinct objectives, hence the concern of different phases should be clear cut and well separated. Otherwise, there may be chaos during progress measurement, quality and other management problems. Without separation of concerns in different phases, it will be tough to set milestones effectively, in turn that will make the progress measurement task difficult.

- **Simplicity and Flexibility:**
  Neither all simple things are better, nor are all complex things bad! The process model should be simple to understand, easy to follow and well manageable. More

the process model flexible, it is easier to manage and follow for execution. As change is inevitable due to specific nature of system projects, flexibility to accommodate changes is a basic need from a process model.

- **Software Process Improvement and Feedback:**
  Software Process Improvement (SPI) is an approach to designing and defining new and improved software process to achieve basic business goals and objectives i.e. increase revenue or profit, and to decrease operating costs by manipulating or changing the software process. The objectives of software process improvement (SPI) are to process produce products according to plan while simultaneously improving the organization's capability to produce better products (117). Perfection of any process is done via constant improvements. During the project execution or at the end, the team members give feedback in the form of reports suggesting different ways to improve the development process for the next iterations or future. One of the conditions required to improve the development process is to have the input from previous cycles so that the team's opinion and experience gathered during previous phase can be disseminated among other teams. This supports the process improvement throughout the whole organization, by adopting one of the core lean principles such as knowledge sharing, which in this case drives forward another lean principle – perfection of the process (118). The benefits of SPI include increased customer satisfaction, productivity, quality, cost saving and cycle time reductions.

- **Statistical control:**
  As Lord Kelvin said: "when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the stage of science" (109). Statistical control means that if the work is repeated in roughly the same way, it will produce approximately the same result. Measurement is the primary principle behind the statistical control. A SDLC model should be under statistical control of project management team. Such a process model will produce the desired results within the anticipated limits of cost, schedule and quality. If the process is not under statistical control, no progress is possible until it is (114).

- **Suitability to Projects:**
  Not all process models are suitable for every type of projects. Some process models are suitable for large projects, while some may be better for small projects. But, there are some flexible process models which are suitable for different types of projects. Any process model should not be suitable to only a specific type of project or project scope. The model should be designed in such a way that it should support different types of project with their varying scope through process tailoring. Hence, suitability to projects may be considered as an evaluation criteria for a process model.

- **Support to the Modern Tools and Technologies:**
  Over the time, significant developments are made in the field of new techniques and methodologies. Those are to be incorporated, accommodated and supported in the process models to make it a sustainable for modern software development. If a process model fails to accommodate these new technologies, they gradually become obsolete and useless. For example, during last few years there has been significant development in the field of CASE tools for project management, configuration management, software design, modeling and many more. It is a proven fact that usage of CASE tools(119) increases the product quality and reduces the total project development cost and time to market. Hence, the process model should be designed to support and utilize the CASE tools.

- **Usability:**
  The usability requirement addresses the various ways in which the SDLC will be used by the team members easily, efficiently while at use. Usability is a composite property of a process model. It is a composition of five attributes i.e. i) Learning ability, ii) Efficiency, iii) User retention over time, iv) Error rate, and v) Satisfaction (120). Any process model should incorporate these usability attributes. Many usability process has been proposed by several people i.e. ACUDUC (Approach Centered on Usability and Driven by use cases) by Seffah et al (121), Usability Engineering Process Model (UEPM) by Granollers (122) and Xavier Ferre (120) has proposed an integrated model of usability and different SDLC activities to integrate usability especially in different SDLC models. Adoption Centric Usability Engineering (ACUE) facilitates the adoption of usability engineering methods for software engineering practitioners and thereby improves their integration into existing software development methodologies and practices (123).

5.5 Conclusion

An important initial step in addressing software problems is to treat the entire development process as a performable, controllable, measurable and improved process as a sequence of tasks that will produce the desired result. Any fully effective software process must consider the interrelationships of all the required tasks, tools and methods, skills, training and motivation of the people involved. In general, any process model must bear the properties that are investigated, identified and specified in this chapter. At the same time, an ideal SDLC process implementation should be quicker, cost effective and easy to implement and follow. It should also be stakeholder and project team friendly. Finally, the ideal process model should address the top challenges experienced by project managers. Although the degree of importance of these individual characteristics may vary from project to project and depends on situational demand, but we recommend that all the above discussed characteristics should be beard by any process model to suit for the modern real projects.