International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163
Volume 1 Issue 4 (May 2014) http://ijirae.com

SOFTWARE DEVELOPMENT LIFE CYCLE MODEL INCORPORATED WITH CLEMENCY BRASS

J.Srinivasan [1], R.Agila [2]
Assistant Professor[1], Research scholar[2]
Department of Computer Science [1, 2]
Adhiparasakthi College of Arts and Science [1, 2]
Kalavai, Tamilnadu[1, 2]
jsjagan@yahoo.com[1], agi.msc@gmail.com[2]

ABSTRACT- The System Development Life Cycle framework provides a sequence of activities for system designers and developers to follow. The ideas about the software development life cycle (SDLC) have been around for a long time and many variations exist, such as the waterfall, spiral, prototype and rapid application development model (RAD). These variations have many versions varying from those which are just guiding principles, to rigid systems of development complete with processes, paperwork and people roles. Clemency brass plays an important role in every software project since it is concerned with the delivery of a high quality product to end-users. This paper explores release practices employed by volunteer free software projects and shows problems that occur. A challenge that has been identified is the difficulty of coordinating a distributed team of volunteers in order to align their work for a release.

KEYWORDS- Rapid application development model, Clemency Brass, Experiment.

INTRODUCTION

A Systems Development Life Cycle (SDLC) adheres to important phases that are essential for developers, such as planning, analysis, design and implementation. A number of system development life cycle (SDLC) models have been created: waterfall, spiral, prototype and rapid application development model (RAD).

Various software development life cycle models are suitable for specific project related conditions which include organization, requirements stability, risks, budget and duration of project. One life cycle model theoretical may suite particular conditions and at the same time other model may also looks fitting into the requirements but one should consider trade-off while deciding which model to choose.

A software life cycle 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. Descriptive models may be used as the basis for understanding and improving software development processes. 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.

***Rapid Application Development (RAD) Model*- RAD** is a linear sequential software development process model that emphasis an extremely short development cycle using a component-based construction approach. If the requirements are well understood and defines, and the project scope is constraint, the RAD process enables a development team to create a fully functional system with in very short time period. RAD is a process through which the development cycle of an application is expedited. Rapid Application Development thus enables quality products to be developed faster, saving valuable resources.

CLEMENCY BRASS-The new proposed model is developed by incorporating Clemency brass within the scope of the SDLC basic phases like analysis, design, coding, testing and maintenance. Clemency brass is the concept which is quite new in the field of Software Engineering. The concept of clemency brass derives itself from the core concept of project management employed in Software Engineering. Software how-so-ever efficient and effective cannot be

considered commercially successful until and unless the software remains in the market for sufficiently long duration, in order to recover the cost that incurred during development and deployment of the software.

The clemency brass process is a relatively new but rapidly growing discipline within software engineering of managing software releases. As software systems, software development processes, and resources become more distributed, they invariably become more specialized and complex.

Furthermore, software products are typically in an ongoing cycle of development, testing and release. To this an evolution and growing complexity of the platforms on which these systems run and it becomes clear that there are a lot of moving pieces that must fit together seamlessly to guarantee the success and long-term value of a product or project. The need therefore exists for dedicated resources to oversee the integration and flow of development, testing, deployment and support of these systems.

Experiments on Rapid application development Model with Clemency brass

The proposed simulation model is built using the Simphony.NET simulation tool and with the concept of clemency brass. In fact, Simphony.NET consists of a working environment and a foundation library that allow the development of new simulation scenarios in an easy and efficient manner. A project in Simphony.NET is made out of a collection of modeling elements linked to each other by logical relationships.

Essentially, the proposed model consists of a set of resource, queue, task, probability branch, capture, release policies and counter modeling elements. The resources are the basic employees and workers assigned to work on the phases of the Rapid application development model. They are respectively the business analyst, the designer, the programmer, the tester and the maintenance man and the clemency brass team.

On the other hand, the Rapid application development phases are modeled as a set of task modeling elements each with a capture and release elements. The capture element binds a particular resource to a particular task and the release element releases the resource from the task when it is completed.

Additionally, several probability branch elements exist between the different tasks of the model whose purpose is to simulate the error probability that a Rapid application development task might exhibit after completion. The probability element has two branches: Branch 1 with Probability =0.1 denotes that 10% of the small-scale projects are subject to errors; and branch 2 with Probability =0.9 denotes that 90% of the small-scale projects will not exhibit errors after the completion of every phase. These branches simulate the recursive property of the rapid application development model to loop over the preceding task if an error was found in the current task.

Moreover, another probability branch element exists at the beginning of every project development cycle whose purpose is to simulate the scale of projects under development. It actually has three branches: Branch 1 with Probability=0.7 denotes that 70% of the incoming projects are small-scale; branch 2 with Probability =0.25 denotes that 25% of the incoming projects are medium-scale; and branch 3 with Probability =0.05 denotes that 5% of the incoming projects are large-scale.

The model starts with a new entity element which sets the number of incoming projects and a counter that counts the number of projects being received, and ends with another counter that counts the number of projects being delivered. Fig A-1 shows the different modeling elements for simulating small-scale type projects.

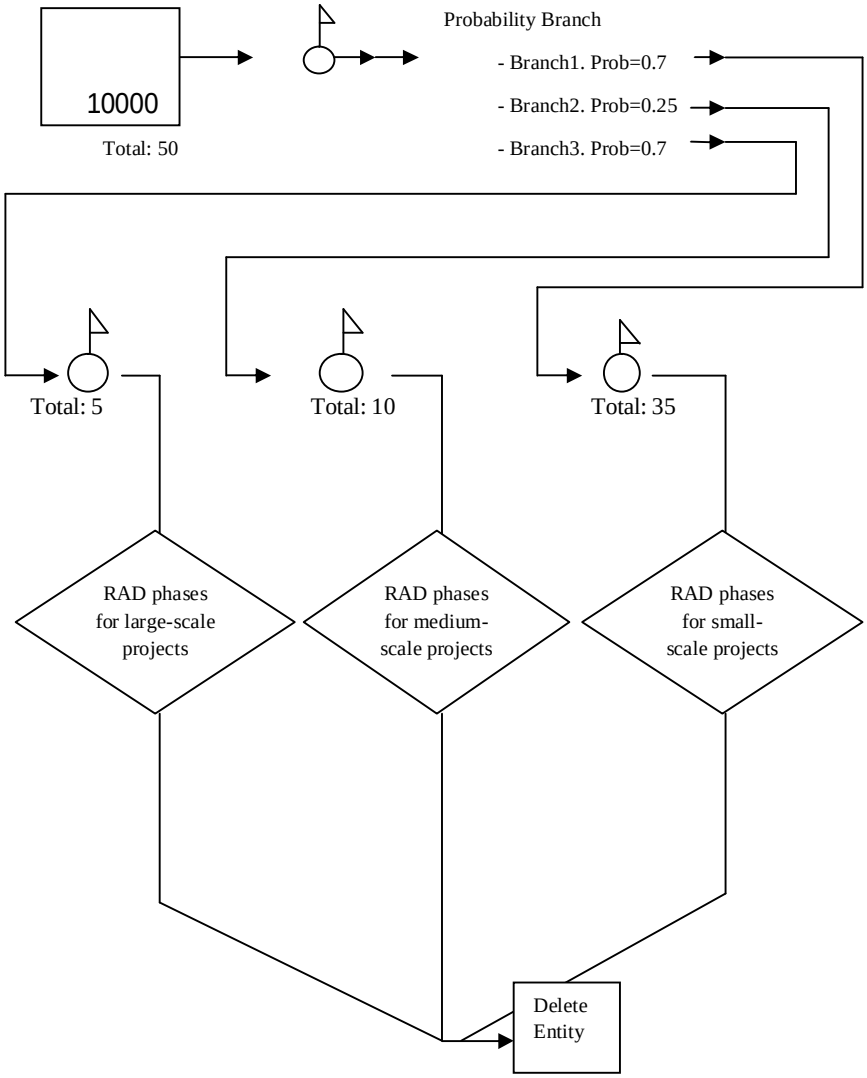

Fig. A-1: Projects with Different Scales for Rapid Application Development Model

CONCLUSION AND FUTURE ENHANCEMENT- Clemency brass is an entirely new concept that has come into scenario in recent days. Conceptually clemency brass itself has some major phases that ensure smooth and timely delivery of the software-system to the client. The proposed SDLC model merges the basic phases of software development and clemency brass in such a manner that a new SDLC model is evolved making the fullest use of clemency brass thus increasing the effectiveness and software life in the market.

The above model is practically successful as all the phases and the roles and responsibilities of each person involved are clearly established. Their interaction and inter-relation with each other is well defined and thus this model successfully addresses all development phases that are integrated with the concept of clemency brass.

Clemency brass can be incorporated with several other models like RAD, Prototype, Incremental, Spiral etc., in future to improve the effectiveness of performance and timely delivery.

REFERENCE:

1. Ian Sommerville, *Software Engineering*, Addison Wesley, 9th ed., 2010.
2. Richard H. Thayer, and Barry W. Boehm, "*software engineering project management*", Computer Society Press of the IEEE, pp.130, 1986.
3. Craig Larman and Victor Basili, "*Iterative and Incremental Development: A Brief History*", *IEEE Computer*, 2003.
4. N. Munassar and A. Govardhan, "*A Comparison Between Five Models Of Software Engineering*", IJCSI International Journal of Computer Science Issues, vol. 7, no. 5, 2010.
5. P. Humphreys, *"Extending Ourselves: Computational Science, Empiricism, and Scientific Method"*, Oxford University Press, 2004.

