The Structured Prototyping Life Cycle Model for Systems Development Management

E. K. Park and Kijoon Chae
Computer Science Department
United States Naval Academy
Annapolis, Maryland 21402

C. S. Kang
The American University
Washington, D.C. 20016

Abstract

There are many different methodologies and models for the concept of a project life cycle for software and knowledge engineering applications. These methodologies range from radical, all activities take place in parallel, to conservative, one activity is finished to completion before the next is started. By combining methodologies that come from both ends of this spectrum, a new methodology that contains the best aspects of each can be produced. The structured project life cycle is a methodology that is moderately conservative. The prototyping life cycle is a methodology that is radical. By combining these methodologies a new life cycle called the structured prototype life cycle is produced. This new methodology combines the rigor of the structured approach with the flexibility and quick production of results from prototyping. The new methodology can be considered a conservatively radical life cycle.

1 Introduction

Recently the approach taken to systems development has begun to change. More and more, large and small organizations are adopting a single, uniform project life cycle model. The documented project life cycle model provides a common way for everyone in the systems development organization to go about the business of developing a computer system. There are three primary objectives of having a project life cycle model[1,2]: (1) to define the activities to be carried out in a systems development project; (2) to introduce consistency among many systems development projects in the same organization; (3) to provide checkpoints for management control for go/no-go decisions. It seems apparent that the project life cycle model is desirable.

There are many different methodologies and tools for the concept of a project life cycle for software and knowledge engineering applications [1-4]. These methodologies range from radical, all activities take place in parallel, to conservative, one activity is finished to completion before the next is started. By combining methodologies that come from both ends of this spectrum, a new methodology that contains the best aspects of each can be produced.

Before introducing our structured prototype life cycle, it is important to examine some project life cycle models used in many systems development organizations today, primarily to identify their benefits, limitations and weaknesses. In particular we examine the structured project life cycle and the prototyping life cycle in section 2 and 3 respectively. By combining these two methodologies a new life cycle called the structured prototype life cycle is produced. Our new methodology combines the rigor of the structured approach with the flexibility and quick production of results from prototyping. The new methodology can be considered a conservatively radical, or middle of the road, project life cycle model. Section 4 describes our new methodology in detail. The discussion of observation and performance related issues is given in section 5. Finally, in section 6, conclusions are given.

2 The Structured Project Life Cycle

The structured project life cycle is a moderately conservative approach to the project life cycle. Unlike strictly conservative approaches, a moderate conservative approach means that many activities can be carried out in parallel [1]. Figure 1 shows the different activities that take place during the structured project life cycle. The arrows in the diagram show data flow only, the implication here is that many activities can be performed in parallel as long as the input data necessary for that activity is available. The fact that many activities can be performed in parallel means that more personnel on the project team can be
involved therefore reducing the overall cost of development. This is one of the benefits of the structured project life cycle.

An aspect of the conservative life cycle in general that is also beneficial is the development of a rigorous paper model of the system. This paper model is developed during the analysis activity of the life cycle. The reason that this is beneficial is that the underlying business policies that were in use during development should be documented in these paper models. This is important later on down the road when the system is being maintained or an enhancement is being added.

The major weakness of the structured project life cycle is the apparent lack of useful progress. What this means is that during the process of producing the paper models the users become discouraged because they see no tangible results being produced. Users generally do not understand the usefulness of these paper models and therefore think the analysts are wasting time.

3 The Prototyping Life Cycle

The prototyping project life cycle is a radical approach to the project life cycle. This approach is unlike most other methodologies for the project life cycle. Prototyping, as its name indicates, implies that the system model that is developed is an actual model of the system, not the paper model that is developed during other project life cycles\(^5-8\). Figure 2 shows the different activities that take place during the prototyping life cycle. Most of the time spent in the iterative refinement of user specifications. The process of refinement and implementation of the specifications is performed until the model is in a form that the user accepts as an accurate representation of the system. This process of iteration is performed with continual
involvement from the user. Because of user involvement, the user gets a feeling that progress in the development of the system is being made. The user involvement is the most beneficial feature of prototyping.

The most dangerous feature of prototyping is the fact that the prototype should be thrown away when the iterative refinement is completed. A complete production system should be built based on the characteristics of the prototype. Two problems arise out of this fact, firstly, the prototype is not thrown away and an attempt is made to make the prototype the production system. This leads to a system that can not handle the load put on it, the prototype was built for show not for efficiency. Secondly, the prototype is thrown away and all records of the users requirements and policy go with it. This often happens out of carelessness and renders the system impossible to maintain in future years.

4 The Structured Prototype Life Cycle

The structured prototype life cycle is a combination of the structured project life cycle and the prototype life cycle. The basic theory is to use the infrastructure of the structured project life cycle but to replace the analysis activity with prototyping as shown in figure 3 (compare with figure 1 and 2). The prototyping activity is basically the same sequence of activities as shown in figure 2, but there are some modifications, as shown in figure 4, that need to be made so that it can fit neatly into the new methodology.

4.1 Replacing Analysis with Prototyping

The driving force for replacing analysis with prototyping is not because of what is produced by these techniques but rather how these techniques produce
their results. The analysis activity produces a structured specification of the system by developing paper models of the user's environment. Prototyping produces a working model of the system and supporting documentation by iteratively refining users requirements. The output of both techniques is conceptually the same, a specification of what the system should look like and how it should act.

How the specification is produced is what sets these techniques apart. The analysis activity is performed by the analysts taking the user's policy and project charter and rigorously going through these documents to produce the specification. This activity includes some interaction with the users, but this interaction is minimal. The prototyping activity is performed by taking the user's policy and project charter and the users themselves and iteratively refining requirements to produce a working model. The key here is the amount of interaction by the user. The more interaction by the user the better.

4.2 Changes to the Prototyping Life Cycle

The main structure of the prototyping life cycle stays intact. The changes occur where the prototyping activity interacts with the rest of the methodology. The feasibility study (in figure 2 activity 1) and the question of if this project is a good candidate for prototyping are removed and included as part of the survey (in figure 3 activity 1). The remaining steps (in figure 2 activities 2 through 7) remain and are executed in the prescribed order. The last activity, clean up prototype and document (in figure 2 activity 7), is changed to accommodate the extra interfaces to the new methodology (in figure 4 activity 2.6). As before, the output of clean up prototype and document (in figure 4 activity 2.6) is sent to the design activity (in
figure 2 activity 8, in figure 3 activity 3). Two new interfaces must be added to the prototype activity, they are acceptance test generation (in figure 3 activity 5) and procedure description (in figure 3 activity 7).

The prototyping activity of figure 3 is executed the same way as it would be if it was being used as the whole project life cycle. What has been done to change this activity is now the interfaces have been modified to make prototyping part of a more rigorous methodology.

4.3 Why the Structured Prototype Life Cycle

The structured prototype life cycle is an attempt to create a life cycle methodology that is both structured but flexible enough to adapt to a wide range of development problems. The structured project life cycle is an excellent method of rigorous system development. The problem with it is that it is not enough user interaction. The prototyping life cycle is also a very good tool for system development. The problem with it is it is not as rigorous as an approach as the structured project life cycle. By combining the two we get the best of both worlds, a rigorous approach to systems development with a lot of user interaction.

4.4 Other Benefits

The structured prototype life cycle is basically some philosophical change to the structured project life cycle. The benefit of this is when we come across a project that does not lend itself to prototyping, such as a batch oriented system, all it takes is an easy switch
back to the structured project life cycle to be able to handle the task. It is as simple as switching from prototyping back to analysis in activity 2 of the structured project life cycle. All other activities are the same so it is a simple adjustment and flexible in some cases.

5 Observations

One question that might rise out of the discussions of using prototyping in place of analysis is that doesn't the overall development time increase by using prototyping method. The answer is no. The time it takes for the prototyping activity itself may be slightly longer but because of the nature of prototyping the follow-on activities, design and procedure description, should consume less time.

6 Conclusions

It is important to note that the project life cycle model definitely is not in charge of the project. It will not relieve the project manager of the difficult responsibility of making decisions, weighing alternatives, negotiating with recalcitrant users, boosting the morale of dejected programmers or any of the other project-related trials and tribulations. The only help that project life cycle model can provide is that it can organize the manager's activities, making it more likely that the right problems will be addressed at the right time.

Prototyping is rapidly becoming a widely accepted approach for systems development. While early predictions were that prototyping would help develop systems faster, there now is a strong trend toward the use of prototyping as a way to produce a more accurate requirements specifications, which lead to systems closer to user's expectations [9]. This is the reason why a way must be found to include prototyping in a rigorous project life cycle methodology. The structured prototype life cycle model we presented in this paper is a simple way of including new and coming methodology (prototyping) within the proven framework of an existing life cycle.

Acknowledgements

The authors would like to thank C. Shaw for his initial surveying effort on the topic.

References