Best Practices of RUP® in Software Product Line Development

Faheem Ahmed¹ and Luiz Fernando Capretz²
¹ College of Information Technology
United Arab Emirates University
P.O. Box 17555, Al-Ain, Abu Dhabi, United Arab Emirates
f.ahmed@uaeu.ac.ae
² Department of Electrical and Computer Engineering
University of Western Ontario
London, Ontario, Canada, N6A 5B9
lcapretz@uwo.ca

Abstract

Software industry has always been tried to figure out the best practices of software development in order to increase the success rate of software projects. Rational Unified Process presents and incorporates the six best practices of software industry to achieve the goals of high quality, reliable, cost effective and within schedule software projects. We present the concept of managing RUP® best practices in the development of software product lines. This concept further strengthens the core activities present in the process of software product line development. We implemented this approach in developing an E-Commerce Software Product Line presented as case study in this paper.

I. INTRODUCTION

Best practice in a particular field is one that is commonly used by most of the people and is always recommended to adopt for successful results. Software best practices are those, which are commonly used by most of the software development teams or individuals and are highly recommended for successful software project in terms of cost, quality and schedule. Rational identified and incorporated six best software practices in its software process model called Rational Unified Process (RUP), they are:

- Develop Iteratively
- Component Based Architecture
- Verify Quality
- Control Changes
- Manage Requirements
- Visual Modeling

Iterative development is composed of several iterations within a project. This approach leads to the concept of design little, build little and test little methodology. The customer and developer can see and monitor the progress of the product development easily during the iterations. Component based architecture emphasis on modular design, separation of functionality and reusability. It divides the entire product into pieces of sub systems. Those sub systems independently define core functionality. Verify quality involves in verifying and validating the functional and non-functional requirements of the system under development. This activity is recommended in all iterations to find out inconsistencies in requirements, designs and implementations, which ultimately result in the identification of defects at an early stage.

A control change accommodates tracks and monitors changes in the system under development. Since iterative development refines the requirements after all iterations, therefore an effective change control will provide a basis for reallocation of resources, managing requirements, design alterations and planning. Manage requirements describes how to elicit, organize, and document required functionality and constraints; track and document tradeoffs and decisions; and easily capture and communicate business requirements [1]. Visual modeling provides facility to build a graphical visual model to illustrate the structure and behavior of the architecture and components of the software product [2]. This helps in elaboration of requirements, and provides a source of documentation about the functionality of the system.
A. Software Product Line

A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way [3]. The objectives of a software product line is to reduce the overall engineering effort required to produce a collection of similar systems by capitalizing on the commonality among the systems and by formally managing the variations among the systems [4]. The essential activities involved during a product line development are:

- Core Asset Development
- Product Development
- Management

Figure 1 illustrates the inter-relationship among these three essential activities. Each rotating circle represents one of the essential activities. All three are linked together and are essential activities of product line development. Core assets in software development are those viable entities that had already been used or intend to use in software development of different projects. They can be requirement models, domain models, test cases, plans, designs, code, reusable components, etc [5].

Core assets provide the initial repository for the product line. The goal of the core asset development activity is to establish a production capability for products [5]. The product development is an activity in which actual products are developed based on core asset repository, requirements, product line scope and production plan. The products are developed according to production plans, and requirements are implemented by using existing core asset’s component or new components. This activity gives feedback to update the core assets and make changes in the production plan according to circumstances. Management is a vital activity in the process of software product line development. Activities must be given resources, coordinated, and supervised [5].

The management involves technical management and organizational management. Technical management is concerned with ensuring that all the technical activities including core asset development, development of products are following certain procedure defined to carry out job. Organizational management is concerned with defining certain necessary resources like manpower, money and equipments for smooth production.

II. MANAGING BEST PRACTICES IN SOFTWARE PRODUCT LINE DEVELOPMENT

A. Develop Iteratively

The Figure 2 describes the essential activities performed during product line development. Each rotating circle represents one of the essential activities. All three are linked together and in perpetual motion, showing that all three are essential, are inextricably linked, can occur in any order, and are highly iterative [5]. Core assets are used to develop new products and there is a constant chance of adding up the piles of core assets either as an outcome of new product development or COTS. The management takes its inputs from core assets and development phase and continuously gives feedback to both. The whole process clearly indicates that inherently iterative
development approach is considerably adopted throughout the software product line development.

B. Component Based Architecture

Conceptually software product line emphasis on reusability of existing component. Component based architecture practice will enable to identify existing and new components to fit into development activity. The development activity structures the whole product into modular component architecture to utilize the core assets identified during core asset development.

C. Verify Quality

The verifying quality practice introduced in an early stage during product development will reduce the product defect considerably. The requirement is not only to verify the end product but also to verify and validate the whole process starting from the beginning to end for verifying quality. Therefore it is recommended to introduce verifying quality practice during all three-core activities of software product line development. This will enable us to verify the core assets, validate the management activities and perform a quality assessment of the development process and the product itself.

D. Control Changes

The arrows in the rotating circles of Figure 2 indicates that the essential activities of software product line development gives and receives feedback from next activity. For example in any iteration after identifying core assets, management phase describe how to make use of those core asset and then development phase implements the concept, any deficiency present in the development phase will reflect in introducing changes in the other activities. The new requirements can be accommodated in iteration after defining the core assets and updating the production plans. Control Changes practice would enable to introduce changes in the existing requirements of the product line development and can define the procedures to update the management activity in order to accommodate changes in the development phase.

E. Manage Requirements

The manage requirement practice if introduced during core asset development and management activity then it would elaborate and elicit requirements in an effective way to reduce inconsistencies in the product line development requirements. The effective requirement analysis will lead to low defect probability in the product. Properly managed requirements will enable management activity to organize technical and non-technical aspects of product line development.

F. Visual Modeling

The visual modeling practice in core asset development will elaborate requirements and gives us a visual model of the entire product under development. This will help us in understanding requirements, functionality of the product, identification of various stakeholders and their perception about product line. Software product line analysis, which is an integral part of core asset development activity, recommends use case model, feature diagram and the object model to represent the requirements.

III. E-COMMERCE PRODUCT LINE CASE STUDY

We used reactive approach of software product line development, and build an E-Commerce Application for the online pharmacy. During the development phase we strictly follow the best practices of Rational Unified Process. Our component based design approach yields various components of the application and we started building the Core Asset repository. Control changes approach made us possible to introduce changes in our application as well as keep our core assets updated.

We model visually the entire application by using UML, which helped us to clearly understand and manage requirements. We verified quality of each and every component. After completing the first application for Canada Medicose Ltd, we organize our core assets and produced two more products with the name Canada Family Pharmacy and Pharma Super Store based on the same architecture but variability in certain functions. For all successive products we used the existing core assets for development and keep on adding new entities in our core asset repository.

Figure 3 shows the development time for the three products. It describes how development time is reduced in successive product development.

IV. CONCLUSION

The best practices of Rational Unified Process can play a considerable role in further strengthening the process of software product line development. Iterative development practice is inherently present in the conceptual model of software product line development. Control changes practice can provide a
systematic way to accommodate the changes required in the product line.

Verifying quality at all iterations during various activities of software product line development can reduce defects in product line. Manage requirements can increase the productivity of cores asset development and management activities of software product line development. Visual modeling can elaborate requirements for product line development. Component based architecture can result in increase of core assets and further increase the chances of reusability. This work illustrates how best practices can be incorporated and managed at various activities of software product line development.

![Development Time of Ecommerce Products](image)

**Figure 3.** Development time of products in E-Commerce software product line

ACKNOWLEDGMENT

Dr. L. F. Capretz is currently spending his sabbatical leave with the Department of Computer Science at the University of Sharjah, in the United Arab Emirates.

REFERENCES


