EasyChair Preprint

№ 5941

A New Process Model of Incremental Asset Building for Software Project Management

Farzana Sadia, Mahady Hasan, Nuzhat Nahar and
M Rokonuzzaman

June 28, 2021

A New Process Model of Incremental Asset Building for Software Project Management

Farzana Sadia
Dept.of Software Engineering
Daffodil International University
Dhaka, Bangladesh
sadia.swe@diu.edu.com

Mahady Hasan
Dept.of Computer Science & Engineering
Independent University, Bangladesh (IUB)
Dhaka, Bangladesh
mahady@iub.edu.bd

Nuzhat Nahar
Dept.of Computer Science & Engineering
Independent University, Bangladesh (IUB)
Dhaka, Bangladesh
nuzhat_n@hotmail.com

M Rokonuzzaman
Dept. of Electrical and Computer Engineering
North South University (NSU)
Dhaka, Bangladesh
zaman.rokon.bd@gmail.com

***Abstract*— Although global software industry is growing up rapidly, however surviving in software industry has not become remarkable as earning revenue and profit is below satisfactory in most of cases. Software companies are getting failed to make business by implementing typical software project management. The reasons of failing to earn profit is developing customized software of each client. In this paper, we have proposed a process model which will be followed additionally with typical software project management. By adopting this proposed process model will help the software companies to make profit of software business. This process model will build core and reusable software components based on targeted market segment and will integrate those components while developing software. Thus, software can be built with minimal customization and software companies can save their resources.**

Keywords*— *Software Industry, Software Project Management, Process Model, Reusability, Incremental Asset Development.

I. INTRODUCTION

Now a day's software plays an important key role in easing peoples' daily activities. Therefore, globally software business industry expansion is growing rapidly. Many people preferred software business as the demands of software application is getting higher every day. Although, software business is emerging, but success rate of software business is very low. People cannot make profit as their expenditure of software business is high. Therefore, surviving in software industry got very difficult for them. The reason behind their failure in doing software business is following traditional software project management where customized software is developed. Customized software development for each client costs unnecessary additional time and resource. Therefore, building and using reusable software components with less customization will help to build software with less time and resource and will help to get their business growth. [1-4]

In the early stage of software development, software was built for in-house projects and later this software industry expanded itself by developing software as a contract with client. Manufacturing contract based customized software for a particular client is still a continuous trend in the industry where software requirements are collected from clients, the budget is set, time and resources are allocated and so to develop such contract-based software application of the body of knowledge of project management is supposed to be successful. [5-10]

However, billing the total expenditure of customized software to a single client is implausible and also building similar projects from beginning for every client is expensive and wastage of time and resources that leads to project failure and loss of industrial sustainability due to the shortcomings of project management, although the body of knowledge of project management is being followed. [10-14]

Moreover, the current trend of developing customized software is changed to innovation where specific clients do not ask for neither software requirements nor cost, rather general software is built for many customers to meet common needs and the total cost is charged in small portion for the large number of clients. Therefore, when the companies grow in the industry with the help of innovation the net profit also becomes high. [15-20]

The objective of this paper is to recommend a process model concentrating on incremental asset building. The proposed model will help to reduce the average cost of products by increasing the scale benefits from practicing the product line engineering, where the total cost would not be charged to a single client rather it should be distributed among numerous clients with similar products.

AI. LITERATURE REVIEW

Any organization can develop core components or assets and later on they can be benefited by reusing those assets. For decreasing the cost of software and increasing value for the customer, organizing products into product families in needed [26]. Organization can analyze market, insight the market climate and determine the scope of the project [21-22]. Each organization should analyze its need and reuse measurements should be implemented [23]. Organization can gain benefit through practicing product line engineering [29]. Building a successful project through creating core reusable assets requires different management approaches. Managerial capability of project manager would assist to build software with optimum financing and boost the probability for earning revenue through which the firms can grow and sustain in the industry [24, 25]. To develop core asset, it is necessary to ready product line scope, core asset base and production plan [27]. Pierre America, Henk Obbink, Rob van Ommering, Frank van der Linden came up with a family of methods that guides to develop product line architecture which initiates significant interaction in development of many product families [30]. Paul Clements, Linda Northrop proposed an insight of wealth on running software development firms for managers [31]. Simon Gerlach proposed an approach to minimize the time of build series of products [32]. Takebe and Yasuaki describes method for software product line engineering for organization based of product oriented development [33] Kang, Kyo C., Jaejoon Lee, and Patrick Donohoe demonstrated the benefit of using Feature-Oriented Reuse Method which leads to the efficiency of software product line engineering with example [34]. Deelstra, Sybren, Marco Sinnema, and Jan Bosch showed a framework of terminology and concepts regarding product line engineering

[35]. Again, another method is proposed and demonstrated for feature modeling, architecture design and component design by using core assets [36, 37]. Kang, Kyo C., et al. exhibits design problems from the marketing perspective and proposed design driver which will help to build proper marketing strategy for building projects using product line architecture [38]. Lee, Kwanwoo, Kyo C. Kang, and Jaejoon Lee provided practicing guideline for developing project through software product line architecture [39]. Borba, Paulo, Leopoldo Teixeira, and Rohit Gheyi presented and established a language independent theory of product line refinement [44] Reinhartz-Berger, Iris, and OraWulf-Hadash came up with an idea of improving the management of product line engineering through domain knowledge extraction and cross PL analysis [45]

BI. Researchable Issues

Project failure which leads to zero or low revenue is the main concern of the unsustainability of software firms in this industry.

Bangladeshi software firm commonly delivers customized software projects, where the total cost of development is charged on a single client, which is very high and not affordable. Project managers assigned in multiple projects are diverse in nature and have less opportunity for reusability. Thus, most of the companies are experiencing very low or zero profit. This conventional way of software business is disrupting the growth limit of software firms.

To overcome such issue, bringing innovation to the delivery and optimizing the total cost of the project is very important. Rather targeting single customer for a single product, firms should sell single product to multiple clients. This can only be possible if the firms continuously enhance their products with incessant innovation.

Our proposed model will help to build incremental and reusable core software assets which can be used in consequent projects to deliver similar application in the same market segment. Building incrementally and reusing software components will decline the cost of project for every client and also the total cost of project development is spread out on multiple clients instead of one.

IV. Proposed Suggession

Software companies cannot grow and sustain in the industry as net profit is low and this occurs because of the high cost of the software is charged to a single client. The proposed "project incremental asset building" will assist to overcome this issue so that the software companies grow and sustain in the market and also clients get their best customer experience in terms of quality, cost and time. The proposed model is described below:

A. Incremental Asset Building Management

Project incremental asset building process model includes the activities to identify, define, combine and coordinate different processes and activities within project management process groups. This includes detecting market segment by analyzing market and gradually build asset for company.

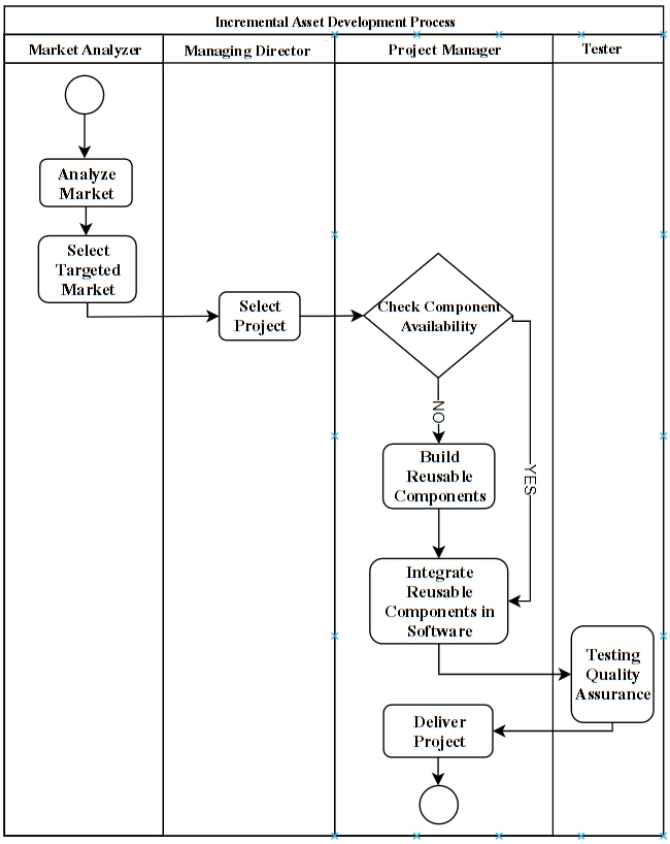

Fig-1: Incremental Asset Development Process

Fig-1 demonstrates the process of incremental asset building management. Firstly, market analyzer will analyze market and select targeted market, then managing director will select project. Based on the selected project, the core components to build the project will be checked. If components are available then the components will be integrated in the project, but if the components are not available then new reusable components will be developed then will be integrated in the project. After developing the project, the project will be tested to assure quality and then the project will be delivered to client or will be release for usage. The overview of project incremental asset building management is given below:

1. *Market analysis*: The process of determining the characteristics of particular market and analyzing the information, defining the scope of the project and building the reusable components.
2. *Select project:* The process of selecting a particular domain-based project based on the targeted market segment.
3. *Build reusable components:* The process of developing reusable components and incrementally adding more components for enhancing the product also for using in subsequent and future projects.

B. 10 Elements for Incremental Asset Building Process: 10 elements of a process describe about the purpose, performance parameters in producing output, policies, procedures, policies, standards, knowledge, skills, human resources, environment, tools, measurements, controls, improvements of the specific process to be executed. The 10 elements of the incremental asset building process in given below:

1. *Purpose:* Reduction the cost of project and sell the product to multiple clients

2. *Performance parameters in producing output:* Time and cost will be reduced and quality will be increased.

3. *Policies:* Build reusable components and integrate those components to build a project.

4. *Procedures:*

    a) *Analyze market and select a targeted market*: A quantitative and qualitative assessment of market helps to identify a targeted market. Table-1 demonstrates market information and market information gathering techniques is required for market analysis. Defining market segmentation is first task for analyzing market. Defining the elements and selecting suitable market segments also required for market analysis.

Table -1: Input, activities, output and artifact of Market Analysis

Input	Activities	Output	Artifacts
1.Market Information 2.Information gathering techniques	1. Define Market Segmentation 2. Analyses the market 3. Define the element of market analysis 4. Select the suitable market segment	Target Market segmentation	1.Market Research /Analysis report 2.Overview of the selected target market

    b) *Select projects on targeted market:* Projects selection on targeted market helps to focus on core assets. Table-2 represents that market research report and selected targeted market is required to select projects. Then finding and matching core assets helps to reuse components.

Table -2: Input, activities, output and artifact of Project Selection

Input	Activities	Output	Artifacts
1.Market Research /Analysis report 2.Overview of the selected target market	1.Identify project for the market segment 2.Check availability in the core asses 2.1 If yes then move to Phase 1 2.2 If no then select project	1.Selected project	1.Decision making evidences 2.Project selection communication 3.Overview of the Project

    c) *Build new reusable components:* Using core reusable components and building new reusable components are the key to build projects. Table – 3 shows that based on the market analysis report, decision evidences, project selection communication and overview of the projects, reusable components are developed. If no core components are found, then new components are built.

Table -3: Input, activities, output and artifact of Reusable Components Building

Input	Activities	Output	Artifacts
1.Market Research/Analysis report 2.Overview of the selected target market 3.Decision making evidences 4.Project selection communication 5.Overview of the Project	1.Understand the project scope 2.Check availability of the project component in the core asses 3.1 If yes then move phase 1 3.2 If no then build/add/delete/modify new component 4.Integrate reusable component and develop the project 5.Test the new/updated project	1.New/updated core assets (reusable component) 2.New/Updated user manual	1.New/Updated source code 2.Unit test case 3.SIT test case 4.Defect log

    d) *Integrate new and previous reusable components and build projects:* Projects are built by combing new and existing reusable components. Table-4 demonstrates new or updated source code and implementation plan is required to develop a new release of the software.

Table -4: Input, activities, output and artifact of integration of reusable components

Input	Activities	Output	Artifacts
1.New/Updated source code 2.Implementation/ Delivery Plan	1. Release the new. Updated project/product 2.Receive release note	New release /New version release	Release Note

5. *Standards:* Following different software development standards assist to develop the project efficiently. These standards are the common protocols, formats and practices which enable interoperability among the developers. The standards which must be followed are given below:

    a) Following design patterns and best practices in code.

    b) Develop de-coupled and generic components.

6. *Knowledge, skill, environment and human resource:* The core development team of software must be expertise and experienced on having knowledge of software development and software architecture. Moreover, the team have to be skilled on software management and human resource management. Additionally, proper workspace and infrastructures should be provided to the human resources.

7. *Tools:* Different tools on software requirement analysis, software architecture design, software development, software testing must be provided to the software development team. The team must also be facilitated with hardware with proper

configurations, required third party software services etc.

8. *Measurements:* Identifying and monitoring the progress of project management, project development and testing must be done. Moreover, maintaining schedule and cost is mandatory.

9. *Control:* Different control measures will help to develop the project in better way. Following regular meetings and collect work updates from developers, take feedback from clients will help to develop the project efficiently.

10. *Improvement:* As reusable components will be used in developing the project, so these components will be updated based on the feedback from the clients. Additionally, based on the market segment new core reusable components will be added with the already existing components.

V. CONCLUSION

The evolution of software industry started with in-house development and taking software projects as contracts. Now is the era for innovation and a software company can sustain in the market if innovative software is developed that has reusable components so that with little customization the same software can serve multiple customers whereby development costs decreases and the price of software goes down, making it reasonable for the customer and at the same time a good revenue is earned by the software firm. In order to achieve this, market analysis is proposed to be the first activity in the project management so that the innovative software to be developed can be identified first and then other activities in the project management can follow.

In today's trend for innovative software development, there are six screening criteria for software requirements and one of the criteria is the ability to create network externality effect. It is the effect that user of a software creates on the perceived value as the number of users of the software increases. Keeping this in mind, innovative software is to be developed.

Software companies have been failing to reduce cost, improve quality of project and increase net profit as it has been found that companies' project management strategy is totally detached from taking advantage from economic scale and scope. Thus, companies continuously fail to deliver projects within the expected budget and time. Therefore, software firms should follow incremental process building model. Through this market analysis, project manager would get visions about the scope of building reusable assets and benefiting from them. Such insights would help to develop components with minimum cost and time which can be used in current project and as future reusable components. So, in absence of unification of market insights and product line engineering in conventional project management body of knowledge, software companies would not be able to succeed in developing profitable software business and sustain in software industry for long time. The limitation of this work is that the proposed knowledge area is not applied on companies and observed the outcome of practicing the proposed knowledge area.

Future works on other phases of software development is open for research on Bangladeshi perspective. Our work has covered software marketing, project management, software requirement engineering and software process. Future works can be done on software architecture, software testing and business process re-engineering based on Bangladeshi software firms with innovative software solutions.

REFERENCES

1. Z. Rokon, Software and Wealth Creation, Accessed on: 30March,2021[Online].Available: http://softwarewealth.blogspot.com/2016/09/software-project-management-knowledge.html
2. PasiTyrväinen, JoonaSelin: How to Sell SaaS: A Model for Main Factors of Marketing and Selling Software-as-a-Service. ICSOB 2011: 2-16
3. Linda Hill, Kay Fielden: Marketing Innovative Software: A New Zealand Case Study. PACIS 2007: 17
4. Hershey H. Friedman, Linda Weiser Friedman: Marketing methods for software. Journal of Systems and Software 7(3): 207-212 (1987)
5. KimmoAlajoutsijärvi, Kari Mannermaa, HenrikkiTikkanen: Customer relationships and the small software firm: A framework for understanding challenges faced in marketing. Information & Management 37(3): 153-159 (2000)
6. Nishino N., Takenaka T., Ueda K. (2008) Product Diffusion in a Market with Network Externalities: An Approach from the Viewpoint of Value-creation among Consumers. In: Mitsuishi M., Ueda K., Kimura F. (eds) Manufacturing Systems and Technologies for the New Frontier. Springer, London
7. Blankart C.B., Knieps G. (1993) Network Evolution. In: Wagener HJ. (eds) On the Theory and Policy of Systemic Change. Studies in Contemporary Economics. Physica-Verlag HD
8. Z. Rokon, Software and Wealth Creation, Accessed on: 30March,2021[Online].Available: http://techpolicyviews.blogspot.com/2016/12/innovation-journey-from-material-to.html
9. Z. Rokon, Software and Wealth Creation, Accessed on: 30March,2021[Online].Available: xhttp://techpolicyviews.blogspot.com/2017/02/evolving-roles-of-software-requirements.html
10. Z. Rokon, Software and Wealth Creation, Accessed on: 30March,2021[Online].Available: http://techpolicyviews.blogspot.com/2016/11/innovation-journey-from-material-to.html
11. Z. Rokon, Software and Wealth Creation, Accessed on: 30March,2021[Online].Available: http://techpolicyviews.blogspot.com/2016/08/natural-tendency-of-monopoly-why-does.html
12. Yuan-yuan Lu, The Role of Advertising and Promotion Strategy of Media Firms in Two-Sided Markets with Negative Network Externality, 2011 International Conference on Management and Service Science
13. Externalities and peer effects of collective adoption in networks, Arash Vesaghi; Mo Mansouri, 2016 11th System of Systems Engineering Conference (SoSE)
14. A simulation research of innovation diffusion based on [network externality, Xiaoqing Zhao; Siyuan Lou; Yaxiu Yang, 2011 International Conference on E-Business and E-Government (ICEE)
15. The Network Externality and Game-Playing Time Characteristics in Different Types of MMORPGs, Hui Peng; Yanli Sun, 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing
16. Should we wait? Network externalities and electronic billing adoption, Y. A. Au; R. J. Kauffman, Proceedings of the 34th Annual Hawaii International Conference on System Sciences
17. https://en.wikipedia.org/wiki/Network_effect#Software
18. 'Software Sector of Bangladesh', ACB Foundation, Dhaka, 2015.
19. Farzana. "Reusable components use by software companies of Bangladesh". Survery. 11 October, 2016
20. Farzana. "Innovation software development by software companies of Bangladesh".Survery. 11 October, 2016
21. Project Management Institute, A guide to the project management body of knowledge. 2012, p.91, 132, 169, 220, 256, 284, 314, 338, 382, 418.
22. Paul C. Clements, Lawrence G. Jones, and Linda M. Northrop, John D. McGregor, 'Project Management in a Software Project Line Organization', IEEE Computer Society, 2005.
23. 'Success factors of systematic reuse', IEEE Software(Volume: 11, Issue: 5, Sept. 1994)
24. [7] RokonZaman, 'Software Project Management knowledge: Why is it incomplete?', 2016.

25. Arthur B. Pyster, Richard H. Thayer, 'Software Engineering Project Management 20 Years Later',IEEE Computer Society, 2005.
26. Pierre America, Henk Obbink, Rob van Ommering, Frank van der Linden, 'CoPAM: A Component-Oriented Platform Architecting Method Family for Product Family Engineering', Springer, 2000.
27. P. Clements and L. Northrop, A Framework for Software Product Line Practice, version 4.2; Software Engineering Inst., Carnegie Mellon Univ., 2004; www.sei.cmu.edu/plp/framework.html.
28. P. Clements and L. Northrop, Software Product ['Lines:Practices and Patterns, Addison-Wesley, 2002.
29. P. Clements and L. Northrop, A Software Product Line Case Study (CMU/SEI-2002-TR-038, ADA412311), Software Engineering Inst., Carnegie Mellon Univ.,2002;Accessed on: 30 March, 2021. Available: www.sei.cmu.edu/publications/documents/02.reports/02tr038.html.
30. Pierre America, Henk Obbink, Rob van Ommering, Frank van der Linden, "CoPAM: A Component-Oriented Platform Architecting Method Family for Product Family Engineering", in Experience and Research Directions, 576, Session 3, Patrick Donohoe, Springer US, pp 167-180
31. Paul Clements, Linda Northrop. "Software product lines: practices and patterns [book review]." IEEE Software 19.4 (2002): 131-132.
32. Gerlach, Simon. "Improving efficiency when deriving numerous products from software product lines simultaneously. "Proceedings of the 15th International Software Product Line Conference, Volume 2.ACM, 2011.
33. Takebe, Yasuaki, et al. "Experiences with software product line engineering in product development-oriented organization." Proceedings of the 13th International Software Product Line Conference. Carnegie Mellon University, 2009.
34. Kang, Kyo C., Jaejoon Lee, and Patrick Donohoe. "Feature-oriented product line engineering." IEEE software 19.4 (2002): 58-65.
35. Deelstra, Sybren, Marco Sinnema, and Jan Bosch. "Product derivation in software product families: a case study." Journal of Systems and Software 74.2 (2005): 173-194.
36. Kang, Kyo C., et al. "Feature-oriented product line software engineering: Principles and guidelines." Domain Oriented Systems Development: Perspectives and Practices, K. Itoh et al., eds (2003): 29-46.
37. Kang, Kyo C., et al. Feature-oriented domain analysis (FODA) feasibility study. No. CMU/SEI-90-TR-21. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 1990.
38. Kang, Kyo C., et al. "Using a marketing and product plan as a key driver for product line asset development." International Conference on Software Product Lines. Springer Berlin Heidelberg, 2002.
39. Lee, Kwanwoo, Kyo C. Kang, and Jaejoon Lee. "Concepts and guidelines of feature modeling for product line software engineering." International Conference on Software Reuse. Springer Berlin Heidelberg, 2002.
40. Bass, Len, Paul Clements, and Rick Kazman. "Software Architecture in Practice, Addisson." (1998).
41. Gamma, Erich. Design patterns: elements of reusable object-oriented software. Pearson Education India, 1995.
42. Prieto-Diaz, Ruben. "Implementing faceted classification for software reuse. "Communications of the ACM 34.5 (1991): 88-97.
43. Weiss, David M. "Software product-line engineering: a family-based software development process." (1999).
44. Borba, Paulo, Leopoldo Teixeira, and RohitGheyi. "A theory of software product line refinement. "International Colloquium on Theoretical Aspects of Computing. Springer Berlin Heidelberg, 2010.
45. Reinhartz-Berger, Iris, and OraWulf-Hadash. "Improving the management of product lines by performing domain knowledge extraction and cross product line analysis. "Information and Software Technology 59 (2015): 191-204.
46. Northrop, Linda, and P. Clements. "Software product lines."URL http://www.sei. cmu. edu/library/assets/Philips 4.05 (2001).
47. Heie, Anders. "Global software product lines and infinite diversity." keynote address, Second Software Product Line Conference. 2002.
48. Toft, Peter, Derek Coleman, and Joni Ohta. "A cooperative model for cross-divisional product development for a software product line. "Software Product Lines. Springer US, 2000. 111-1