Exploring the Business Case for Transitioning from a Framework-based Approach to a Software Product Line Engineering Approach

Bedir Tekinerdogan[1], Eray Tüzün[2], Ediz Şaykol[2]

[1] Bilkent University, Department of Computer Engineering, 06800 Bilkent Ankara, Turkey
bedir@cs.bilkent.edu.tr

[2] Havelsan A. Ş., Peace Eagle Program, Research and Development Team,
ODTU Teknokent, 06531, Ankara, Turkey
{etuzun,esaykol}@Havelsan.com.tr

***Abstract*. In this paper we report on our experiences in transitioning from an application framework for Command and Control systems to a product line engineering approach. The application framework has been developed in the context of Havelsan, a large company in Turkish defense industry. Large investments have been undertaken in the framework development to support gross-level reuse of command control systems. We report on the obstacles of the application framework for supporting large scale reuse, explore the business case for transitioning the framework to a product line engineering approach, and provide the lessons learned in the overall process.**

Keywords: application frameworks, software product line engineering, business case, transition strategies

I. INTRODUCTION

The benefits for adopting a product line approach has been analyzed and discussed before by several authors [4][5][11]. The key motivation for adopting a product line engineering process is to develop products more efficiently, get them to the market faster to stay competitive and produce with higher quality. In alignment with these goals different software product line engineering processes have been proposed [5]. Unfortunately, a transition to software product line engineering is not easy. In general it requires large upfront investments and as such forms a serious risk if the desired return-on investment is not achieved. Furthermore, there is not a single fixed strategy for adopting a PLE. This is because each organization has its own specific context and usually decides to adopt a product line approach with specific business goals in mind [10][11].

In this paper we report on our experiences in exploring and defining the business case for transitioning a framework for Command and Control (C2) systems to a product line engineering approach. This work has been done in the context of Havelsan, a Turkish software and systems company having business presence in defense and IT sector. Havelsan is one of the subcontractors of the Peace Eagle Program of the Turkish Air Forces, the primary contractor of which is Boeing. In the C2 domain, Havelsan has several projects for Turkish Armed Forces, such as the Meltem project, where Thales is the primary contractor. Earlier the company has started to focus on software reuse and this resulted in a definition of application framework for C2 systems. The software framework is intended to be developed to satisfy the needs of Turkish Air Forces, Turkish Naval Forces, and Turkish Ground Forces, and to be used in both upcoming projects and the upgrade programs of the ongoing projects.

To provide a broader support for reuse, transitioning to a product line engineering approach is seriously considered. However, due to the large investments that have been made for developing the framework, the key constraint for this PLE approach is to preserve the framework as much as possible. Before deciding on the transition to a product line a clear business case was required. We report on the obstacles of a framework-based approach and the exploration process for the business case for a transitioning to a product line engineering approach.

The remainder of the paper is organized as follows. In Section 2 we discuss the development of the Command and Control Software Framework (CCSF), at Havelsan. In Section 3, we discuss the possible alternative management decisions for either continuing with the framework-based approach or transitioning to a product line engineering approach. Section 4 describes the cost model for the framework-based approach and the cost model for transitioning to a product line engineering approach. Finally we conclude the paper in Section 5.

II. Application Framework for Command and Control Systems

During the last year, two C2 systems were developed utilizing the CCSF. We have had several years of experience now with the framework-based approach and can derive important lessons for the next generation future projects and the adopted reuse approach. First of all, in our experience we could observe that the framework-based approach did indeed pay off and resulted in the expected return on investment compared to single system development. Besides of the benefits of frameworks various authors have also indicated the corresponding challenges [3][6] These include development effort, learning curve, integration with other frameworks, evolution of the framework and maintainability, validation and defect removal, efficiency, and lack of standards [3][6]. In alignment with the literature we have carried out a study in the company to assess the framework-based approach that was adopted. For this we have discussed and interviewed various stakeholders of the framework for several weeks.

The problems that were described were considered from three different perspectives: *business perspective*, *organizational perspective* and *technical perspective*. From the *business perspective* it was not easy to define the right scope that would pay off. The problems related to *organizational perspective* included among others, the lack of a proper organization structure for reuse and, lack of appropriate communication channels among the framework developers, framework users and the customers. Finally, problems related to technical perspective included the difficulty for handling variability, difficulty of defining domain scope, lack of reuse of assets other than code, and different methodologies applied by framework developers and application developers (framework users).

In the near future, in addition to the maintenance of the framework for the current applications, we are expecting more projects that would require using CCSF for rapid development of C2 applications inside the organization in the near future. When consulting the product line engineering literature most of the identified problems seem to be addressed and better solved by adopting a systematic product line engineering approach. In the light of these facts, at Havelsan it was decided to explore the business case for adopting a product line engineering approach.

III. Decision Alternatives

The obstacles of the framework-based approach were clear with the real practical experiences but a shift to a more mature reuse approach, the product line engineering approach, would require even more additional investments while it could lead to unforeseen risks and costs. As such, if the shift to a product line engineering approach would not be well managed, it could even decrease the benefits that were gained with the conventional framework-based approach. On the other hand, it was necessary to improve the reuse level of the company and achieve a more systematic reuse approach from the future perspective with the expected return on investments.

This whole reflection process from the upper management led to the decision to provide a feasibility study on the adoption of a product line engineering approach. For this a systematic plan had to be defined to reason about the possible risks and benefits of both the approaches. In fact, we had to decide among the following two alternatives:

(1) Continue with the current framework-based approach or

(2) Adopt a product line engineering approach.

We have illustrated the problem by drawing Fig 1 to compare the approaches for system development that we have categorized as single system development, framework-based development, and product line engineering approach.

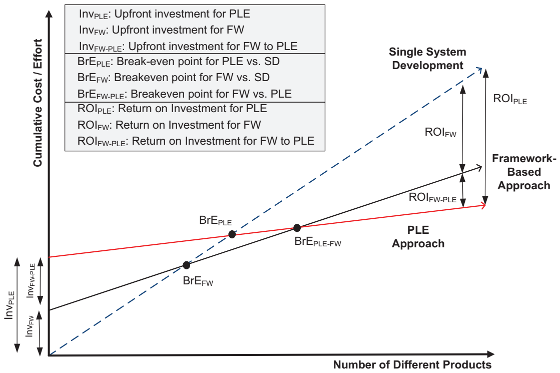

Fig 1. Comparison of PLE, framework-based approach and single system development

Here we can distinguish between three different types of upfront investment: upfront investment for moving from a single system development to a product line engineering approach (Inv_{PLE}), upfront investment for moving from a single system development to a framework-based approach (Inv_{FW}), and upfront investment for moving from a framework-based approach to a product line engineering approach ($Inv_{FW\text{-}PLE}$). In parallel we can distinguish between three different types of ROI: ROI for framework-based approach with respect to singles system development (ROI_{FW}), ROI for product line engineering approach with respect to single system development (Inv_{PLE}), and ROI for product line engineering approach with respect to framework-based approach ($ROI_{FW\text{-}PLE}$). Finally, we could distinguish between three types of breakeven points (see Fig 1). The first global observation from Fig 1 is that the framework-based approach falls between the single system development approach and product line engineering approach.

IV. Cost Model of Adoption Strategies

The cost model for the adoption strategy determines the general cost *C* for establishing a software product line of *n* products p_i according to the formula in Eq. 1 as follows[11]:

$$C = C_{Org} + C_{Cab} + \sum_{i=1}^{n}(C_{Unique}\,(p_i) + C_{Reuse}\,(p_i))$$
(Eq. 1)

As defined by Pohl et al. [11] these parameters represent the cost of adapting the software reuse approach for the organization (C_{Org}) , the cost to define the asset base (C_{Cab}), the cost for developing unique products (C_{Unique}) and the cost of reusing core assets (C_{Reuse}). Pohl. et al. [11] also provide different formulas for different strategies based on Eq.1. Additionally, since we had to decide whether to continue with the framework-based approach or to switch to a product line engineering approach, we have defined the cost models for the alternative decisions. These are explained in the following subsections.

A. Cost of Framework-Based Approach

We have defined the cost formula for developing *n* products with our framework-based approach, based on Eq.1, as follows:

$$C_F = C_{f-Org} + C_{frw} + \sum_{i=1}^{n}(C_{Unique}\,(p_i) + C_{Reuse}\,(p_i))$$
(Eq. 2)

The parameters of Eq. 2 can be explained as follows:

$C_{f\text{-}org}$: The cost for adapting the organization for framework-based approach. It is assumed that $C_{f\text{-}org}$ is significantly less than C_{Org} in Eq. 1, since there are fewer required roles in framework development process, and application engineering process is to be done by the framework users.

C_{frw} : The cost of developing a framework. This is similar to C_{cab} in Eq. 1; however, C_{frw} is assumed to be less than C_{cab} since the commonality/variability analysis are not as detailed as in product line engineering, such as code-reuse is of primary concern rather than documentation, test-cases reuses, etc.

C_{unique} : The cost to develop unique parts of the application that are not reused from framework. $C_{unique}(p_i)$ costs will be more or less the same, but due to the smaller reuse scope in framework-based approach, C_{unique} will be more than that of product line engineering.

C_{reuse} : The cost of using the framework to develop unique applications. Similarly, this cost is expected to be higher than that of product line engineering.

An interesting observation based on our experiences in Havelsan, C_F is strongly dependent on the decision that the framework is either to be used within the company itself or to be sold to other companies. If the main purpose of framework development is to obtain inside-company reuse, the cost model in Eq. 2 applies. In case of selling the framework to other companies, the costs C_{unique} and C_{reuse} reduce to 0, and $C_{f\text{-}org}$ becomes very low.

B. Cost of Transitioning to Product Line Engineering:

We have defined the cost formula for transitioning from framework-based approach to product line engineering as follows:

$$C = C_{org-transition} + C_{frw-to-cab} + \sum_{i=1}^{n}(C_{Unique}\,(p_i) + C_{Reuse}\,(p_i)) \quad \textbf{(Eq. 3)}$$

The parameters of Eq. 3 can be explained as follows:

$C_{org\text{-}transition}$: The cost of organizational transition to a product line organization from framework-based organization. $C_{org\text{-}transition}$ is assumed to be very close to C_{Org} in Eq. 1, since $C_{f\text{-}org}$ is significantly smaller than C_{Org}.

$C_{frwk\text{-}to\text{-}cab}$: The cost to an organization to convert the framework into a set of reusable assets as described in product line architecture. Since framework is similar to the reusable assets concept in product line engineering, $C_{frwk\text{-}to\text{-}cab}$ is assumed to be significantly less than C_{cab} in Eq. 1, which is the cost of developing reusable core assets from scratch.

C_{unique} and **C_{reuse}** are assumed to be the same as in Eq. 1, since product line engineering is achieved after the transition.

C. Evaluation of Strategies

Havelsan made investments on single-system development earlier, and then transitioned to framework development to obtain a desired level of reuse, as shown in Fig **2**. This was the first decision made (see d_1 in Fig 2), and require up-front investments for framework development. The ROI for this investment was gained after the use of the framework when compared to single system development. At the time being, Havelsan is at another decision point (see d_2 in Fig 2) regarding the transition to product line engineering. Greater level of systematic reuse in terms of design, documentation, and development process and more ROI are expected based on this decision of transitioning to product line engineering approach.

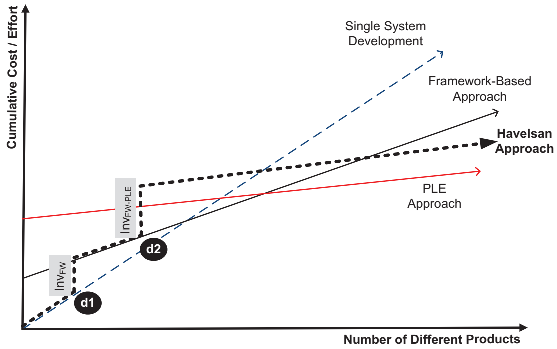

Fig 2. Evaluation of Havelsan's Strategies based on Cost Estimations

When we compare *Product Line Engineering Approach from Scratch* (PLE Approach in Fig 2) with *Framework-Based Product Line Engineering Approach* (Havelsan Approach in Fig 2), the main cost difference $(C_{cab} + C_{org})\text{-}(C_{f\text{-}org} + C_{frw})$ is based on Eqs. 1 and 2. This cost difference designates the up-front investment for PLE Approach is deferred to a later stage, which is a desired issue from the financial perspective. Moreover, by deferring the large up-front investment in PLE approaches, we had a chance to analyze our reuse strategy and also even get a small ROI from pilot applications (with shorter time-to-market compare to PLE) using our framework. With the framework-based reuse strategy Havelsan would be more inclined to pay the up-front investment difference for transitioning to PLE, which was avoided before.

V. Conclusions

In this paper we have presented our experiences in an industrial project for defining and exploring the business case for transitioning from a framework-based approach to a product line engineering approach. From the practical experiences we started to observe a return on investment with the framework-based approach with respect to single system development. Nevertheless, based on interviews and discussions with the framework stakeholders, it appears that the company had also to cope with several obstacles related to framework-based development. The problems relate to business perspective, organizational perspective and technical perspective. On the other hand the obstacles of the framework-based approach also triggered the exploration of a business case for transitioning to a product line engineering approach. Although the concept of product line engineering was known in the beginning the upper level management did not decide to directly make the transition due to the required high risks and investments. Moreover, due to the very high time-to-market constraints it was very difficult for the company to wait for the development of the product line engineering assets before the first product could be delivered. Currently the company is at the point of transitioning to a product line engineering approach. As stated before, the obstacles that are related to the framework-based approach form an important input to the business case for the transitioning to a product line engineering approach. Our future work will include the definition of the concrete steps for transitioning to a product line engineering approach.

References

[1] D. Batory, R. Cardone, and Y. Smaragdakis. *Object-oriented framework and product lines*. In *SPLC: Experience and Research Directions: Experience and Research Directions,* P. Donohoe, Ed. Kluwer Academic Publishers, Norwell, MA, 227-247, 2000.

[2] A. Birk, G. Heller, I. John, K. Schmid, T. von der Massen, K. Müller, *Product Line Engineering: The State of the Practice*. IEEE Software. Vol. 20, No. 6, pp. 52-60, November, 2003.

[3] J. Bosch, P. Molin, M. Mattson, P. Bengtsson. *Object oriented frameworks—problems and experiences. Building Application Frameworks*, Fayad ME, Schmidt DC, Johnson RE (eds.). Wiley & Sons, 1999.

[4] S. Buhne, G. Chastek, T. Kakola, P. Knauber, L. Northrop, S. Thiel. *Exploring the context of product line adoption*. in 5th International workshop on Product Family Engineering. F.V.d. Linden, Editor. Siena, Italy, LNCS, 2003.

[5] P. Clements, L. Northrop. *Software Product Lines: Practices and Patterns*. Boston, MA:Addison-Wesley, 2002.

[6] M.E. Fayad, D. Schmidt. *Object-Oriented Application Frameworks*, Vol. 40, No. 10, Communication of the ACM, October 1997.

[7] A. Helferich, G. Herzwurm, S. Jesse, and M. Mikusz. *Software product lines, service-oriented architecture and frameworks: Worlds apart or ideal partners?* In Trends in Enterprise Application Architecture, volume 4473/2007, pages 187–201. Lecture Notes in Computer Science, Springer, 2007.

[8] C. W. Krueger. *Easing the Transition to Software Mass Customization*, 282-293. Proceedings of the 4th International Workshop on Software Product Family Engineering. Bilbao, Spain, October 3-5, 2001. Berlin, Germany: Springer-Verlag, 2002.

[9] C.W. Krueger, *New Methods in Software Product Line Development*. BigLever Software, Austin, TX; in Proc. of 10th Software Product Line Conference, 2006

[10] J.D. McGregor, S. Jarrad, L.M. Northrop, and K. Pohl. *Initiating Software Product Lines*, IEEE Software, vol. 19, no. 4, July 2002, pp.24–27.

[11] K. Pohl, G. Böckle, F. van der Linden. *Software Product Line Engineering – Foundations, Principles, and Techniques*, Springer, 2005.