

# Analysis of a Small Company for Software Product Line Adoption — An Industrial Case Study

N. Nazar and T. M. J. Rakotomahefa

**Abstract**—While there are many success stories of adoption of software product line (SPL) practices in large companies, there is a need to investigate how well the product line strategy works for small organizations and what issues and challenges they could possibly face. This case study provides a comprehensive detailed analysis of a small company's tendency of adopting Software product line. It also reveals how new trends in software engineering, for example, agile methods interact with product line strategy. We used an ethnographic approach of collecting data by spending some time at a company in China and gathering information through interviews, documents, and studying organizational processes. The collected data was analysed using grounded theory. The findings were evaluated with regard to business, architecture, process and organizational concerns of a company, generally known as BAPO model. In the end, the challenges faced by the company were addressed and potential improvements were suggested to reap the benefits of software product line. The results show that the company is unknowingly using some SPL practices and has a potential to adopt SPL practices fully.

**Index Terms**—Agile development processes, BAPO model, empirical study, grounded theory, software product lines.

## I. INTRODUCTION

Software Product Line Engineering (SPLE) is an idea of building and managing a family of products with shared commonalities, targeting a certain market area and is developed from a common set of core resources. SPLE is therefore beneficial for a company developing a software suite with common or somewhat similar features between the products. To manage this product line, the commonalities are first identified and modularized. This commonality has certain variational points separating different products. There exist possibilities to derive new products from the platform by creating new variabilities. Thus, decreasing production cost, time to market, increased product quality, and exploring new market ventures could be achieved by reusing such variations and commonalities in a proper manner [1]. Usually, managing commonality and variability between different products are conducted using systematic variability management techniques [2]. Such techniques help achieve a number of benefits including defining the sources of variability and the different variants, tracing variability from the model to the code and vice versa, and communicating variability to the different stakeholders [3].

The SPLE is proactive in nature and often structured to facilitate two separate roles: domain engineering and application engineering [4]. Domain engineering concerns the commonalities in order to build and manage reusable or core assets such as requirement, design documentation, code and test artifacts. While the application engineering role works with the development of software products or applications using these core assets [5]. Although SPLE can reduce cycle time in developing product variants, it involves significant upfront commitment in exploiting a flexible product platform [6].

Software product line engineering is one of the main practices in software engineering industry. Transitioning a project from traditional software engineering approaches to software product line engineering requires strategic decisions [2], [7]. Several studies [8]-[11] reports on how large companies transitioned towards a product line approach. According to these reports, for product line adoption success, a strong management commitment is needed as well as a long term vision, which is generally lacked in small companies. It is normally believed that small companies cannot afford the start-up cost, burdensome process, strict organizational rules and the tedious planning; a core prerequisite for adopting Software product lines [12]. Furthermore, small companies lack the same resources and time to plan as a large company with regard to adopting new platforms or strategies.

In actuality, SPL offers small companies with their best hope of success [13]. Time to market and economy for production, both benefits of SPL, are more critical for a small company than for larger ones. Of course, large software houses would like to cut costs and get products to market sooner, but they have the advantage of having their customer wait for new product announcements [13]. However, small companies, for the most part, have neither a global reputation nor the expectation of a global size volume as large companies and sometimes it is hard for them to survive in a market. These companies work in an ad-hoc manner which is time, resource and revenue consuming. Their road to success lies in turning out products that are impossible to customise, in an impossibly short time, using an impossibly small staff.

In a dynamic market environment with stringent time to market constraints, small software development organizations need agility in handling changes. Agile methods such as extreme programming (XP) promote iterative development, frequent interaction with the customer, small and frequent releases, and rigorously tested code [6]. The most important characteristic of agile development is its adaptive, responsive approach to requirement changes. Although much of the literature and industry practices have focused on applying either SPLE or agile methods in isolation, few studies have

Manuscript received October 14, 2014; revised May 12, 2015.

N. Nazar is with School of Software, Dalian University of Technology, China (e-mail: najamnazar@126.com).

T. M. J. Rakotomahefa is with the School of Civil Engineering, Dalian University of Technology, China (e-mail: ymamesoj@outlook.com).

examined whether these two seemingly contradictory methods can be integrated [10], [14], [15]. Most SPLE practices relate to a product family's long-term strategic objectives, whereas agile practices focus on short term tactical objectives of developing a single product. Integrating both practices balances the ability to quickly handle changes and maximize product variety [6], [10].

Software designed in a typical small company often has similar user interfaces and may also yield repetitive functionality in some cases. The market traditionally has many companies delivering the individual solutions with wider supply of different solutions. Companies expand as both demand and revenue increase, thereby creating more potential for investment. This is done by compromising their current flexibility or efficiency. It has been noticed that sometimes small companies have similar products but are not developed from a common platform, using product families unknowingly [8]. As such, instituting a companywide overview must first be carried out to identify how much the company has an ability to use its resources and tendency of adopting software reusability properly before adopting some concrete steps to fully implement product line practices. For this purpose, we have considered four perspectives related to the Software Engineering (SE) concerns of Business, Architecture, Process and Organization (acronym BAPO).

Van der Linden *et al.* [12] propose four-dimensional software product line evaluation framework based on the BAPO concept of operations that incorporates the business, architecture, process, and organizational aspects. This framework provides a preliminary foundation for a systematic and comprehensive strategy for a process maturity evaluation of software product line engineering. Besides the BAPO model, we have evaluated if Agile methodology and SPLE work together. Preliminary experiences and studies have indicated that this combination of approaches to software product development is both feasible in practical terms and, more importantly, that it, to some extent, have improved the organizations' ability to evolve their line of software products, meeting the demands in an increasingly dynamic marketplace [10] with rapid technological shifts.

In this paper we have conducted an empirical study in a small company in China, to investigate its potential to adopt SPL. The company provides security solutions. We have spent some time at their facility and gathered the required information by conducting interviews, analysing documents, and observing processes. Later, the findings are analysed, and issues and challenges faced by the company in each BAPO dimension are discussed. Besides, we evaluate if some practices from Agile methodology could be adopted along with SPL practices for the company. In the end, we provide some immediate improvements in all four dimensions of BAPO model.

The study is conducted in light of the following research questions:

- 1) The Applicability of the BAPO model in evaluating the company's potential to adopt SPL practices.
- 2) What are the general issues and challenges faced by a small company, while making a transition towards product lines with some potential improvements?
- 3) Is Agile Development along with SPL suitable for the

company in this case study?

The rest of the paper is organized in following sections: Section II explains the related work and research projects on the integration of SPL practice in industry. Section III gives a brief overview about our research context, providing information about the company, how data is collected and analysed, and the limitations of our study. It also mentions the BAPO model briefly while section IV presents our results using the BAPO model. Section V discusses the challenges and issues faced by the company and some major improvements for it while Section VI gives the conclusion and future direction of our work.

## II. RELATED WORK

The transition to SPL has increasingly become a noticeable trend in the industry and literature is available on success stories of transition towards SPL [14]. The literature review shows that extensive research is being conducted in almost all major industries belonging to wide variety of domains, such as consumer electronics, telecommunications, and avionics to information technology, on SPL adaptation.

Clements *et al.* [9] reported that software product line engineering is a growing software engineering discipline, and that organizations such as Philips, Hewlett-Packard, Nokia, Raytheon, and Cummins are using SPL to achieve extraordinary gains in productivity, marketing time efficiency, and product quality. Linden [16] presented that the ESAPS, CAFÉ and FAMILIES projects are initiated by the cooperation among different European universities and enterprises. ESAPS project investigated the development process, and variability management from an architecture and quality viewpoint. The CAFÉ project introduced more business concerns, requirements, asset management and testing. The FAMILIES project consolidated the results from CAFÉ and exploited the ESAPS findings, improving and automating them through model driven family engineering and reusing over family borders. Family Evaluation Framework (FEF) is one of the consolidated results of FAMILIES. The purpose of the FEF is to evaluate the performance in software product family engineering of larger or smaller parts of companies including business units, divisions, and even complete companies [16]. Later, the PRAISE project was launched to investigate the process issue of software family engineering [16].

Hetrick *et al.* [17] presented their experience of transitioning to software product lines which was carried out in an incremental manner in order to avoid the typical up-front adoption barrier. They asserted that the transition required tackling a number of issues including technical issues (e.g. consolidating core assets, quality assurance) and non-technical issues (e.g. team organization, processes). Griss [18] identified a number of factors that need to be taken into consideration in order to enable the success of software platforms such as business drive, architecture, process, and organization.

M. Matinlassi [19] performed a comparative study for Software Product Line design architectures and introduced an evaluation framework. This evaluation framework was

introduced for comparison between five design methods COPA, FAST, FORM, KobrA and QADA. Sellier *et al.* [20] proposed an IDEAL model of SPLE transition for the development of Embedded System for medal processing lines. G.K. Hanssen and T. E. Fægri [10] presented a case study for a company that has integrated practices from SPLE and agile software development (ASD). SPLiCE, a software product line system for health care industry, proposed by Gomes *et al.* [21], weaves together the clinical data model, developed in Acme language using model driven engineering approach to achieve software reusability.

In the field of aviation, Sozen *et al.* [11] offered a scheme to manage the complexity of avionics software systems through variability management tools. Similarly, Fant *et al.* [22] presents a practical solution to a real life industrial problem in the unmanned space flight software (FSW) domain using software product lines and software architectural design patterns.

Looking at the literature review above, we can clearly infer that most of the efforts of software product line transition has been conducted for large companies and very little effort has been expended on small companies to evaluate their maturity towards product line adoption. Also, to the best of our knowledge no work has been done on using BAPO model for evaluating company's potential to adopt SPLE practices. Betz *et al.* [23] has used the alignment of BAPO perspectives in a software product development context. But we are using it for evaluating the company's maturity and tendency towards SPL practices. Therefore, our work is unique and substantially different.

### III. RESEARCH CONTEXT

#### A. Company Background

Our research has been conducted in a software company located in the Dalian Software Park (DLSP), Dalian, China. To comply with the non-disclosure agreement signed with the company, we have used the pseudonym 'Boyun' to refer to the company thereafter. Boyun design solutions that protect enterprise information assets, corporate applications, and other network resources, specializing in encryption content security and secure access technologies. Boyun mainly sells separate products in two different categories; data at rest and data in transit. Boyun was founded in 2003 in order to better reflect the expanding range of IT security solutions. Since its inception, it has gradually grown into a company of 40 employees; about half of them work in an engineering division (software development). Besides, engineering Boyun has Marketing, Sales, Professional Services Team (PST), and Product Steering Group (PSG) teams.

Though small in size, it is considered as one of the few influential companies working in the area of security software outsourcing in China. Their product portfolio has a wide variety of products focusing on email, file and folder, hard drive, USB flash drive encryption, VPN encryption etc. Being located in Dalian, not so far from Japan, most of the small companies operating in Dalian regard Japanese as an operational language, because of the vicinity, along with Chinese (Mandarin) as a main communication language.

However, some companies also use English as main operational language. As an exception, Boyun has both English and Japanese clients. It is active in computer security or more specifically the market for Data Leak Prevention (DLP).

These DLP solutions are developed for use in finding and preventing unauthorized people from gaining access to vulnerable information. The solutions monitor, secure/encrypt, filter and block outgoing e-mail, instant messaging, file-transfers, and other type of message traffic. DLP solutions mainly manage internal threats, in difference to security for external threats (virus, intrusion etc.). Recently, the company driven by its new business strategy planning to offer Security-as-a-Service (SaaS) for organizations that choose to adopt effective and affordable cloud computing services and VPN for Android Phones (currently at planning phase). By 2017, Boyun visions to be one of the leading providers in global Data Leak Prevention, and ultimately become the market leader. They also plan to expand their business in European markets targeting Germany, Sweden and other European giants in DLP.

#### B. Data Collection

In our research, we have used the ethnographic approach of collecting information [24] which involves spending time in the field to make first-hand observations. In order to obtain the holistic view of the problem under investigation, we employed the methodical qualitative approach of collecting and analysing data [14]. The data collected over the course of the study including observations, questionnaires, interviews, and reading process documentation, brochures and marketing pamphlets.

The authors have spent over 3 weeks at company as a consultant in gathering information, understanding the processes, and introducing software product line awareness among teams. During these visits the company agreed to give presentations about its inception, products and processes and providing interviews, filling questionnaires, sharing process documentation, sales brochures and marketing pamphlets. Furthermore, annual reports of company, software documentation, hand-outs, data sheets, and sales and marketing brochures were provided. Also, some external factors such as market orientation in Data Leak Prevention products and competition among different competitors in the same area were considered because of the subjective context in which the data was acquired. These documents gave us a good overview in understanding current state of the business and organization, sales and marketing approaches at company and its future direction.

Over the course of study, to form a broad view of an organization and its processes, eight in depth interviews with individuals (two interviews each) from different teams and at different ranks were conducted. The interviews lasted on average between 2-3 hours with 15 minutes break after each hour. In the selection process of interviewees, the goal was to get a sample of individuals that covered different aspects related to our research interest. A wide range of roles (representatives from each teams), two of which were involved in product aspects while remaining on sales, marketing, and administrating company were interviewed.

The interview questions were organized keeping the business, architecture, process and organizational perspectives of company in mind. Product/process managers were interviewed for more architecture and process oriented questions, likewise, sales and market managers for the organization and business questions, since they are directly related to the business domain of the company. These interview proceedings were recorded using audio media and crucial points were noted down.

The interviews provided us in depth information as well as detailed overview about the company in all aspects. For the first couple of interviews the main focus was to acquire a detailed insight about the company, its products and functions. For this purpose, a semi structured interviews were conducted. In a semi structures approach main focus lies on areas of interest based on basic knowledge about the company and to discover more questions later for further interviewing. The other reason for adoption of a semi structured approach is that it offers the interviewer the opportunity to explore issues or services in details, giving a deep insight about what is being searched. It covers questions such as: *How does the Engineering team work? What are the major products there? What roles and responsibilities they have?* etc. This allows the conversation to flow where it needs to in order to deal with issues, as opposed to cutting someone off because they stray from the topic.

For remaining interviews, we adopted more structures approach as compared to semi-structures previously. Question were more targeted and well prepared. The CTO of Boyun was interviewed for more architecture and process oriented questions. For organization and business questions, a Corporate Account Manager was interviewed because of his experience related to the business domain of the company. While performing interviews, taking notes or recording the actual conversation were considered. As the company wanting to remain anonymous, recording of voice could be problematic, however, later company agreed on it. Although making notes is a time consuming process and there are chances of missing or losing the important information was high. Therefore, after each meeting we considered sharing the meeting minutes with the interviewee to minimize the risk of losing some important information. During interviews, the interviewees were also asked to explain certain aspects of the platform and draw diagrams and figures to illustrate their understanding of the overall architecture, if necessary. The artifacts produced by the interviewees helped the researchers understand the problem and context better and revealed important issues underlying communication within and across teams.

The interviews are a great way of getting different perspectives of certain areas of interest. On the other hand, the interviews are dynamic and good results from it depend much on the interviewers, interviewee, and the time spent on interviewing. To complement the information from the interviews we used documentation for further understanding the research area. Documentation is a wide subject and could be helpful in analysing information and drawing conclusion well. The documentation can be detailed in certain areas, like testing specification, and it can also be a general overview of the company like handouts, brochures, roadmaps, and

organizational map etc. Therefore, we were granted access to the documented material communicated among the upper management and teams to obtain a better understanding of the company's vision, strategy, and working.

As a whole, data from the interviews, the documents, as well as the researcher's observations (diaries and audio recordings) are used to complete the information collection process. The gathered information was compared together to overcome the triangulation problems, where information from a certain perspective did not coincide with reality. For instance, a company often has an external view as well as an internal. A sales manager often provide the external view for the customers while the brochures provides internal information. The data collection phase was stopped when we started to get no new insights from interviews and documentation.

### C. Data Analysis

The collected data was analysed using Grounded Theory. Grounded Theory is a qualitative research method in which generation of a theory occurs by looking into the collected data for patterns and concepts [25]. We started by reading notes/diaries and listening to recorded tapes of interviews. In the next step we analysed the documentation provided and extracted the useful information from it. Furthermore, we used our knowledge and experience in the software product line to further strengthen our analysis. Moreover, we thoroughly checked the company's website in order to verify if we have missed any useful information. This analysis strategy provided us with important details about the company and highlighted the problems encountered by them. Next, we applied the business, organization, architecture and process aspects (BAPO) of software concerns to evaluate whether current practices at company correspond with the product lines approach. In the end, we compared our findings to the existing literature in relevant research areas in order to identify similarities and differences.

**BAPO Model:** BAPO is a four-dimensional evaluation framework for software product family engineering. It evaluates how well the company is using product lines on a five level scale in each of the dimensions. In general, these dimension are interdependent and interrelated to each other, consequently, if one dimension is changed, other dimensions will also be affected. A high value indicates a company working according to characteristics for SPL, proven to be effective, in a specific dimension. Lower value means the company does not resemble characteristics of SPL. Following are the four main dimensions of BAPO model [12], [23]:

- 1) **Business:** How to make profit from products and services.
- 2) **Architecture:** Technical means to build the software.
- 3) **Process:** Roles, responsibilities, and relation- ships within software development.
- 4) **Organization:** Actual mapping of roles and responsibilities to organizational structures.

As greater detail about BAPO is out of the scope of our paper and cited resources explained the model in details However, in Section IV, we briefly state each dimension and evaluate the company in the four dimensions of BAPO to verify its tendency towards SPL practices. Fig. 1 illustrates the four dimensions of BAPO family evaluation model.

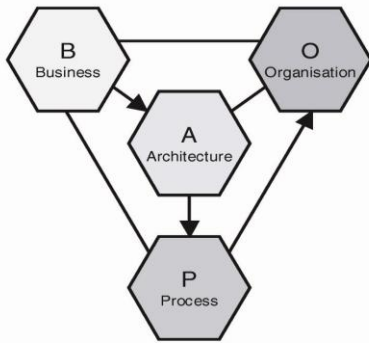


Fig 1. The different dimensions of BAPO model [12].

#### D. Limitations

This study subject to following limitations:

- 1) This is a single case study, which naturally affects the generalizability or external validity [26] of the conclusions, yet there are rational reasons for choosing such an approach. Focusing on a single case means that the study can be more thorough than a study of multiple cases, with respect to available resources. Yin [26] discusses the single case study design and presents several arguments in favour of choosing such a design. One of these is particularly applicable to Boyun, namely that it is a unique case. According to Yin, such a study may act as a prologue to further studies of a relatively new topic.
- 2) The collected data comes from single respondent interviews with internals and externals. This type of data can be potentially biased, incomplete or even wrongful due to misunderstandings, lack of insight, etc. We have sought to address this threat by collecting data from various respondents and supplementing these data with documents and observations.
- 3) The third limitation concerns the completeness of the study. Only a subset of the employees was contacted. Likewise, relatively few samples of all available documentation were collected and analysed. This is due to natural limitations such as limited time and resources.
- 4) The fourth limitation considers the applicability of the findings and conclusions of this study. The organization investigated is a small size organization and results do not necessarily apply to all other types of small software companies.

## IV. RESULTS

We have divided the results in four perspective of BAPO model, which are useful for understanding if company has the potential for adopting Software Product Lines. We have also used the snippets from our interviews as a support for our results. In the following subsections, we state our findings from our analysis (Section III-C) and classified them under the four dimensions of BAPO model briefly.

#### A. Business Perspective

By “business” we generally refer to many aspects involved in running a profitable organization including the organization’s identity, vision, objectives, strategic planning, sales and marketing, and competition.

Boyun launched their first product in 2003, an email encryption system, and it became successful in market because of its user friendly interface. Following its success they have launched more products, now part of a single package called Simple Encryption Platform (SEP). The SEP is designed to empower end users to secure data easily. Whether information is in transit; email, flash drives, portable media, or stored on a corporate network or SharePoint Intranet, SEP can secure it. By 2017, Boyun wants to be one of the leading providers in Data Leak Prevention Systems.

“...The concept we sell is DLP, data leak prevention...”

Boyun’s marketing strategy is to provide product information by using several channels i.e. company webpage, partners, direct mailing, trade show, email or physical letters, and telesales-persons that call existing or prospective customers. In one of the interviews the CTO mentioned:

“...We have webpage, market channel, we have partners, salespeople call new/old customers. We have done direct-mailing campaign, we sent out emails or physical letters...”

Products are available in following three main user groups; major enterprises, small and medium sized companies, and home users. The company reuses the single-encryption platform (SEP) for everything from planning to development. By doing this, everyone has the same understanding of the product. The company always urges the customers to purchase the complete software suit based on their SEP, especially the large enterprise. This shows that company is proactive in nature.

#### B. Architecture Perspective

By architecture we mean the construction of platforms and systems. Here, we have covered following aspects such as if the company has product family architecture? What is the level of reuse? What are the variations in system and quality?

Architecture is the most important information and behaviour inside a given organisation. In the case of Boyun, the system architecture is not clearly specified but occasionally utilised and they reuse most frequently used software artifacts among projects. As reusability is one of the most important concepts of SPL, it is important for checking the SPL transformation of company. Boyun prefers to build up a component from scratch and then work on adding different security layers. During these layers they reuse development modules such as XML files, and mainly this reuse occurs in the same layer (network encryption and network platform layers) between different products. As mentioned by the CTO, the management of reuse is not very well structured or planned.

“...Core data objects, XML parser, the framework and layers that we reuse...”

There are modules differentiating from product to product depending on customer needs. On the platform layer, it is easy to find commonalities and variabilities among products as there are slightly different modules for different products. Before, they used to have a product cycle of one year which they admit was too long for a software project. Therefore, they shifted to a shorter product cycle of one month where they focused on less features each time.

“...We had release cycles once a year, which was really

heavy. Now we've turned to small releases, one each month, with content scope..."

Similarly, testers were dedicated from development team before, but later the testing approach was changed to more collective way focusing mainly on unit and integration testing. In addition, we recommended them to shorten the product life cycle and give more coverage to maintenance and testing.

Thus, we can clearly see that Boyun has a standardized infrastructure for architecture but lacks clear reuse.

### C. Process Perspective

Process deals about how the system works and other aspects related to predictability, repeatability, and measurement matrices of system.

Boyun has a predefined process for development at certain level, for instance at development but sometimes it lacks consistency. Sometimes they experimented with different methodologies and approaches for achieving goals. Currently, the company is automating the tasks, thus, moving towards achieving predefined process consistently.

In general, the processes are quite predictable, but lack consistency. The processes are carried out in different manners depending upon the requirements of the project, especially in small enhancements or changes. Mostly the working style is primarily repeatable but not predefined, sometimes it is changeable due to the fact that different projects have different pros and cons and have to be carried out slightly differently in order to avoid problems.

Processes are not improved based on quantitative data in an established way. There are no tools or measurement, except customer data features i.e. feedback from customers.

"...No tools or measurement, except customer data features. We're improving here..."

As data from past projects is available, this could be used for improving product quality more systematically in future. Hence, we can clearly say that the process is partially managed and improving from time to time.

### D. Organization Perspective

Organization perspective deals with the mapping of roles and responsibilities to organizational structure touching mainly the geographical distribution, culture, roles and responsibilities, and product life cycle entities. As mentioned in section III-A, Boyun is a small company, having offshore engineering operations office in China with sales and marketing departments in Japan and United States. The CTO says this about geographical distribution:

"...We have developed here, even engineer bases here. Only salesperson there but everything is here..."

In general, Boyun is divided into the following major departments/teams: 1. Engineering Team, 2. Marketing Team, 3. Sales Team, 4. Professional Services Team (PST), 5. Product Steering Group (PSG).

**Engineering Team:** The Engineering Team is a main part of the company. The main tasks of Engineering Team are the development of products and applications. It also provides product maintenance if needed. They are also responsible for licensing servers, web shops, and making Intranet modules. In total, it has 21 members.

**Marketing Team:** This team consists of three permanent members. Other members are added if required depending on

the situation of the company. This team is mainly responsible for making templates, documents, PowerPoint presentations, brochures, literatures and pamphlets for presenting to customers in meetings and various other events.

**Sales Team:** The Sales team is responsible for making a list of their current customers and maintaining contact with them. They also search for new customers for future tasks and projects keeping the sales process active. After selling a product, they have to keep in touch with customers for maintenance and support activities.

**Professional Services Team:** The Professional Services Team consists of three highly professional members, who work on more technical parts like installation of systems, answering technical questions from customers, educating employees and end users regarding the upgrade of products, future aspects and other activities. PST also handles the internal Intranet, working with Microsoft SharePoint and other servers as well as providing support by testing and evaluating software. More members are added if required.

**Product Steering Group (PSG):** The Product Steering Group makes decisions regarding development, sales, services and engineering. The company also has a rotation policy therefore, this team does not have fixed members. The main purpose of this policy is that everyone should be specialized or flexible in all domains. As described by CTO:

"...The development seems to be team-based, but we are creating more and more cross-teams like PSG that makes decision about development and is a group of sales, professional services and engineering..."

As per BAPO-O they are Business Line Oriented- having well established units with slight dependency among them.

## V. DISCUSSIONS

The results from Section IV revealed a set of issues and challenges that small, slightly distributed, and agile oriented organization likely to face, when determining the potential of adopting product line practices. We have classified the challenges in four main categories of BAPO model, business, architecture, process, and organizational challenges. In the following subsections, we discuss each of the four categories briefly and provide some immediate potential improvements in more details.

### A. Business Challenges

Our findings from Section IV-A show that there are two main issues in Business dimension that can introduce major challenges, such as, the business strategy and product development.

**Business Strategy:** Boyun's new business strategy is targeting a new segment of customers and developing SAAS based applications to improve and fulfil the security requirements. The services that had been previously provided to clients (Home, Medium and Enterprise Users) by the common platforms (SEP), needed to be adjusted or customized in order to accommodate new scenarios those products were required to support (e.g. increasing security demand). This requires a considerable reengineering of some existing components (XML files and security protocols). Although this issue is not specific to platform centric

development, the approach of Boyun shows that some components are reused among platforms. The reason is that products rely on these components might need some modifications. As in Section IV-A, by 2017 Boyun visions to be the leading security provider in market, change in business strategy is important to consider for achieving this goal.

*Product Development:* In traditional models of building platforms, a platform philosophy is dominant as evident in practices like software product line engineering, where there is an emphasis on developing domain and application artifacts [5]. This succession means that an organization does not start building products until development of the platforms underlying these products has made considerable progress. On the other hand, in Boyun, we have noticed that platform development is product driven in the sense that SEP is reused in all products and customized to fulfil the requirements of an ongoing project. As explained by the technical leads (in Section IV) the main purpose of adopting this strategy was to a) reduce the investment cost and (b) achieve a faster return-on investment by delivering products to end customers more quickly. However, our findings show that this statement contradicts with the practice they have adopted. Although the SEP is reused, other parts are developed from scratch, thus increasing the reengineering costs at upper layers.

#### B. Architecture Challenges

Architecture is quite expensive and the developers do not appear to have the resources for a big architectural change, more than the one being established for the moment. As the big improvement being made by the new platform for architecture, we believe that the establishment of domain engineering units will reduce the cost for this transition.

*Establishment of Domain Engineering Units:* One of the improvements we would like Boyun to consider in business perspective is to establish a domain engineering unit. Generally, a small company with a restricted amount of developers wants their developers to work on current applications to achieve profit sooner instead of reusing core assets. They fear that domain engineers may lose focus on products and merely highlight reusable assets which may not be generating profit for the company. Instead, they lose themselves in aesthetic, generic and useless abstractions [27].

Despite this scepticism, the domain engineering unit has advantages for a company. The core assets used for reusability assets will be better managed and as a result, it improves the efficiency of updating the software suite. And with a more managed platform, the companies will be able to research and derive new product more easily, as the market requires bigger software portfolios from companies nowadays. But with an improved platform, deriving products from reusable assets with only small variability will be possible. This will help in increasing company's product portfolio and hence making them bigger competitors in the market. Even if the domain engineering does not increase revenue when first established, in the future, it can generate huge profits [4]. We have noticed that without sufficient domain knowledge, engineers cannot make decisions as to what is common and what is variable in a given component. Thus, a good understanding of the domain is vital in developing useful and reusable components for Boyun [14].

#### C. Process Challenges

The process dimension is more or less ad-hoc (creative chaos) at the moment but improving. However, by establishing XP and SCRUM practices from agile development processes, development efficiency can increase. One way of doing this could be by collecting data from previous projects to gain more insight into those projects and employ this knowledge for future projects. A process discipline is more critical for SPL. However, it should not really matter which it does not really matter which discipline (extreme programming, agile methods, scrum etc.) a company decides to use, as long as one is chosen. If not, there have been cases where the product line effort languished [28].

*Operational Processes:* We suggest them to organize operational processes first and then move to adopt Agile practices. Such operations cover for day-to-day activities such as support, training, key account management, etc. Information such as error logs, support requests and general feedback should be in the strategic process. However, in cases of urgent errors or problems, hot fixes are made to correct the problem as soon as practically possible. Such corrections do not imply a change in functionality and can easily be deployed to most customers as the solution is operated as a service. Customers that operate their own installations naturally need to update the system locally.

*Separation of Product Concerns:* After accomplishing these operational improvements Boyun could make two distinct refinements in their process. On one hand, they should introduce the separation of concerns of building a robust platform, and effectively building products based on that platform, on the other hand. This will ultimately make Boyun's product line developed and released as a single unit. The organization will manage to handle both the development of core assets and applications simultaneously [29]. This could be achieved by creating next release corresponding to application engineering, and developing the roadmaps corresponding to domain engineering by PSG team. As Boyun is not doing domain engineering as a separate or dedicated process, in the sense that they produce domain artifacts that are to be used later in a separate application engineering process. The best way could be that the artifacts are developed based on identified needs and ideas coming from the roadmaps, but with constant corrective feedback from agile development iterations, making it a reactive approach in the end.

#### D. Organizational Challenges

A wide range of issues and challenges arise due to the nature of platform development that requires participation and involvement at the organizational level. In the following subsections, we discuss the organizational issues we encountered in the data. Our findings show that, from an organizational aspect, Boyun has issues in communication between team, motivating them, and decision making among and across the teams.

*Motivation among Teams:* In the case of Boyun, one of the main challenges is to motivate the individual teams to talk to each other beyond formal meetings through proper communication, where things might have been overlooked or misunderstood. When this motivation is not there, developers

resort to hunches to resolve a dependency or may integrate with other components in a less than ideal way. Besides, where this communication is not effective, teams may work on overlapping areas of the platform causing redundancy and rework as we have observed in the company.

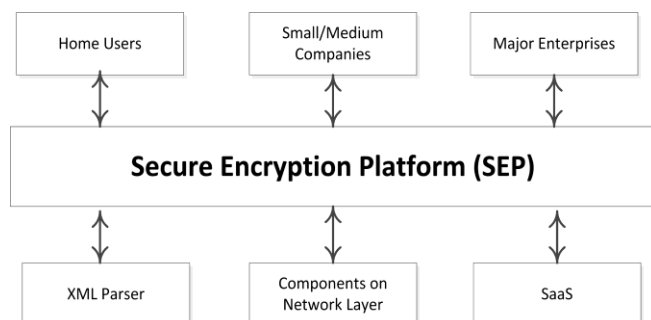


Fig. 2. The Reusability components in Boyun. Three major products on top (home, small/medium and major enterprises). The components at the bottom are plugged into the SEP platform which in turn becomes a part of each product.

*Communication among Teams:* Teams in Boyun need to communicate at all levels. This is because of the fact that the engineering team provides services that are consumed by other teams. The product team needs to know the working and integration of platform. They should also be responsible for providing feedback to platform team about existing and missing features. Also, product teams provide feedback to platform teams on existing features and report missing ones. As one of the developers pointed out:

“...Achieving this communication, however, can sometimes be tricky...”

When an issue has arisen in product development, some developers found it easier and quicker to find workarounds. This not only caused a lot of rework and redundancy in the code, it also made testing and maintenance cumbersome in the future. Therefore, making this communication effective, the engineering team needs to understand the value of keeping communication channels active at all times (i.e., realize the technical problems associated with redundancy). In addition to discussing this issue with individuals in the organization, we also find that online communications at different locations is much trickier. As mentioned in section III-A the company has offshore offices in Japan and America as such the language barrier is a significant challenge that they have faced on numerous occasions. Different time zones, absence of body gestures and facial expressions, and lack of proper spoken skills, sometimes resulted in many miscommunications and inconveniences among offshore teams.

*Improving Decision Making Process:* At Boyun we observed that they have a problem in decision making. Generally, the process is slow and not clearly expressed between the teams. One case we came across in Boyun involved decisions that needed to be taken on whether to reuse an existing platform (or what part of it) or take a different direction such as building an independent variant to satisfy a certain business concern. Usually, the Decisions are made by sales, marketing and PSG teams regarding their respective areas and engineering team, in general, not taken in to

confidence about main decisions. This causes lack of coherence and collaboration among teams. In order to overcome this problem, a separate team consisting of one person from each team should be made. This will be a temporary team that would gather only when the company has decision making meetings, once or twice in a month. With the help of it, the decision hierarchy will become more solid and will facilitate in better decisions making in a short span of time.

*Better Roles and Responsibility Definition:* Another area which needs improvement is proper definition of roles and responsibilities among the teams. As we noticed in section IV-D that each team is working in parallel on different tasks, except the engineering team, which works on product development. The engineering team remains isolated from other aspects which creates imbalance, resulting in an inability to resolve problems in time. In addition, more members should be added to professional services team, because it is difficult for just few members to perform installation, resolving customer issues, keeping internal intranet running in short time duration. The addition of more members will reduce the workload on the team and increase the collaboration and cohesion in an organized way. The company should also restrict the writing access to the documents for the normal members in each team on MS SharePoint server over local intranet. These documents are used for the members to understand and inform about policy or decision changes. This will increase the transparency and better understanding of roles and responsibility among team members at an organizational level.

#### E. Adoption of Agile Practices

During the past decade, agile software development has gained great momentum and found its way to an overwhelming number of organizations of different scales [14]. Agile methods preach a raft of principles and provide a wide range of practices to achieve these principles. Although the initial focus of agile methods was centred on the efficiency of the team as a unit of operation. Recently, there has been a movement towards scaling agile methods up to the enterprise level [30]. In this section, we list some of the challenges towards the adoption of Agile practices in the Boyun.

*Development of Feature Based Teams:* Agile focuses more on feature development teams rather than component based teams in order to increase tangibility of product and it is generally perceived that component teams are disadvantageous [14]. As described in Section IV-D that different teams are performing overlapping tasks. This practice yields that the Boyun can introduce feature based team within engineering department to improve development and testing.

*Rotation Policy:* From the interviews we found an important aspect for Boyun to consider is the rotation of employees. The roles of the employees can be rotated to give broader experience and knowledge to the employees. For instance, a software developer could work as software tester at the same time, under the Agile environment as engineering units resides at the same place. The company does not have a well- defined process but uses experience and interaction while developing. As all the developers working in the same



room as a single development unit, this allows flexibility, interaction and knowledge sharing while developing.

TABLE I: CHALLENGES AND IMPROVEMENTS

Challenges	Potential Improvements
Business Challenges	Business Strategy Product Development
Architecture Challenges	Establishment of Domain Engineering Units
Process Challenges	Improve operational processes Separation of Concerns
Organizational Challenges	Team Motivation Team Communication Decision Making among Teams Definition of roles and Responsibilities
Agile Culture	Feature Based Teams Rotation Among Teams Planning Game Component Reuse

As all the developers work in the same room as a single development unit, this allows flexibility, interaction and knowledge sharing while developing. The Agile way of working is often less strict and more flexible than using well-defined processes with lot of documentation. The feature of individual and interactions over processes and tools [31] makes Agile perfect for small to medium sized companies where groups are small and are not distributed over larger areas.

*Planning Game:* The planning game concept from the Agile approach can give benefits for a product line approach by enhancing the agility [32]. This method ideally contains a customer representative sharing the customer point of view on what features the product needs. It has also been shown that using the most sensible concepts from agile, product line version of the planning game can be efficient and usable. The process relies on face to face communication which seems to be a typical result when merging agile with other approaches [33]. The planning game is useful for organizations not distributed at different locations (or participants onsite) [34]. In case of Boyun, the engineering unit resides at the same place therefore, planning game fits well for Boyun.

*Reuse within Components:* Boyun mainly focuses on reuse of commonality artifacts by adding more features to create new software. To increase usability among features, Boyun should separate roles and responsibilities among teams in two steps. First, the company should establish roles that focus on the reusable assets by using SPLE approach [12]. As a suggestion, they can apply the SPL's domain engineering concept to develop these reusable core assets. Further, they can plan ahead to create a set of products families and a well-managed reusable repository by the configuration management. Next, with regard to the roles that create a single product from the reusable artifacts, we suggest that agile methods should be adopted. It is because the Boyun has faced time-to-market issues. Agile methods can increase the speed of software development and focus on immediate customer requirements [35]. This will promote customer satisfaction, who always demand new software features, thus reducing time-to-market and improving product quality.

In short, we strongly suggest Boyun to incorporate both

SPLE and Agile practices together. SPLE as a proactive, strategic reuse approach forms the basis to develop new products in less time and with higher quality. Agile methods can be used in application engineering to perform the customization or calibration of a product for a specific customer [32]. Also, not only is the combination of both process is feasible, the simplified approach makes the organization more flexible and thus capable of serving in a volatile market with fast changing technologies.

## VI. CONCLUSION AND FUTURE WORK

In this article, we have presented an empirical case study that aimed to uncover the issues and challenges associated with the transition of a small scale organization to a software product line strategy. We have evaluated the Business, Architecture, Process and Organization (BAPO model) aspects of the company to evaluate if the company has a potential to adopt product line practices. We have also presented some potential improvements in all four areas of the BAPO model which would make the organization more inclined towards product line practices. The adoption of some practices of Agile development process, such as planning game from extreme programming with software product lines should prove useful for the company. We came to the conclusion that the company is unknowingly using some practices from product lines. Our proposed improvements will further bring it in line more towards product line approach.

However, there is still work to do, which we are planning to incorporate further in future. As mentioned in section III-D, the data collection method has some limitations therefore, it needs further improvements. Besides BAPO model, more models such as Family Evaluation Framework [16], [34] need to be evaluated. Furthermore, the creation of software product line engineering maturity model for small enterprises is required to be tested further. Another challenge we plan to address is how to evaluate the product in detail for improvements, along with the further study of variability management in Boyun. Moreover and most importantly, we are waiting for Boyun to give us a feedback about our improvements which they have implemented and under ongoing testing. This will help us in defining new models for evaluation of small companies. It will also help us in further improving the research in the area of adoption of product line practices for small companies.

## ACKNOWLEDGMENT

We would like to thank the company for allowing us to spend time in their facilities to conduct this research. Also, we would like to extend our gratitude to the individuals who dedicated their time and effort to coordinate and participate in the interviews. Special thanks to anonymous reviewers for their valuable feedback on the content of this article.

## REFERENCES

- [1] K. Schmid and M. Verlage, "The economic impact of product line adoption and evolution," *IEEE Software*, vol. 19, no. 4, pp. 50–57, July 2002.

- [2] K. Pohl, B. Gnter, and F. van der Linden, *Software Product Line Engineering: Foundations, Principles and Techniques*, Springer, 2005.
- [3] L. Chen, M. A. Babar, and N. Ali, "Variability management in software product lines: a systematic review," in *Proc. the 13th International Software Product Line Conference*, 2009, pp. 81–90.
- [4] J. McGregor, J. S. Northrop, and K. Pohl, "Initiating software product lines," *IEEE Software*, vol. 19, no. 4, pp. 24–27, Jul-Aug 2002.
- [5] A. Metzger and K. Pohl, "Variability management in software product line engineering," in *Proc. the 28th International Conference on Software Engineering*, 2006, pp. 1049–1050.
- [6] K. Mohan, B. Ramesh, and V. Sugumaran, "Integrating software product line engineering and agile development," *IEEE Software*, vol. 27, no. 3, pp. 48–55, 2010.
- [7] G. Bockle, P. Clements, J. D. McGregor, D. Muthig, and K. Schmid, "Calculating roi for software product lines," *IEEE Software*, vol. 21, no. 3, pp. 23–31, 2004.
- [8] D. Benavides and J. A. Galindo, "Variability management in an unaware software product line company: An experience report," in *Proc. the Eighth International Workshop on Variability Modelling of Software-Intensive Systems*, ser. VaMoS '14, 2014, pp. 1–6.
- [9] P. Clements, L. Jones, L. Northrop, and J. McGregor, "Project management in a software product line organization," *IEEE Software*, vol. 22, no. 5, pp. 54–62, Sep.-Oct. 2005.
- [10] G. Hanssen and T. Fægri, "Process fusion: An industrial case study on agile software product line engineering," *Journal of Systems and Software*, vol. 81, no. 6, pp. 843–854, 2008.
- [11] N. Sozen and E. Merlo, "Adapting software product lines for complex certifiable avionics software," in *Proc. 3rd International Workshop on Product Line Approaches in Software Engineering (PLEASE)*, Zurich, June 2012, pp. 21–24.
- [12] F. van der Linden, J. Bosch, E. Kamsties, K. Kansala, and J. H. Obbink, "Software product family evaluation," in *SPLC*, 2004, pp. 110–129.
- [13] C. Gacek, P. Knauber, K. Schmid, and P. Clements, "Successful software product line development in a small organization. A case study," *Fraunhofer Institut for Experimental Software Engineering (IESE), Tech. Rep.*, 2001.
- [14] Y. Ghanama, F. Maurera, and P. Abrahamsson, "Making the leap to a software platform strategy: Issues and challenges," *Information and Software Technology*, vol. 54, no. 9, pp. 968–984, September 2012.
- [15] S. Urli, M. Blay-Fornarino, P. Collet, and S. Mosser, "Using composite feature models to support agile software product line evolution," in *Proc. the 6th International Workshop on Models and Evolution*, ser. ME '12, 2012, pp. 21–26.
- [16] F. van der Linden, *Family Evaluation Framework Overview & Introduction*, 2005.
- [17] W. Hetrick, C. Krueger, and J. Moore, "Incremental return on incremental investment: Engenio's transition to software product line practice," in *Proc. the 21st ACM SIG-PLAN Symposium on Object-Oriented Programming Systems, Languages, and Applications*, 2006, pp. 798–804.
- [18] M. Griss, "Software reuse architecture, process, and organization for business success," in *Proc. the Eighth Conference on Computer Systems and Software Engineering*, 1997, pp. 86–98.
- [19] M. Matinlassi, "Comparison of software product line architecture design methods: Copa, fast, form, kobra and qada," in *Proc. ICSE*, 2004, pp. 127–136.
- [20] D. Sellier, G. Benguria, and G. Urchegui, "Introducing software product line engineering for metal processing lines in a small to medium enterprise," in *Proc. Software Product Line Conference*, 2007, pp. 54–62.
- [21] A. Gomes, A. Ziviani, M. Correa, M. Teixeira, and M. Moreira, "Splice: A software product line for healthcare," in *Proc. 2nd ACM SIGHIT International Health Informatics Symposium*, 2012, pp. 721–726.
- [22] J. Fant, H. Gomaa *et al.*, "Software product line engineering of space flight software," in *Proc. 3rd International Workshop on Product Line Approaches in Software Engineering (PLEASE)*, 2012, pp. 41–44.
- [23] S. Betz and C. Wohlin, "Alignment of business, architecture, process, and organisation in a software development context," in *Proc. the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '12, 2012, pp. 239–242.
- [24] M. Hammersley and P. Atkinson, *Ethnography: Principles in Practice*, Tavistock, London, 1983.
- [25] A. Strauss and J. Corbin, *Grounded Theory in Practice*, SAGE, 1997.
- [26] R. Yin, *Case Study Research: Design and Methods*, Sage Publications Inc, 2014.
- [27] J. Bosch, "Software product lines: Organizational alternatives," in *Proc. the 23rd International Conference on Software Engineering*, 2001, pp. 91–100.
- [28] L. M. Northrop, "Sei's software product line tenets," *IEEE Software*, vol. 19, no. 4, pp. 32–40, July 2002.
- [29] G. K. Hanssen, "Agile software product line engineering: enabling factors," *Software - Practice & Experience*, vol. 41, no. 8, pp. 883–897, July 2011.
- [30] P. Knauber, D. Muthig, K. Schmid, and T. Widen, "Applying product line concepts in small and medium-sized companies," *IEEE Software*, vol. 17, no. 5, pp. 88–95, September 2000.
- [31] (2014). Manifesto for agile software development. [Online]. Available: <http://agilemanifesto.org>
- [32] R. Carbon, M. Lindvall, D. Muthig, and P. Costa, "Integrating product line engineering and agile methods: flexible design up-front vs. incremental design," in *Proc. 10th International Software Product Line Conference*, USA, August 2006.
- [33] M. Noor, R. Rabiser, and P. Grunbacher, "Agile product line planning: A collaborative approach and a case study," *The Journal of Systems and Software*, vol. 81, pp. 868–882, October 2007.
- [34] R. Carbon, J. Knodel, D. Muthig, and G. Meier, "Providing feedback from application to family engineering — The product line planning game at the testoag," in *Proc. the 12th International Software Product Line Conference*, September 2008, pp. 180–189.
- [35] Y. Ghanam and F. Maurer, "An iterative model for agile product line engineering," in *Proc. 12th International Conference on Software Product Lines*, Ireland, September 2008.



software product repositories.

**Najam Nazar** received his B.Sc (Hons.) degree in computer science from University of the Punjab, Lahore, Pakistan in July 2005 and the M.S. degree in software engineering from Chalmers University of Technology, Sweden in March 2010. He is currently working towards his Ph.D. degree in software engineering at Dalian University of Technology, Dalian, China. His current research interest includes lines, agile methodologies and mining software



**T. M. J. Rakotomahafoa** got his bachelor and master degrees with high distinction from the Physics Department of Antananarivo University, Madagascar in 2006 and 2009.

Since September 2010, he has been doing his PhD degree in civil engineering under HVAC division at Dalian University of Technology, China, where he develops a model to efficiently calculate and reduce energy consumption of an office room. He closely cooperates with School of Software while creating enormous codes in order to transform that model into software.

He has main interests in the field of simulation, programming, designing in various domains namely, software engineering, computational fluid dynamics and applied maths and physics.