Positioning software companies along the volume-variety diagonal

André Leme Fleury (DEP – POLI – USP) fleury@fl2.com.br
Mauro de Mesquita Spinola (DEP – POLI – USP) mauro.spinola@poli.usp.br

Summary
Software importance keeps growing fast and consistently for many organizations. The growth of software functionality in manufactured products and the emergence of digital media, convergent spaces including digital content, software, and multi-channels to the market, are recent examples of organizational changes where software assumed a central position for the corporate strategy.

This paper analyzes the alignment between strategic objectives and development process at software companies and proposes a conceptual framework to ensure that development practices are aligned with corporate core capabilities. The research process included the conduction of a survey.

Keywords: software strategy, software management, software development

1. Introduction

Software importance keeps growing fast and consistently for many organizations. Recent examples include the diffusion of spreadsheets, enterprise resource planning, and Internet based transactional systems. Currently, the emergence of “digital media” - convergent spaces including digital content, software and multi-channels to the market\(^1\) - impact previous market dynamics and can change its competitive basis (Gorton and Brown, 2004); the growth of software functionality in manufactured products moves innovation from hardware to software, changes research and development strategies and brings software management into the core of corporate strategies (Probert et al., 2005).

Despite the growth of software importance, the reality of the software market is challenging. After a decade of booming growth driven by euro conversion, Y2K fears, Internet “bubble” and a widespread belief in the productivity-enhancing power of IT, these days the software industry is experiencing slower growth rates (DTI 2004), and consequently, promoting the emergence of new software development standards (Eriksson and Penker 1998; Kruchten 2000; Chrissis and Konrad, 2003), consolidating industry players through mergers and acquisitions (Shankland, 2002), and experimenting with new business models (Austin, 2004).

In this new context, software developers have to guarantee the maximum return to their customers operating with fewer resources.

Much of current research on software does not consider the linkages between software market and software development. At the business level, analysis of the software market usually classifies software companies as “services providers” or “product developers”, where “service providers” develop software systems for one customer only and “product developers” develop software systems for more than one customer (Hoch and Roeding, 1999; Veloso et al., 2003; DTI, 2004). Although this categorization is useful in the analysis of software business models

\(^{1}\) According to Gorton et al. (2004), the term multi-channel is used “when the same service offering is available, to a greater or lesser degree, over more than one delivery platform or type of device (e.g. fixed Internet, TV, wireless)”
and economics, including market structure, economies of scale, sales model and localization of new product development (Hoch and Roeding, 1999; DTI, 2004), it provides little guidance for software companies when they are defining their technological strategies, because software development differs significantly from manufacturing and services processes (Sommerville, 2004). As a consequence, software engineering practice and research is done in a “value-neutral setting” (Boehm, 2003) and from a software developer perspective, different development activities are considered equally important for the achievement of corporate goals.

These observations reinforce the relevance of developing research that analyzes the alignment between corporate strategy and software development considering current and planned capabilities of the software company. This paper presents part of the outcomes of a research project addressing the question of the alignment between strategic objectives and development processes at software companies. Two main research questions guided our work:

1. How to create a consistent link between corporate strategy and software products and services, differentiating among software companies according to their core capabilities;

2. How to deploy a software development strategy from identified software products and services, prioritizing different corporate capabilities and facilitating the assessment, communication and implementation of process improvement initiatives;

To answer those questions, firstly we developed an exploratory survey (Forza 2002), and from the outcomes of this survey, we propose a categorization for software companies. Secondly, we customized the technology roadmapping technique for each main category of software company, and conducted four case studies (Voss, Tsikritksis et al. 2002) to validate the proposed categorization. This paper presents the first outcomes this research project, addressing the question of the differentiation among software companies according to their core capabilities.

2. Literature Review

Software companies are usually classified as “service providers” or “product developers”. However, software development models differ significantly from production and service process models, and these differences are caused because of software fundamental characteristics. According to Brooks (1987), software can be characterized as complex, changeable and invisible. Software entities are made of data sets, relationships among data items, algorithms, and invocations of functions. Every piece of software is unique, and it should be reused if the tasks it performs needs to be implemented again. Software products are embedded in a cultural matrix of applications, users, laws and machines, which changes continuously, and software needs to be adapted. Finally, software is not a physical artefact and is not embedded in space; therefore, it cannot be represented geometrically.

Software development improvement initiatives have always faced difficulties originated by these fundamental characteristics, which considerably increased communication requirements among team members. The two most common responses in organizations facing these communication difficulties are: (1) establish and follow more formalized production methods for building software products and (2) use teams of software development specialists and the potential positive synergy that arises from their interactions (Sawyer and Guinan, 1998).

Software development follows a life cycle (Boehm, 1989; Beck, 1999; Kruchten, 2000),
beginning with an idea or a need that can be satisfied wholly or partly by software and ends with its discontinuation. Software development includes four fundamental processes, which are common to all software projects (Sommerville, 2004). These processes include:

a) **Software specification**: methods, techniques and tools for the definition of the functionalities of the software and constraints on its operations; includes requirements analysis and definition, which should be understand and agreed by customers, users and development team.

b) **Software development**: comprehends methods, techniques, and tools to design software architecture and develop it; includes system and software design which partitions software requirements, generates the overall system architecture and detailed design, and software implementation, when software design is realized in sets of programs.

c) **Software validation**: comprehends methods, techniques, and tools to guarantee that the produced software does what the customer wants; includes unit and system tests;

3) **Software evolution**: comprehends methods, techniques, and tools to evolve the software to meet changing customer needs; each evolution cycle can be considered a small software development cycle.

In addition to development processes, software projects also comprehend a set of managerial processes that must be performed in order to guarantee that the project will be concluded appropriately; therefore, the analysis of these software management activities also provides relevant knowledge for the categorization of different software companies. According to the Software Engineering Institute (SEI 2005), “software management deals with the concepts, methods, and techniques for managing software products and projects. Software management includes activities concerned with project management, risk management, software quality assurance, and configuration management”.

### 3. Survey Methodology

Since the objective of the proposed research was to analyze software development practices among different categories of software companies using a single research instrument, and we could not identify research initiatives with similar scope, we conceived a survey aiming to gain preliminary insight on this topic, to determine the concepts to be measured in relation to the phenomenon of interest, how best to measure them and how to discover new facets of the phenomenon under study. Therefore, we realized an exploratory survey, designed and implemented based on the guidelines proposed by Forza (2002). During this process we had to deal with the following issues: how to create a common framework for the analysis of different players in this industry, which companies should participate in order to obtain a representative sample, which methodology should be used during survey application, and which methodology should be used to analyze obtained data. The first three issues will be discussed in this section, and the last one will be discussed during the analysis of obtained results.

To address the first issue and make possible comparisons among different categories of software companies, it was necessary to build a unique framework, including most relevant software development and commercial practices. Therefore, we reviewed the literature concerning software development concepts, frameworks, methods, and techniques, including the Capability Maturity Model for Software (Paulk and Weber, 1994), the Capability Maturity

From the reviewed literature, the ISO/IEC 12207 presented a complete framework for analysis, describing software development process and its related activities during each development stage, since product inception until product discontinuity. Therefore, we adopted the ISO/IEC 12207 as an initial framework, and for each activity identified in the norm, we selected correspondent practices presented by the other mentioned theories. The result was a 300 questions survey, divided into five main areas: company characteristics, software projects characteristics, fundamental processes, support processes, and organizational processes.

To validate the initial questionnaire, we invited companies with previous relationships with our University to analyze its applicability; initial respondents considered the questionnaire appropriate. We also established as potential respondents organizations listed among the 200 biggest Information and Communications Technology (ICT) companies operating in Brazil.

To find a suitable research strategy, we reviewed software development process surveys conducted worldwide. The survey “State of the Practice in Software Engineering” (Glass, 2003), presented an extensive review concerning “Software Development Practices”, The majority of the conducted surveys used web-based questionnaires. Therefore, we converted our survey into a web-based questionnaire, published it at the University web-site, and invited respondents to complete the survey questionnaire.

4. The software industry and its development practices

Thirty eight companies completed the survey process, and to analyze obtained results aiming to understand the importance of different development practices among different software companies, we clustered companies using two main variables: software purpose and commercial practices. We could not identify different categories of companies based on the commercial purpose of the software, and analysis using the number of active projects and number of active customers did not provide meaningful results. However, we identified that using the relationship between number of active customers and number of active projects to segment companies proved to be useful, and we found three relevant categories of software companies, which will be presented next section.

4.1. Characterizing Software Companies

According to the SEI (2005), software management deals with the concepts, methods, and techniques for managing a software project and its development activities. In this context, each software project can be considered a “unit of analysis”, since a complete “software life cycle”, which include software management and software development activities, will be performed each time a software project is executed.

Each software project generates one final system; therefore, the number of customers that can benefit from this project output (final system) clearly generates a perspective of scale for the results of a software project. If a project output has thousands of potential customers, corporate strategy should be based on mass-selling of this output; on the other extreme, if the number of customers that can benefit from this project output is the initial customer only, the development company should focus on selling more software projects for this specific
customer. As a consequence, the relationship between total number of customers and total number of projects in a software company reveals this company’s specialization towards customized or mass development.

Our analysis using number of customers divided by number of projects as the segmentation criterion revealed three software companies’ groups:

- **Customer Oriented**: companies with less than one customer per project, which usually develop several software projects for specific customers. The main objective is to develop software on demand for their customers; as a consequence, the stronger the customer, the bigger the opportunities for the software company.

- **Service Oriented**: companies with few customers per project. They usually work with few software systems and provide services related with these systems for their customers, including software customization, training and operation (outsourcing).

- **Product Oriented**: companies with several customers per project, usually working in the development of standard software solutions, which are commercialized with several customers. Standard software solutions may be applications, which can be used directly by end users, complex products, which must be customized before being used by end users, and software tools and components, which are used to produce other software systems.

Using these categories of products and services, we suggest that the software industry dynamics can be illustrated as presented in figure 2:

![Software Market Dynamics Diagram](image)

Figure 1: Software market dynamics
Source: Created by the authors

Based on the analysis of the outcomes of our survey, the most important characteristics of each category of company can be summarized as follows:

<table>
<thead>
<tr>
<th>Description</th>
<th>Customer Oriented</th>
<th>Service Oriented</th>
<th>Product Oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original software developers</td>
<td>Software service providers</td>
<td>Innovative mass software products developers</td>
<td></td>
</tr>
<tr>
<td>Focus</td>
<td>Customer orientation</td>
<td>Service Orientation</td>
<td>Product orientation</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------</td>
<td>---------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>Research and development</td>
<td>Customer portfolio</td>
<td>Service portfolio</td>
<td>Product portfolio</td>
</tr>
<tr>
<td>development focus</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Products / Services</td>
<td>“One-off” software</td>
<td>Software Services</td>
<td>Packaged software,</td>
</tr>
<tr>
<td></td>
<td>systems, based on</td>
<td>including customization,</td>
<td>including applications,</td>
</tr>
<tr>
<td></td>
<td>customers specification</td>
<td>codification, training, and</td>
<td>tools, components or</td>
</tr>
<tr>
<td></td>
<td></td>
<td>software operation</td>
<td>complex products.</td>
</tr>
<tr>
<td>Quality</td>
<td>Process Quality</td>
<td>Project Quality</td>
<td>Product Quality</td>
</tr>
<tr>
<td>Certification</td>
<td>SW-CMM/CMMI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Major Risks</td>
<td>Requirement changes and project duration</td>
<td>Project cost and project duration</td>
<td>Project duration</td>
</tr>
<tr>
<td>Project Management Focus</td>
<td>Management of customer expectations</td>
<td>Management of project constraints</td>
<td>Management of product quality</td>
</tr>
<tr>
<td>Training</td>
<td>Programming and configuration management</td>
<td>Programming and system modelling</td>
<td>Programming and system tests</td>
</tr>
</tbody>
</table>

Table 2 – characteristics of software development company categories.
Source: Created by the authors

4.2. Proposing a Software “Volume-Variety” Diagonal

“Volume-variety” diagonals have proved to be very useful for manufacturing and service systems research, since they provide means for unifying different manufacturing and service process dimensions in a single model by correlation against product or service volumes (Silvestro, 1999). Similarly, we propose the following “volume-variety” diagonal for software companies:

![Figure 2 – The Software “Volume-Variety” diagonal](image)

Source: Created by the authors

We consider the conceptualization of a “volume-variety” diagonal for software systems important because, from the analysis of the different categories of companies presented in this section and further inputs provided by software practitioners who were invited to validate this initial model, it became clear that a company should develop its strategic analysis based on its current and expected position at the “volume-variety” diagonal.
5. Conclusions

The analysis presented last section provides relevant considerations for a software company planning its business strategy. Based on the perspective of scale of its software projects, the company’s current market position and related core capabilities are identified; from the analysis of future market opportunities and threats and corporate strengths and weakness, it is possible for the company to plan the desired evolution of its products and services.

Considerations about the most relevant software development processes for each category of company can also be drawn from the analysis of the software “volume-variety” diagonal, and can be applied for the contextualization of this strategy.

Customer oriented companies usually develop several projects for specific customers. Because this customer has great importance for corporate revenues and a similar software system has never been developed before, software requirements are not completely understood before the project approaches its end; therefore, requirements changes continuously, and consequently, two processes acquire significant importance: requirements management, which has to deal with the presented uncertainties, and software configuration management, because changing requirements imply that new versions of different programs and sub-systems will be available, and will have to be managed. Another consideration for this category of company arises because the developed software is unique, and it is not possible for the customer to analyze its quality standards comparing its features with similar systems; therefore, software quality assurance processes should focus on the adoption of process oriented methodologies, such as the Capability Maturity Model Integration [6].

Service oriented companies experience a different reality. Since this category of companies commercializes services related with software customization, implementation, training and outsourcing, it is possible for the customer to evaluate the benefits provided by different service oriented companies and to compare the cost of each supplier. As a consequence, the level of resources available for developing the software project tends to be restricted, and project management processes acquire great importance in order to guarantee that the project will be delivered on time, budget and according to expected quality parameters. The customization process usually uses programming languages and tools developed by the producer of the software. As a consequence, the customization process is defined by the producer company, and the professionals that work in a service oriented company should be able to understand different customization processes and how the implementation of the software will impact the business of the customer.

Finally, product oriented companies have another set of important processes. Since the final product will be packaged to be delivered directly to the end user or to be customized by another company, the product oriented company has to allocate part of its resources to guarantee that the product has reached desired quality standards, creating teams of software testers and adopting formal test procedures. The software quality assurance process has a different focus, and should be able to guarantee that produced software meets the standards proposed by norms such as the ISO/IEC 9126 (ISO 2003). Another important consideration for this category of companies is based on the fact that the knowledge embedded into the software will evolve, and therefore, requirements management should considers the evolution of these requirements over time, and the programming process should present capabilities to implement the forecasted requirements. Finally, documentation processes are also quite relevant for this category of companies, because documentation will be used as the basis for
eventual software customization and to support software use.

Those considerations apply for the company in the present and in the future; therefore, if a company positions itself as customer oriented in the present and wants to position itself as service oriented in the medium term, it should incorporate the presented service oriented considerations into its strategy.

6. Acknowledgements

This work has been funded in part by CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico, a Brazilian Governmental Agency.

References
BECK, K. Extreme Programming Explained, Addison-Wesley, 1999
BOEHM, B. Value-Based Software Engineering. ACM SIGSOFT Software Engineering Notes, 2003
INTERN.STANDARD ORG. ISO/IEC 12207 - Information Technology - Software life cycle processes, 1995
INTERN.STANDARD ORG. ISO/IEC 15504 - Information technology - Process Assessment, 2004
PAULK, M. & WEBER, C. The Capability Maturity Model: guidelines for improving the software process, Addison-Wesley, 1994
PROBERT, D., HUNT, F., FRASER, P., FLEURY, A.L. Developing software content for manufactured products: inside or outside the firm? R&D Management, 2005
SOFTWARE ENGINEERING INSTITUTE. Software Engineering Body of Knowledge, 2005
SHANKLAND, S. IBM grabs consulting giant for $3.5 billion. CNET News.com, 2005.

SOMMERVILLE, I. Software Engineering. Addison-Wesley, 2004
