4Ps of Business Requirements Analysis for Software Implementation

Mingtao Shi
FOM Fachhochschule für Oekonomie & Management
University of Applied Science
Bismarckstr. 107, 10625 Berlin, Germany
Tel: 49-171-2881-169 E-mail: Consulting_Shi@yahoo.de

Abstract

Introduction of new software applications to achieve significant improvement of business performance is a general phenomenon that can be observed in a variety of firms and industries. While carrying out such complex activities, firms are frequently struggling with quality and time, which, as this paper argues, can be achieved by basing the implementation upon 4Ps of business requirements analysis: Process, Product, Parameter and Project.

Keywords: Requirements analysis, Software implementation, Process analysis, Product analysis, Parameterisation, Project management

1. Business Requirements and Software Implementation

Industrial firms today are achieving significant scale and scope advantages by introducing new software applications tailored to firm-specific value chain activities. The pervasive deployment of software and burgeoning growth of specialist vendors has fostered the emergence of industrial applications based upon core systems that can be individualised by parameterisation and customisation. Flexible core systems have become the general pattern of dominant software products in a variety of industries. This trend is especially favourable for small and medium-sized firms that necessarily concentrate on software application rather than on software production. These firms are typically in possession of an own low-budget IT department or are outsourcing their software-related activities to external system integrators or software consultants.

The internal IT department is organically integrated and may be provided with business knowledge quickly, but it is in most cases too small to conduct software development from scratch for comprehensive business activities. The external IT specialists may have remarkable software knowledge, but it is organisationally more difficult for them to assimilate business information of the potential software user firm. Core software systems that can be parameterised and customised smoothly fit in such business and technical scenarios that are common in a wide range of industries, such as wholesales, logistics and banking.

How to implement such core systems in accordance with the firm-specific business requirements therefore has become an area of common interest for practitioners as well as for academicians because of its huge financial implication. Most of such software systems are expensive. Software itself must be paid. Hardware must be purchased to accommodate the software. Personnel are to be trained for configuring and using the applications, most probably, on a short-term run. Furthermore, unsuccessful implementation would lead to inefficient or even insufficient business performance, causing further unmanageable costs. Under such circumstances, software requirements analysis for system implementation undoubtedly is crucial for a firm to further prosper or even survive in the marketplace.

Classical approaches in the area of requirements engineering are rigorously defined and extensively discussed in the literature. Probably the techniques advanced by Pressman (2004) and Pressman (2009) are most systematic, which aver that requirements analysis centres on a few key elements, including scenario-based, flow-based, behaviour-based, and class-based system views, and the data modelling. Other authors have argued more or less in a similar manner (see Wiegers, 2003; Robertson & Robertson, 2006; Pohl, 2007). However, these rather technical methods are beneficial for software development from scratch. On the one hand, most of small and medium-sized business firms lack the capability or are reluctant to spend much resource to conduct such kind of highly technical analysis. On the other hand, firms adopting software applications would not be able to touch the codes and there is no need for them to manipulate the underlying technical design in the core. The essence of the challenge for them is rather the match between business requirements and software application through parameterisation and customisation.

2. Analysis of Business Processes

The depictions of business processes are vital notations to describe, examine, streamline a firm's value activities. The business requirements analysis may begin with process mapping. The typical notation in this context is the activity diagram. Although textual use-cases are also widely used for this purpose, the activity diagram is less time-consuming and more powerful in terms of ease of use and intelligibility, especially for implementation projects with stringent time demand.

Professionals dealing with Software Engineering frequently use Unified Modelling Language (UML) as a unified standard platform to map the business processes for requirements analysis purposes. However, in order to gain the necessary skill, potential analysts have to take part in formal trainings. Furthermore, commercial UML tools equipped with regular upgrades and updates are likely to be expensive. Microsoft Excel, on the contrary, is available to most firms and can also be used for process mapping. Whichever tool is used, a number of essential aspects must be taken into account to sufficiently decipher important information in business processes.

A *role* is a group of system users performing the same functions in the business processes. The description in the activity diagram needs to unambiguously define the role conducting a certain *activity (process step)*. If necessary, detailed explanations can be given to each activity, in order to map the essential content of the activity. It is highly recommended that the *system-related activities* are highlighted. By doing so, the business specialists and the analysts can subsequently defined the *data fields* of *system inputs and outputs* at a particular system process step. It is beneficial that these inputs and outputs are streamlined at a later stage to make the data structure of future application more meaningful. *System printouts* need to be indicated, analysed and carefully defined at relevant process steps. Although the activity diagram is not supposed to be a *system specification*, it is recommended that the analysts define as much detailed information as possible, in order to gain time advantages for further implementation. In a business environment, where time is always in shortage, successful process mapping with detailed information might be a possible substitute of a time-consuming specification.

Importantly, the mapped processes then need to be tested in the pre-configured software to be introduced, so that the business specialists can witness that the mapped processes can be realised by the system. This kind of test is not the highly stringent software testing in the traditional sense, but a kind of functional and performance tests at a high level. The result of the test is a brief but written *Gap Analysis*, documenting the activities, process steps or step sequences that cannot be exactly realised by the system. In most cases, the solution to bridge the disparities can be found by providing *workarounds* and *customisation*. Workarounds are techniques that utilise and combine the existing system features, in order to achieve the missed linkages to the business processes. Customisation means that programming at the application level is necessary to map the system to the processes identified, however without the necessity to touch the source codes in the core system. While implementing a new or new generation of software system, changes in the core system (*development* efforts by the coding firm) is generally not recommended, because this would imply intensive time and resource capacity for expertise exchange between the implementing and *coding* firm, unless it is unavoidable. Painful costs are foreseeable.

3. Analysis of Business Products

Not all important information is contained in the business processes. In a retail system for example, firms must certainly comply with governmental pricing regulations, which are normally readily available in the system. However, these firms may also desire to create their own product-related pricing and charging schemes based upon proprietary calculations. This kind of product-specific information typically resides in *documented product descriptions*.

The realisation of a product may involve different activities in different processes. Process and product views shed light on the business portfolio of a firm from different angles of perspective. While analysing the business products, analysts need to pay intensive attention to *features, interfaces, ancillary products*. Features are functionalities (e.g. proprietary calculations), delineating what the product should perform and how the added value is created. Interfaces include system-internal communication with other products within the portfolio and system-external communication with other systems if necessary. Today, standardised interfaces have made the industrial value chain operate more smoothly. Ancillary products are resultant aspects of an existing business portfolio. Businesses not only reply upon on activities and values, but also on the reporting, controlling and audit of these activities and values.

Similar to the process mapping, product analysis within the context of system implementation should include a *Gap Analysis* that documents the vacuum between the wished products and features provided by the software. Workarounds or customisation should be conceived, designed and carried out subsequently if necessary. Again, core changes should be held in a minimum scale.

4. Determination of Application Parameters

After having analysed the processes and products, the business specialists and system analysts need to focus on the definition of application parameters. *User rights* and categorised *parameter tables* must be discussed in great detail, before figures, number and ranges can be setup in the system to achieve desired products and processes.

Careful definition of user rights is highly salient for the security and smoothness of the business operations. The analysis of user rights for each user group must at least deal with access to system masks, level of data inputs and processing, authorisation of data processing, access to data outputs (e.g. reports), and access to system administration.

Other parameters can be categorised in individual tables as the basis of discussion. Business specialists from different product or process background must first understand what the parameters defined by the core software system mean. System specialists or external consultants familiar with the system are required. The central task here is to map the system defined parameters to the product parameters used in the businesses, in terms of both definition and terminology. These discussion sessions are highly important. One major result of the parameterisation is to enable the purchased system to perform the business content of the purchasing firm, partially through workarounds. Another major result should be that customisation and external development tasks are figured out in detail, upon which follow-up resource and budget needs can be based.

Multinational businesses are additionally faced with the difficulty of parameter differences that are necessary for different national marketplaces. Therefore, system implementation may require the definition of a *multinational parameter mix*. Because experts of local parameters are located in the respective local markets, parameterisation under such circumstances may require intensive communication with subsidiaries and representatives located in other countries. Analysts, business and software specialists in the central headquarters can use this opportunity to reside in the multinational sites for information elicitation and analysis, and by doing so, amplify the knowledge of the local business environment. It is worth mentioning that short stays in countries with lower living standard may not be highly comfortable, but the learning effect potentially achievable may be comfortably high.

Although parameter difference may exist across the national borders, unified process and product definition may be beneficial to achieve scales economies in the global strategy of the business firms.

5. Project Management

Tailoring the software system to the firm-specific business requirements is a daunting task of high complexity, consisting of hundreds or even thousands of work units and packages. Without proper management the whole project would be a monstrous amount of work without prosperity. Project management is in most cases inevitable in general. In particular, *time*, *resources* and *budgets* are to be planned and managed delicately.

Time management should include a highly detailed listing of work packages and their interdependencies. The duration of each work package is estimated carefully. Statistical methods such as beta distribution can be deployed here. Project professionals use to apply network depictions to illustrate the structure of the project and identify the critical path, upon which the complete project duration may be estimated. Resource loading diagram and resource levelling help the project management maintain an overview of deployed resources and avoid extreme over- or under-occupation. In software implementation projects, human resources must be dealt with more carefully. Holidays and travel plans must be considered. Furthermore, the costs of workarounds, parameterisation and customisation mentioned above are certainly also a part of overall costs. Although the top-down approach is the predominant budgeting policy in most implementation projects, the mangers should always have an open ear for bottom-up information coming from project members to insure a more realistic budgeting plan. It is worth mentioning that the project communication with top management and external system supplier must be honest and transparent. Business requirements analysis and software implementation are not just about software and hardware, but also about trust and relationship. These soft factors can sometimes even be decisive for the overall success of the project.

6. Conclusion: 4Ps of business requirements for successful software implementation

Parameterisation-capable and customisable software applications tailored to firm-specific business requirements have become highly coveted in myriads of industries and firms. This paper argues that 4Ps are most essential for integrating such systems seamlessly in the firm-individual operational environment:

- (P)rocess: An effective process mapping should delineate functional roles, process steps and detailed content of process steps. It should highlight the system-related activities and, data inputs and outputs at these activities. It should also define system printouts. Analysts and business specialists must analyse the process a number of times, thereby moving from less to more detailed levels. By conducting a careful GAP analysis, analysts can identify the needs for future workarounds and customisation. External development is to be avoided as much as possible.

- (P)roduct: Products must be defined in a written document, which clarifies the product features, internal and external interfaces and ancillary products. Similarly, a resultant Gap Analysis is strongly recommended. The software to be purchase must perform what the defined products need, at least through workarounds and customisation. Important product features should be realised straightaway by the system. External development is to be held in a minimum scale.
- (P)arameter: User rights, process-related and product-related parameters, parameter ranges to be applied in the system should be defined and reviewed carefully. Sometimes, it is necessary that business parameters familiar to the business specialists must be mapped to the system parameters familiar to the system specialists. The analysts should, accompany and intermediate in, this process. Multinational corporations may have to adapt the parameters to the local operational sites in a parameter mix.
- (P)roject: System implementation tailored to firm-specific business requirements consists of complex activities that can only be effectively and efficiently managed if a proper project management environment is in place. Completion time, resource allocation and budgeting are most important aspects. Project management should also take into account the activities necessary after the business requirements analysis has been carried out. Typical activities are the setup of defined parameters in the system, customisation activities, acceptance testing, go-live of the system and, screening of results and communication for further system improvement.

References

Pohl, K. (2007). *Requirements Engineering: Grundlagen, Prinzipien, Techniken*. Heidelberg: dpunkt.verlag GmbH.

Pressman, R. S. (2004). *Software engineering: A practitioner's approach*. (6th ed.). New York, NY: McGraw Hill.

Pressman, R. S. (2009). *Software engineering: A practitioner's approach*. (7th ed.). New York, NY: McGraw Hill.

Robertson, S., & Robertson, J. (2006). *Mastering the requirements process*. (2nd ed.). Westford, MA: Pearson Education Inc.

Wiegers, K. E. (2003). *Software requirements*. (2nd ed.). Redmond, Washington: Microsoft Press.