Identifying Types of Extra-Functional Requirements in the Context of Business Process Support Systems

Elke Hochmüller[1], Michael Dobrovnik[2]

[1] Carinthia Tech Institute, Department of Telematics / Network Engineering,
Primoschgasse 8, A-9020 Klagenfurt, Austria
E.Hochmueller@cti.ac.at

[2] Groiss Informatics, Strutzmannstraße 10, A-9020 Klagenfurt, Austria
michi@groiss.com

Abstract. Extra-functional requirements are known as critical success factors in traditional software engineering. While business process support systems and classical domain-specific application systems differ from the product point of view as well as from the perspectives of development and institutionalisation processes, the relevance of extra-functional requirements will even increase in case of BPS systems because of the tight relationship of such systems with the structures, responsibilities, and processes within an enterprise. This discussion paper aims at the identification of general types of extra-functional requirements which are most prominent in BPS systems.

1 Introduction

Business Process Support systems differ substantially from traditional domain-specific application systems. Requirements Engineering for conventional application systems includes tasks like information acquisition from different users with different viewpoints leading to an integrated conceptual static model and a model of the required system dynamics which have to be validated against the users' needs (elicitation, specification, validation) [1]. BPS systems require a broader perspective. Analysis for BPS systems include the analysis of the organisation structure, responsibilities, working practices, task structures, inputs and outputs of existing application systems, document traces and flows of data.

In traditional software engineering, improper consideration or treatment of extra-functional requirements caused projects to fail even when the delivered systems met all of the functional requirements. Extra-functional requirements (also known as non-functional requirements or simply –ilities) are constraints regarding quality (e.g. usability, performance, security, maintainability) and economics (e.g. time, cost) of process as well as product components. While existing approaches in software development mainly support the handling of functional requirements, extra-functional requirements tend to be simply forgotten or even neglected from the beginning. When they happen to turn up again (during acceptance test or even during operation), in

most cases it is too late to react properly [1, 2]. Because of the more complex nature of BPS systems (as discussed in section 2), it is highly advisable to learn from previous experiences and not to fall again into the "extra-functional requirements trap" which is already known from traditional software system development. As a first step, this requires to identify possible classes of extra-functional requirements which particularly will apply for BPS systems.

2 Characteristics of Business Process Support Systems

Before we can identify BPS-related types of extra-functional requirements we should try to take a closer look at the specific nature of these kinds of systems [3]. Differences in the development processes between traditional information systems and BPS systems are stated in [4]. A summary of BP characteristics and related requirements engineering issues is included in this volume [5]. The major phases in case of workflow application development processes (WADP) have been already investigated, so far [6].

Some general characteristics of BPS systems which are considered to be important from the operational point of view are the following ones:

- *Process-Orientation.* BPS systems focus on the assistance of people in their working processes. The prevailing concept is that of processes and workflows.
- *Close Tie to the Organization.* A BPS system is closely related to the enterprise in which it is installed. As the BPS system is the common process engine within the enterprise, almost each employee will use it. Real-world processes have to be mapped onto the BPS system in a straight-forward manner such that the system smoothly fits into the working place environment/infrastructure.
- *Large Scale.* BPS systems are rather complex in their nature. They have to coordinate rather complex tasks of different agents in usually heterogeneous and decentralized environments.
- *Integration Feature.* A BPS system is a core component within an enterprise. On the one hand, it serves as a coordinator between different agents within the same as well as across different departments. On the other hand, it acts as an interface between other software systems in use within the enterprise.
- *Process Control.* A BPS system does not only facilitate operational tasks, it can also be used for measuring process success which is a prerequisite for subsequent process optimization activities.
- *Process Adjustments.* Optimization of processes requires continuous changes of process definitions.

3 Extra-functional Requirements for BPS

Based on the observations of the nature of BPS systems, some types of extra-functional requirements can be identified which are most likely to be relevant for BPS

systems irrespective of the actual kind of processes to be supported. However, the degree of importance may certainly vary with the actual process domains.

- *Longevity.* Because of the tight interweaving with the organizational structures and processes, a BPS system must be able to tolerate and survive organizational changes. It must be general enough to be able to be used for many different process domains. A BPS system has to cope with changes in real-world processes, introductions of new cooperating software systems or database management systems and alterations of data (schema evolution).
- *Flexibility, Changeability, Extensibility.* The objective of durability calls for BPS systems that show a high degree of flexibility such that the systems can easily (within reasonable time and budget) be repeatedly adapted to changed situations within the organization, environment, or legislation.
- *Scalability.* As BPS systems are usually long-living systems, they have not only to be able to manage the increasing amounts of information such that processes can also be accessed and retrieved months and years after their completion (high capacity). Above all, these systems have to be able to scale up with increasing numbers of users and with higher process volumes.
- *Security and Privacy.* BPS systems help various persons with different responsibilities in fulfilling their daily work. The underlying role concept (agents and their substitutes, groups of agents with the same rights) together with appropriate access rights (authentication) have to be mapped onto the system.
- *Traceabiliy.* Process control requires the establishment of process traces (audit trails) in order to be able to localize the current state of a required piece of information (Who has already dealt with the document and what did this person do with the document? Who is the current owner of a document?). System characteristics (e.g. average duration of a process) have to be monitored, too.
- *Availability and reliability.* BPS support systems play a central role within an enterprise. They have to run in a stable environment, shall be highly available and behave in a reliable manner. An appropriate transaction management is inevitable.
- *Commercial Availability.* A BPS system should be available on many different platforms and thus it should also act as an interface between other application systems on different platforms (heterogeneous systems). Upgrades have to be guaranteed and delivered for all platforms used.
- *Reusability.* Historical and actual process-related information should be accessible in order to be reused for further process (re-)definition.
- *Prototyping or Simulation Facility.* A BPS system should support prototyping in such a way that processes can be easily defined, simulated and tested, not only within organization units but also across organization boundaries (without interfering with existing processes).
- *Usability.* Last but not least, usability is already known as a quite insidious kind of extra-functional requirement. In traditional systems development, user interfaces will be tailored to the abilities of types of stakeholders. BPS systems usually offer a common look and feel which makes it necessary to agree upon a user interface which has to be (more or less) accepted throughout the whole organization. Usually, workflow management systems do not provide the best user interface regard-

ing ergonomics. Hence, adequate education and training activities are essential to at least partially compensate usability deficiencies.

4 Discussion

What are the benefits for nailing down types of extra-functional requirements which are relevant in the case of BPS systems? The list of BPS-related types of extra-functional requirements compiled above is not a full and deep taxonomical treatment but is meant to give an representative overview of some essential aspects. It can serve as a foundation for three essential purposes:

First, such results can act as a general classification framework which can be used in a domain-independent manner to compare BPS systems. It can help to answer questions like:

- What kinds of BPS systems do exist?
- In what aspects do systems from various vendors differ?
- Are there requirements which are agreed upon in the whole industry or in standards?
- Are there areas of disagreement?

Second, when sufficiently refined and adapted to specific needs in a concrete enterprise or application situation, the identified types of extra-functional requirements can act as criteria for system selection, giving guidance to answer questions like:

- Which extra-functional requirements (system properties) are particularly important for our enterprise?
- How are these requirements met by eligible systems?

Third, the identified requirements types can act as an indicator providing guidance to pinpoint key requirement areas which have to be addressed in each business process support project. Since they are identified to be crucial in this area, one should not neglect them under any circumstances by assuming the tool would somehow magically deal with them on its own. It would be a mistake to take any of those requirements as automatically granted just because the system provides some framework which makes it rather easy to implement them. Without the careful and specific formulation of the requirements in the particular case, no designer or implementer would have the slightest idea about them and would do nothing to ensure their realization. A particular striking analogy comes into mind from the field of database management systems. In this area, not a few projects got into troubles by taking features like e.g. concurrency and transactions simply for granted and by not discovering before the deployment phase that in order to make proficient use of those mechanisms, one must understand them and take their peculiarities into account in the design and implementation phases. Pure reliance on the tool or environment to meet requirements is often equivalent to ignoring them altogether.

References

1. Loucopoulos, P., Karakostas, V.: System Requirements Engineering. McGraw-Hill, London (1995)
2. Hochmüller, E.: Towards the Proper Integration of Extra-Functional Requirements. The Australian Journal of Information Systems, Vol. 7 (1999) 98-117
3. Hollingsworth, P.: Workflow Management Coalition: The Workflow Reference Model. WfMC TC00-1003 (1995)
4. Alexander, I., Bider, I., Regev, G.: Workshop on Requirements Engineering for Business Process Support (REBPS'03), Objectives and Motivation. CAISE'03 Workshop Proceedings (2003)
5. Loucopoulos, P.: The S3 (Strategy-Service-Support) Framework for Business Process Modelling. CAISE'03 Workshop Proceedings (2003)
6. Weske, M., Goesmann, T., Holten, R., Striemer, R.: A Reference Model for Workflow Application Development Processes. In: Georgakopoulos, D., Prinz, W., Wolf, A.L. (eds.): Proceedings International Joint Conference on Work Activities Coordination and Collaboration (WACC), ACM (1999) 1-10