

## Essentiality of Changes in Business Models

Erika Asnina

Institute of Applied Computer Systems, Riga Technical University,  
Meza Street 1/4, LV-1048, Riga, Latvia  
[erika.asnina@rtu.lv](mailto:erika.asnina@rtu.lv)

**Abstract.** Although the fact of adaptation to changes itself does not guarantee the success in the future, an organization must adapt to changes as fast as possible in order to have sustainable business. This paper considers only one aspect of the adaptation process, which is related to changes in organization supporting software. Undoubtedly, the organization should assess the essentiality of software changes to its business activities, their impact on business supporting software, and the necessary speed of adaptation. Business complexity that is a problem for assessment can be handled by business models that represent different aspects of organization's work. This paper discusses the definition of change essentiality, as well as layers of a business model affected by changes, and proposes a common view of components and a way of determination of the essentiality of changes.

**Keywords:** Business model, change, essentiality of software change.

### 1 Introduction

Organizations run in the world that changes fast. Different forces, external and internal, affect their businesses. The ability to assess changes and to adapt to them as fast as necessary is one of ways for organizations to keep their business sustainable.

What does such adaptation actually mean? Large organizations have complex business processes with different business goals, visions, strategies, stakeholders, resources, artefacts and so on. In order to manage the complexity, different information systems (ISs) are used to understand and support business processes, e.g., Enterprise Resource Planning (ERP) systems, Decision Support Systems (DSS), etc. Organization business processes are partially supported by ISs. Changes in those supported parts of the real business processes should be reflected in the corresponding ISs. These changes could affect the implementation of other (related or dependent) business processes.

Usually, software developers take into account a technological aspect of change implementation in ISs, but ignore the impact of the implementation on the organization's work. In order to analyse this impact, the importance of successful implementation of changes should be evaluated. However, organization business processes are quite complex and the only way to deal with this is modelling. Therefore, it is recommended constructing business models, which reflect

organization business processes and their implementation in ISs. The analysis of the impact of software changes in business models can help in the assessment of change essentiality for the organization.

This paper considers the notion of the essentiality of a software change as well as factors or forces, which affect the organization business processes. Determination of the essentiality of software changes is necessary for making a decision about the speed of adaptation to changes in the organization, because every change requires additional resources that in other situation would be used for the elaboration of the business.

The paper is organized as follows. Section 2 discusses a notion of a model of businesses and forces, which affects both business processes and their implementation in software, correspondingly. Section 3 gives a brief summary of components, which constitute such a notion as “essentiality of a change”. Section 4 discusses the related works. The paper concludes with the obtained results.

## **2 Business, Business Models, and Forces of Changes**

Complexity of the organizations and their businesses is a well-known issue nowadays. Knowledge on this complexity is distributed among employers and employees. Senior managers see the whole picture from their eminency but may not know about specificity of some business processes in the organizations. This specificity usually is well-known for lower-level managers and workers. In turn, the technological implementation of this specificity is usually known only for organization’s IT staff. There must be a way to get, keep and share knowledge about complex systems among stakeholders.

Since ancient times, modelling is a natural way to deal with the complexity. There are different frameworks for developing an enterprise architecture and models, which could be used for the entire or partial representation of an organization and its business (its structure, processes, information and resources), e.g., frameworks such as Zachman Framework [1], TOGAF [2], and DoDAF [3], process models represented by BPMN [4] and IDEF [5], object-oriented models represented by UML [6], and so on. All the frameworks dedicated to developing enterprise architecture consider enterprise information systems and software tools as an integral part of existing business activities of the organization.

A business model as software developers understand this concept is a description of those organization’s business activities, which are implemented in support of software tools. Changes in business models are understood mostly as technological changes in software. This leads to a well-known fact that complexity and importance of changes are assessed only from one aspect, namely, their technological implementation in software.

A common thing in enterprise modelling (as it is considered in TOGAF and DoDAF) is that an enterprise can be described from multiple viewpoints in multiple models. For example, TOGAF foresees the business architecture, data architecture, application architecture, and technology architecture. In its turn, DoDAF considers constructing models from the capability viewpoint, data and information viewpoint,

operational viewpoint, project viewpoint, service viewpoint, standard viewpoint, and system viewpoint. These viewpoints and corresponding models can be located at the three layers of a business model [7]: a strategic layer (planning), a business model layer (architecture), and a process layer (implementation). The strategic layer joins vision, goals and objectives of the organization in order to gain income. The business model layer contains, in essence, data structure and money earning logic (or business logic in terms used by developers of information systems). Finally, the process layer represents implementation of the business model in business structure, business processes and infrastructure (including, information technology tools).

As mentioned above, organizations run in the changing environment. Different forces affect organization's business and cause changes in it. These forces can be divided into external (environmental) and internal ones [7]. Internal forces raise changes adapted by the organization's initiative in business strategy, organization and information and communication technologies. However, internal forces themselves are the reaction on the external forces from the surrounding environments - technological, customers', competitive, legal and social ones:

- Technological forces generate a request for a technological change. Since technology touches almost all the activities of the organization, it is the main force for changes in models of organization's business. Sometimes these changes may even destroy the existing way ("fashion") of business run. This means that technological forces cause changes in the implementation layer. And then changes at this layer affect higher layers (architecture and planning) of the business model.
- Competitive forces are also very important. The fast adaptation to changes generated by the competitive forces has a vital importance for the organization's sustainability. Competitive forces affect technological forces and change business run as well as used information and communication technologies. First, these forces affect the planning layer, in which changes may be propagated to the architecture and implementation layers.
- Customer demands are also quite important forces. They affect customer models, income growth, "fashion" changes, etc. This means that they, at first, affect and cause changes in the planning layer. Then these changes may be propagated to the architecture and implementation layers.
- Legal environment also often causes changes in the business model. The importance of fast adaptation of these changes is huge for organization's sustainability. First, these changes may affect all the three layers. For example, if a new law puts some restrictions on the business run, this may have an impact on the planning layer to the architecture layer and to the implementation layer. On the other hand, if a new law demands only indication of some additional data in business reports, this may affect only the implementation and architecture layers.
- Social environment itself has an indirect influence on the organization. It mostly affects customer demands and technological forces.

External forces are interrelated as forces from the social environment, but this moment is neglected in the current paper, since only a direct impact of a force on the business model is considered here. Exactly, only those forces that raise changes in the implementation layer, which further may affect organization's sustainability, are the focus of the current research.

According to [8], the importance of an organization's business model change is a criterion for making a decision on the speed of adaptation to changes in the organization. It could be ranged as *big*, *medium*, or *small*:

- Big importance means that an impact on the organization is crucial, and the changes have to be adapted quickly, in order to keep its sustainability in the market;
- Medium importance means that an impact on the organization is less crucial than in the previous case, but the organization has to react adequately quickly in order to adapt the changes in the environment to its business model;
- Small importance means that an impact on the organization is inessential; thus, the organization may delay the adaptation of changes. However, it is not recommended postponing those changes in order to keep organization's sustainability in the market.

### 3 Components of Essentiality of Software Changes

The notion of essentiality of software changes covers such aspects of changes in the implementation layer of the business model as their importance, scope and costs. This section discusses only importance and the scope of a software change.

Fig. 1 illustrates components of software change essentiality defined within the research. The first part corresponds to the business system dimension. It is importance of changes caused by the external forces. The second part corresponds to the software system dimension. It is a scope of changes in a model that specifies the implementation of business processes.

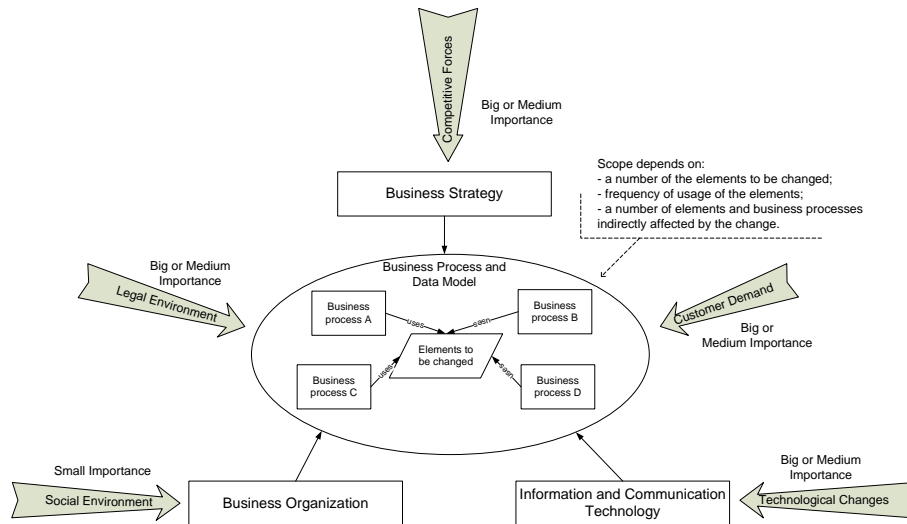
According to the impact of the external forces on the implementation model of business processes, they can be ranged as follows: changes caused by the legal environment, competitive forces, technological changes as well as customer demands have *big* or *medium importance*, and changes caused by the social environment have *small importance*.

The scope of a change is evaluated taking into account three factors:

1. a number of elements (classes, tables, procedures, user interfaces, generated reports, etc.) in software to be changed that are directly affected by the change;
2. frequency of usage of those elements in the work activities of the organization;
3. a number of elements and business processes indirectly affected by the change, i.e., those which should also be changed in order not to lose the integrity of the business model.

The scope of a change could be evaluated by using change impact analysis techniques, for example, like one represented in [9].

In a complex case, evaluation of the general importance of a software change should be defined in business rules by organization's experts [8], since each external force has its own importance and differently affect the business itself and organization's software tools; it depends on the business specificity.



**Fig. 1.** Components of essentiality of a software change.

The general essentiality of a software change is evaluated as a sum of products of the multiplication of importance of a change that is determined by its origin environment and the scope of changes in a business model of implemented processes caused by them.

Let us take as an example a network of stores that purchases and sells products and a bank. Let us assume that they work by the Latvian legislation. According to the transition from the Latvian national currency LVL to the European Union currency EUR, since October 1, 2013 till December 31, 2013 all prices in any documentation must be reflected in both currencies – LVL and EUR. It is a change from the legal environment, whose range is big or medium. In order to evaluate the essentiality of a software change, firstly let us consider the importance of this change in the business system dimension, and secondly - the scope of this change in the software system dimension.

The same expert can measure the general importance of this change for the two organizations differently. This importance is based on risk evaluation for a certain company. A segment of the market (money circulation, a number of clients, etc.) for the store and for the bank may be equal. However, bank's main resource is money. Moreover, accuracy in calculations is vitally important for the bank, because clients' confidence depends on this. Unsuccessful adaptation to this change brings smaller risks for the network of stores than for the bank. Therefore, the importance of successful adaptation to this change is bigger for the bank than for the network of stores.

If we consider the scope of a change in the business processes implemented in software, then it is clear that the new field for the price in EUR must be added to elements in databases (corresponding tables, indices, and stored procedures) and in software code (corresponding entities, such as classes, operations, modules, procedures, calculations in services, reports, receipts, and user interfaces). Changes in the databases are direct. Changes in the software code are indirect, since depend on

changes in the database elements. The scope of changes may be equal for both organizations.

The general essentiality of the currency change is bigger for the bank than for the network of stores, even in case, if the scope of the change is equal.

## **4 Related Works**

The authors in [10] declare a similar thought that the estimation of software change should not be based only on simple evaluation of an impact set. The authors suggest ArchEvol, a programming environment that allows managing concerns related to software changes. Each concern is located in a special concern model and is related to implementing elements in a code. Automated support allows developers to continuously examine, structure and modify the software in terms of concerns.

The authors in [11] use an integrated approach to the change impact analysis that is based on three techniques, namely, information retrieval, dynamic analysis, and data mining of past source code. This allows finding the context of a change that can be missed during usual software change impact analysis. However, the scope of this context is only software elements and their dependencies and does not take into account change origins.

The authors in [12] highlight the importance of keeping knowledge about the origin of software architecture design. They provide a conceptual framework that is implemented as a set of architectural reasoning diagrams, where each diagram is composed of several layers and addresses certain aspects of the system. The layers are the following: the stakeholders' layer, the requirement layer, the analysis and decision layer, and the architectural layer.

The authors in [13] suggest a new tool Panaya Impact Analysis that enables the automatic identification of customization change impact on ERP system behaviour. In order to give the user a clear image of an impact set, it uses a slicing mechanism, several techniques (grouping, similarity, and filters), and three basic static analysis algorithms for the customization change impact analysis. The tool does not change the program, but produces only analytical reports for ERP professionals.

The authors in [14] analyse changes in databases and their impact on data and code of the related software. They suggest identifying the atomic changes in the database and conducting the co-change analysis for predicting an impact set of code lines.

## **5 Conclusion**

The evaluation of essentiality of software changes is important to organizations in order to adequately quickly adapt to the changes and to retain sustainable in the market.

The essentiality of software changes can be evaluated by analysing affected parts of a business model of implemented processes of the organization. The range of general importance of a change is big, medium, and small. The first two require operative actions, while the last one could be postponed. The essentiality is evaluated

as a sum of products of the multiplication of importance of a change that is determined by its origin environment and the scope of changes in a business model of implemented processes caused by them.

Certainly, evaluation rules must include measurements of change importance. These measurements may be provided only by experts in a certain business domain, since they reflect specificity of the business in the concrete economic circumstances.

Thus, the assessment of the essentiality of a software change would take into account factors, which are important for developers of supporting information systems (the scope of changes), and factors, which are important for business sustainability in the market (the importance of changes). It would allow organizations to make an adequate decision on the speed of adaptation to the changes.

The proposed idea can be verified by changes in the same software system (e.g., an ERP system) that is used by several organizations with different and the same business activities.

## Acknowledgments

The research has been conducted within the framework of project No. 1.10 “Competence Centre for Information and Communication Technologies”.

## References

1. John Zachman's Concise Definition of the The Zachman Framework by Zachman International, 2008, <http://zachman.com/about-the-zachman-framework>
2. The Open Group Architecture Framework (ToGAF), <http://www.opengroup.org/togaf/>
3. DoD Architecture Framework, <http://dodcio.defense.gov/dodaf20.aspx>
4. OMG Business Process Model and Notation, <http://www.bpmn.org/>
5. Integrated DEFinition Methods, Knowledge Based Systems Inc., <http://www.idef.com/>
6. OMG Unified Modeling Language, <http://www.uml.org/>
7. Osterwalder, A.: The Business Model Ontology: A Proposition In A Design Science Approach. PhD. Thesis. Universite De Lausanne, Ecole Des Hautes Etudes Commerciales (2004)
8. Jakimavicius, T., Kataria, P., Juric, R.: Semantic support for dynamic changes in enterprise business models. *Journal of Integrated Design & Process Science*. Vol. 14, Issue 2, 1--11 (2010)
9. Soffer, Pn.: Scope analysis: identifying the impact of changes in business process models. *Software Process: Improvement & Practice*. Vol. 10, Issue 4, 393--402 (2005) <http://dx.doi.org/10.1002/spip.242>
10. Nistor, E.C. and van der Hoek, A.: Explicit Concern-Driven Development with ArchEvol. In *Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering (ASE '09)*. IEEE Computer Society, Washington, DC, USA, 185-196 (2009). <http://dx.doi.org/10.1109/ASE.2009.70>
11. Gethers, M., Dit, B., Kagdi, H. and Poshyvanyk, D.: Integrated impact analysis for managing software changes. In *Proceedings of the 2012 International Conference on Software Engineering (ICSE 2012)*. IEEE Press, Piscataway, NJ, USA, 430-440 (2012).
12. Orlic, B., Mak, R., David, I., and Lukkien, J.: Concepts and diagram elements for architectural knowledge management. In *Proceedings of the 5th European Conference on*

Software Architecture: Companion Volume (ECSA '11). ACM, New York, NY, USA, , Article 3 , 10 pages (2011).

13. Dor,N., Lev-Ami, T., Litvak, S., Sagiv, M. and Weiss, D.: Customization change impact analysis for ERP professionals via program slicing. In Proceedings of the 2008 international symposium on Software testing and analysis (ISSTA '08). ACM, New York, NY, USA, 97-108 (2008).
14. Qiu,D., Li,B., and Su, Z.: An empirical analysis of the co-evolution of schema and code in database applications. In Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2013). ACM, New York, NY, USA, 125-135 (2013).