Lecture Notes
in Business Information Processing 148

Series Editors

Xavier Franch
Pnina Soffer (Eds.)

Advanced Information Systems Engineering Workshops

CAiSE 2013 International Workshops
Valencia, Spain, June 17-21, 2013
Proceedings

Springer

Volume Editors

Xavier Franch
Universitat Politècnica de Catalunya
Service and Information System Engineering
Barcelona, Spain
E-mail: franch@essi.upc.edu

Pnina Soffer
University of Haifa
Information Systems
Haifa, Israel
E-mail: spnina@is.haifa.ac.il

ISSN 1865-1348 e-ISSN 1865-1356
ISBN 978-3-642-38489-9 e-ISBN 978-3-642-38490-5
DOI 10.1007/978-3-642-38490-5
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013938414

ACM Computing Classification (1998): H.4, H.3.5, D.2, J.1, I.2

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Towards Understanding Software Process Variability from Contextual Evidence of Change

Tomás Martínez-Ruiz, Francisco Ruiz, and Mario Piattini

Alarcos Research Group, Information Technologies and Systems Institute,
University of Castilla-La Mancha
Paseo de la Universidad, 4, 13071 Ciudad Real, Spain
tomas.mrtnez@gmail.com, {francisco.ruizg,mario.piattini}@uclm.es

Abstract. Software development enterprises need to tailor their own processes before enact them in order to ensure that they fit both the organization and the project. This necessity has, to date, been solved by providing these processes with variability support. Tailoring proposals have traditionally been focused on solving the problem of managing the variability of processes in order to facilitate their adaptation. Process tailoring has not, however, been considered as a solution to a wider problem consisting of the organization, project, laws and some other influencing factors that change according to each project, a problem that software processes must confront if they are to be successful. In this paper we enhance a tailoring framework in order to tackle changes in the context level of the process, and this variability is considered to drive the tailoring of the supporting processes. As a part of the enhanced framework, this paper analyzes the OMG's Business Motivation Model (BMM) in order to apply it to the characterization of the organizational units as a part of the context variation factors, and to link them with subsequent process variations. The proposal is illustrated by means of an application example, which is based on a real industrial case and which has served as a proof of concept. The resulting conclusion is that since software process tailoring depends on the process context, so understanding and managing changes in the latter's drive variability in software processes.

Keywords: context change, evidence in software process, Variant-Rich Processes, tailoring management, process institutionalization, project management.

1 Introduction

The importance of process tailoring is no longer called in question. Software development organizations know that if they wish their projects to be successful, then they must use processes that meet the reality in which they are involved [1]. Tailoring is currently addressed by means of process variability techniques, which are in most cases applied from software products to software processes [2, 3]. Literature provides proof of this, and therefore includes several approaches, such as those proposed by Simidchieva et al [4] or Martínez-Ruiz et al [5].

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 417–431, 2013.

Tailoring has traditionally been treated as a problem that can be solved by means of variability support and other approaches, such as method engineering [6]. Following this line, in previous works we have presented both the Variant Rich Process paradigm [7] and the SPRINTT environment [8]. These are focused on providing process variability support, which is used in process institutionalization (embedding processes inside the organization through its tailoring from the self set of standard processes). They have been tested by using case studies and experiments, and although the results are positive, variability has been focused on the processes themselves, and the reflection of the similarities and differences to the organizations and projects in which these processes are systematically put into operation has been not considered.

In this paper, the process tailoring problem is placed in the context of a wider and more profound problem: *organizations, work teams, projects and even laws,* which constitute the *software process enactment context,* change. Since these have an impact on the software development, the processes must be tailored in order to engrain them and to include these changes. From an overall organizational viewpoint, alignment and traceability between the elements of which the process contexts and the processes themselves are composed must therefore be clearly determined, as must how the evidence of changes in the former affect the latter.

The proposed approach aims to embed the process variability and tailoring mechanisms, even the institutionalization ones, into a large environment, which also has the capability of meshing these mechanisms with the context changes. The proposal, called SPICCE (Software Process Institutionalization based on Context Change Evidence), considers that process variations may and must be developed from variations in the context, which clearly ensures their alignment and traceability with organizational strategies and project necessities. It is consequently promoted in order to apply the tailoring philosophy, i.e., to avoid creating a new process for each enactment to the process tailoring itself, by avoiding the need to adapt entire processes from scratch, and is instead driven by the changes in the context, which determine how to modify the adaptation itself from a process to another, to take into account the specific context requirements of the new one.

The research presented herein characterizes the distinct elements that a context is composed of: an organization's features, its motivation and rules, the projects and the external laws. As first step, it focuses on analyze the organization features, and proposes the elicitation of the changes in business by using feature models, which have been aligned with the OMG's Business Motivation Model (BMM) [9]. It also shows how to trace these high abstraction changes in concrete variations, which we propose to automate in future works. It opens the door to homogenize BMM with other business norms and standards, as well as the regulations that influence process enactment, and which change in between these processes.

In addition to this introduction, this article includes an overview of the state-of-the-art in Section 2, which reviews the process tailoring initiatives found in literature, along with a description of the elements included at organizational levels. The third Section presents the SPICCE proposal and the elements of which it is composed, and also includes an analysis of how BMM could be enhanced in order to make the changes in software organizations explicit. Finally, our conclusions and future work are described in Section 4.

2 State of the Art

Literature includes several initiatives concerning process tailoring, which are in most cases supported by means of modifications in process structures [10, 11]. Rombach then developed the "process lines" approach [2], while Sutton developed the links between aspects and processes [3]. The systematic literature review Martínez Ruiz *et al.* [12] presented describes how processes are modified in order for them to fit the project's needs; it also states the set of requirements a process variability notation must include to support tailoring as real organizations actually need it. It therefore guides the definition of new process variability support mechanisms.

Some other new work has also appeared since the aforementioned systematic review. Simidchieva *et al.* [4] present an explicit differentiation between problem and solution spaces, and identify three types of approaches: generation, navigation and reasoning. Araujo *et al.* [13] propose the management of process variability by identifying the common and different features of a process model. They propose a tool with which to apply MDE transformations to process tailoring [13]. Hurtado Alegría *et al.* [14] also propose tailoring software processes by using MDE and ATL transformations. Simmonds *et al.* [15] propose the creation of Basic Feature Models to represent features of the tailored process in the vSPEM notation, which constitutes our previous contribution (see section 3.1), paying special attention to orthogonal variations. These "new" approaches do not meet at all the requirements that industries require, as has been pointed out in the aforementioned SLR [12].

Of those initiatives which apply variability to software processes in order to align the processes themselves with the projects found, we should highlight the work of Martins and Silva [16, 17], a proposal based on three fundamental steps: i) defining the process, ii) adapting and monitoring the process execution, and iii) measuring the process. Killisperger *et al.* [18] suggest an environment with which to automatically apply variations to processes through the use of variation operations. Silva Barreto *et al.* [19, 20] propose another environment in which to carry out variations in software processes, with the aim of facilitating process reuse, based on the definition of variations in process components. These proposals include some aspects that seek process institutionalization, but none of them really align it with process tailoring.

With regard to tailoring and variability in the description of the organization, there are several works concerning variability in business processes, which also apply *Product Lines* or *Aspects*. Some of these works are those of Lu and Sadiq [21-23], or the AO4BPEL project of *Mezini et al.* [24-26], and they are even considered as families [27]. However, these works are not focused on discovering how the variations in the business processes affect their organization's software processes.

On the contrary, there are a few works about how the context influences business processes, but they show it is a relevant topic in the case of business processes. Ploesser *et al.* [28] presents the necessity of tailoring business processes in accordance with the context and identifies the techniques addressing it. Rosemann *et al*. [29] propose an onion layer system in which classify environmental, external, internal, and immediate influencers. In case of Product Lines, relationships between the context and the technical support have been addressed by using feature models to represent changes even in the context, as Hartman *et al.* proposes [30].

Bearing all of the above in mind, in addition to our goal of linking software process variations with the real facts motivating them in the process context, it is necessary to enhance the previously existing SPRINTT approach through the use of context change evidence management.

3 Process Institutionalization Based on Context Change Evidence

3.1 Previous Works: SPRINTT and VRP

The *Software Process Institutionalization based on Tailoring and sTandardization* (SPRINTT) approach is composed of two elements, which have been termed as the *Institutionalization Cycle* and the *Variant-Rich Process paradigm* (VRP) [7, 31, 32]. The former is the theoretical setting used by organizations to transform and include processes as new and effective assets. It defines four cyclical steps with which to *tailor*, *execute*, *analyze* and *standardize* the processes, as **Fig. 1** shows.

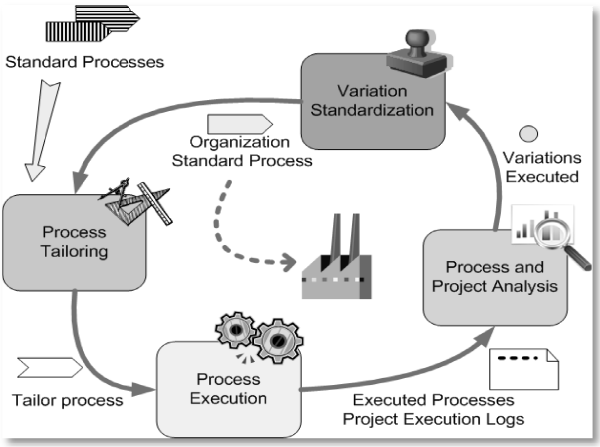

Fig. 1. Institutionalization Cycle overview

The Variant-Rich Process paradigm offers variability support which is suitable for the execution of both the first and last steps of the cycle. The paradigm provides on-point and crosscutting variability mechanisms, in addition to Rationale to support variation decisions. These are based, respectively, on Product Lines (SPLE's) [33], Aspects (AOSE) [34] and Rationale Management [35] from Software Engineering.

The VRP includes generic variability mechanisms. These have been implemented in SPEM [36], resulting in the vSPEM language [5, 7, 37-40], which provides tailoring support, as industries require. The use of the vSPEM notation makes it possible to manage on-point and crosscutting variations in software processes.

The VRP paradigm has been tested using the vSPEM notation. Experiments have shown that it is much easier to adapt processes if they are modeled using the vSPEM language [7]. These results have also clearly demonstrated that the VRP and the

notation are useful in modeling variability in Aerospatiale, Quality Assurance, and Global Software Development domains [5, 31, 38].

This framework is focused on solving the problems that the industrial and scientific communities were worried about several years ago, when our research into this area began, as these communities were totally focused on managing variability in software processes as a reflection of the product's variability support in product lines.

3.2 Enhancing SPRINTT through Seeking Evidence: SPICCE

SPRINTT was designed and built, to provide a solution to process tailoring. The VRP takes a set of tailoring requirements and, by using rationale mechanisms, finds which is the best variation supporting each requirement, which is implemented by the variability mechanisms. **Fig. 2** presents an overview of this tailoring approach.

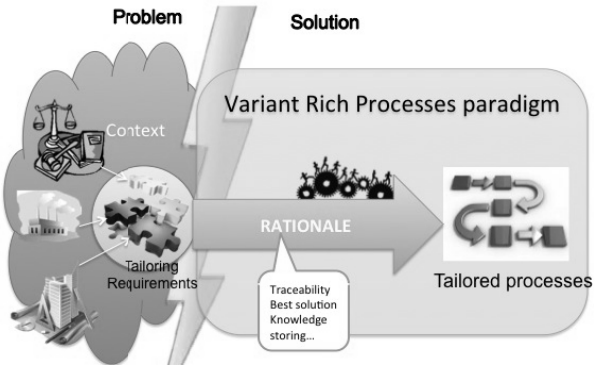

Fig. 2. Tailoring following the Variant Rich Process Paradigm

As a consequence of considering the feedback of the previous validations, we now know that a set of requirements is not a randomized creation. They are generated as a conjunction of three main elements surrounding the process enactment: the *project* (and its characteristics), the *organization* (although the entity focused on in software development is sometimes not the organization, but a department), and the *laws or regulations* (affecting the organization), as left-hand side of Fig. 2 shows. These three elements are denominated as the *process context*, and have also been identified as *project management influencers*, according to the ISO 21500 [41] international standard.

As Fig. 2 shows, the context is therefore set out in the tailoring approach. Only the requirements are used to build the solution, and software processes have additionally been considered as the only "live" entities, with the capability of changing and evolving. Additionally, from the VRP viewpoint, the elements surrounding and affecting process enactment are considered as being fixed, and if they change, SPRINTT cannot manage them.

The first solution to make the context an active part of the variant rich process paradigm would be to simply widen it on the left-hand side (Fig. 2). Tailoring requirements are defined from the context, but it is not part of them. They are

composed of the changes that these contexts' elements undergo[1]. The context is broken down into elements that categorize all the influencers of software processes, but it is open to include whatever ones its put into practice elicits.

- Projects are the most commonly-changed elements. However, they are not and cannot be considered as isolated entities. Organizations must follow some kind of working lines and their projects may therefore have a lot of similarities.
- Software organizations do not usually change, or at least not in case of traditional software development. However, Global Software Development projects, which involve different organizations, will differ from each other. But since they are all involved in the same project, it may be assumed that they will have some similarities, and that the changes or differences between them may be delimited.

  The last point regarding organizations is simply that of their scope. The organizational unit developing software is sometimes not the software organization itself, but one of its specific departments, or the union of several departments. They are therefore termed as "organizational units".
- Laws do not change frequently, and when they evolve there are still some similarities between them. In the case of global projects, affected by different countries' regulations, similarities between the laws also appear.

Tailoring requirements are defined from these changes, and they remain in process variations. Moreover, the latter are used to manage processes according to their similarities and differences and, what is most important, to take advantage of them. Herein, it would appear to be feasible to manage contexts through their similarities and differences, and considering *contexts* as *change-rich* (which is similar to Variant Rich Processes), by means of the changes between them.

Contexts' elements (project, organizational unit, and laws) are still part of the problem (processes must fit them). But now the solution is built after these changes have taken place, or even better after the more profound change evidence that the management of context offers. Mechanisms are therefore needed to control changes in the context, and to make them evident, the sooner (more abstracted) the better.

This updated perspective signifies that SPRINTT must be embedded in a wider framework that can manage changes in contexts and drive the process variations through the use of the Variant Rich Process paradigm. The *Software Process Institutionalization based on Context Change Evidence* (*SPICCE*) has therefore been created. *SPICCE seeks the promotion of process tailoring on the basis of the changes (and the evidence of these changes) that appear around the process context (project, organizational units and laws), rather than according to these elements as isolated entities*. Fig. 3 presents an overview of SPICCE, and the elements of which it is composed.

The contribution of SPICCE is to propose the use of tailoring mechanisms not by means of considering the actual organization, project or laws (crossed arrows in Fig. 3), but by considering how these elements have changed from the last time that

[1] As regards this research, changes are considered to occur in contexts and variations in processes.

the process was tailored. Changes in the context therefore make the changes that the tailoring would need evident, and thus how to obtain the new tailored process from the previous one.

SPICCE must be supported by a set of techniques, practices and knowledge to manage context and its change and evolution. The challenges of SPICCE are now therefore driven in three different ways.

- First, it needs to model the context.
- Second, it is necessary to determine how evidence of change appears in the context elements. This continues to be the principal challenge, because they are usually implicit.
- Third, evidence of change must be engrained with the rationale management (included in the VRP), with the aim of fully defining the context change and tracing it with process variations.

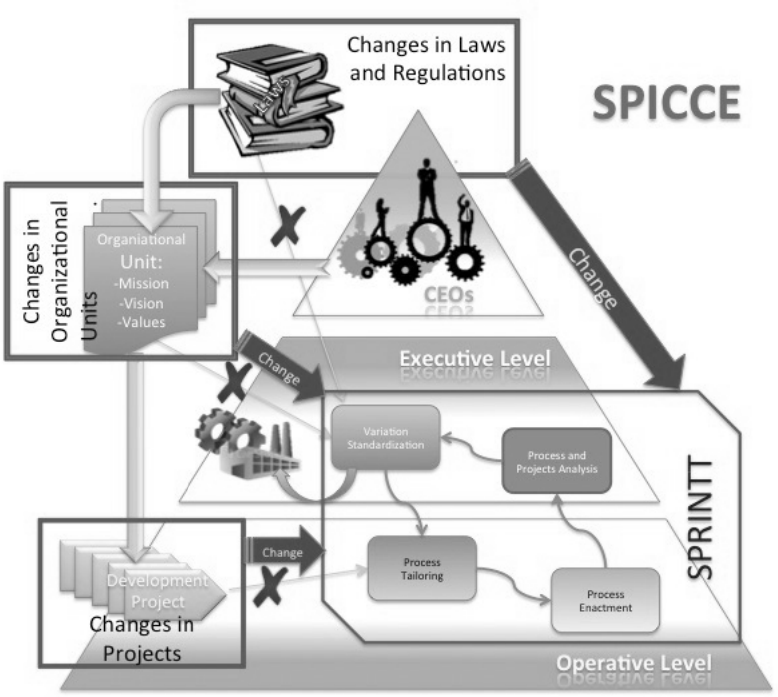

Fig. 3. Software Process Institutionalization based on Context Change Evidence

The following subsections will provide a detailed analysis of how to achieve the three points mentioned above.

3.3 SPICCE Context Modeling Steps

SPICCE needs to model organizational units, laws and projects, and to relate all of them to process variations. Fig. 4 presents a highly abstracted view of the elements of

which SPICCE is composed and the standards that will be used to support them. The parts shown in white have already been completed, while those shown in colors are under development or are planned as future work.

The previously developed SPRINTT framework is the core part of SPICCE, and supports process variations, as is described in Section 3.1. The enhancements in SPICCE are divided into three fields. Their results will be engrained with each other in order to configure an ontology [42] modeling the *evidence of context change*.

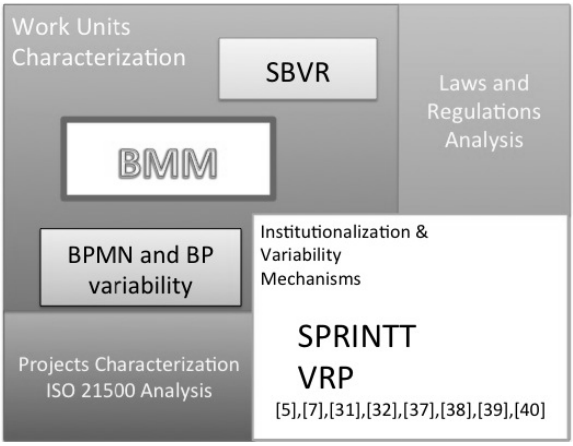

Fig. 4. Work to support SPICCE

- Analysis of Laws and Regulations. Some laws affect software processes. The most common are those concerning Intellectual property, working times, and even some syndical agreements that influence the working day. Changes to laws are, for example, more evident in the case of Global Software Development, but are always in existence.
  Other specific laws are related to desired software functions or characteristics that are under the effects of the project definition.
- Characterization of Projects. Projects are described in various guidelines or regulations, such as the aforementioned Project Management Body of Knowledge (PMBOK) [43], or even the recent ISO standard Guidance on Project Management [41].
- Characterization of the Organizational units. Work units do not have a predetermined size, but are entities with the same objectives. The Object Management Group (OMG) has recently developed several standards, such as the Business Motivation Model (BMM) [9], the Semantics of Business Vocabulary and Business Rules (SBVR) [44], the Business Process Model and Notation (BPMN) [45] or the Organization Structure Metamodel [46], all of which are focused on clearly specifying structure, business rules, strategies, plans, and even business processes (apart from the software processes themselves). All of these standards are commonly used to describe organizations, and therefore affect processes. The BMM has been considered to characterize context, as is described in the following section.

3.4 SPICCE: Eliciting and Representing Changes in Contexts

Changes in process context are not focused on seeking an implementation for them, as occurred with variations. Moreover, context requirements are written in natural language which is informal. More formal representations must therefore be provided to represent context and its changes. In this research, the BMM metamodel and feature models (and FODA –Feature Oriented Domain Analysis [47]) have been considered to represent contexts, break them down and make their changes evident. Feature models present the organization the processes must fit, and the changes that force process tailoring. The following subsections illustrate the approach by considering an application example based on an industrial experience.

Eliciting Changes in the Organizational Units. Organizational units inherit all their characteristics from organizations, and are characterized as them. The Business Motivation Model (BMM) [9] is an OMG proposal whose objective is to structure the development, communicating and managing of business plans in an organized manner. It identifies the elements of business plans, the factors affecting them, and how these are related to each other.

The structure of BMM is depicted in Fig. 5. These sets of elements allow the *means* and *end* to be highlighted, since they describe what the organization is like, or at least, what they wish it to be like; the other important parts are *influencers* and *assessment*. The latter mainly describes how different factors, from inside or outside the organization, affect it. This is particularly important in the case of external influencers which may sometimes be laws. These will be part of a more in-depth analysis of laws and regulations. Assessment depends on each organization, but has no direct effects on the processes or their tailoring.

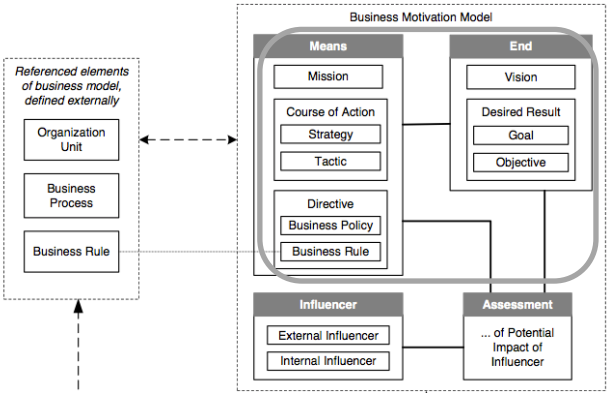

Fig. 5. Structure of BMM

Ends specify how the organizational unit wishes to be at a higher abstraction level, while *Means* include the ways in which to achieve this. Both define the organizational units, and evidence of change may appear in this description. They are decomposed

into several children, as Figs. 6 and 7 present. Finally, some relationships have been defined between the ends and means, and their break down structures. As an example, the *mission* implements the *vision*, while *objectives* are related to *practices*. These relationships make it possible to relate abstract elements (in the *ends*) to the concrete practices that support them.

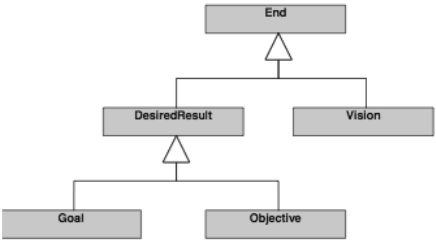

Fig. 6. Hierarchy of *ends* [9]

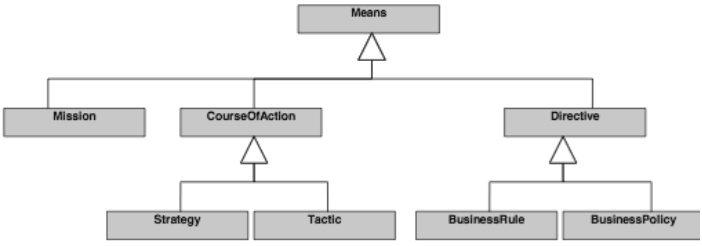

Fig. 7. Hierarchy of *means* [9]

The application study is presented to illustrate how the organization's motivation and its changes affect the process tailoring. This arose after analyzing the results of a previous case study which was conducted to validate VRP in industry.

The application study was executed in the Spanish Alpha software organization[2]. this organization is focused on providing various services, software development, software factories and security, to different clients, banks, public administration, telecommunications, commerce etc. They are experts in several architectures, such as SOA, Client-Server, embedded, business intelligence and so on. Their work is divided into three different departments, *A, B, C*. Each of them has its own version of the same process, as a base process.

They asked us to fuse all their development processes into a single variant-rich process, which would have the ability to be tailored according to different circumstances, and this resulted in the creation of the *Alpha* variant rich process. It also included three crosscutting variations. Each of these tailors the common process according to each of the three departments in the organization:

[2] Real name omitted for reasons of confidentiality.

- Highly complex
- Medium complex
- Medium simple

Some other on-point (specific) variations were also identified.

The variations obtained after executing the case study have been aligned with the organization's motivation, according to the BMM (Table 1). The results shown have been simplified in order to provide a more clear illustration of the proposed approach.

The results in Table 1 were used to align the strategy (in bold type) with the kind of developed projects, and resulted in the kind of tailoring that appears in the *Alpha* variant rich process. The strategy was modeled by using feature models (Fig. 8).

Table 1. Motivation of the organization

	Element	Value
Ends	Vision	To be the leading Spanish software development organization in their market segment.
	Goal	To provide high quality products and services
	Objective	To improve people's qualifications
Means	Mission	To provide software services in Spain
	Strategy	**Efficient and Efficacious management of software development in the** • **Highly complex projects (dep. A)** • **Medium complex projects (dep. B)** • **Medium simple projects (dep. C)**
	Tactic	Contracting highly qualified people who will, by the end of the year, be given a bonus.

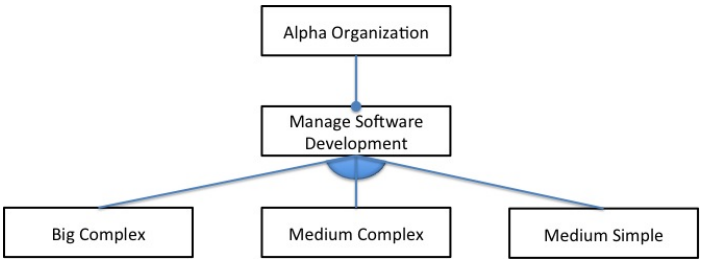

Fig. 8. Feature model of the strategy

It is now easier to see that the organization's motivation is divided into three different sub-points, which are related to the variations needed when tailoring a process.

Tailoring is now simplified to determine how the context has changed, in order to make the correct variations. Fig. 9 shows a graphic depiction of these correspondences. The slashed arrows signify the correspondence between each context change and the variations. They generate the final tailored process.

In order to create a tailored process that fits a "medium complex" project, it is only necessary to choose that change in the context, and the variations supporting it are set automatically. The process is then tailored automatically.

3.5 Engraining Change Evidence and Rationale

Fig. 9 shows a simplified situation in which there is only one change in the context to be taken into account. A real context will imply far more changes, which will occur together. Obtaining the set of variations that best fits each set of changes therefore implies making decisions about them, which is supported by Rationale.

The difference that SPICCE makes to process tailoring is that, instead of analyzing all the tailoring requirements that a context generates, it is only necessary to analyze its context changes. That is, it is only necessary to tailor what has changed.

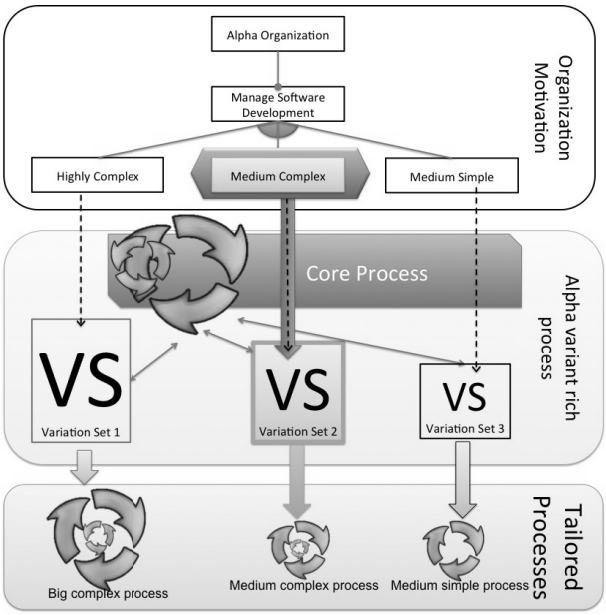

Fig. 9. Tailoring from context change

4 Conclusions and Future Work

Problems must be confronted and solved from the beginning. Existing tailoring mechanisms have attempted to do this, but since the related context causes which can drive variability are omitted, they cannot offer the most suitable solution.

Tailoring signifies that all the characteristics of a process context are included in the process, and tailoring processes according to contexts that are almost the same therefore result in almost the same processes. It would therefore appear to be better to extract the most from the similarities and differences of the contexts, and reflect them in the processes. Although this philosophy implies widening the frontiers of process tailoring to fields that have traditionally been outside this scope, this could provide process tailoring with several advantages. First, tailored processes are strongly linked to their context, and changes in the latter could immediately be traced to the evolution of the processes. Additionally, changes allow process variations to be detected much

earlier than when they are needed. These *earlier variations* provide both the variations requirements and the variability support needed.

This article has presented a proposal that is focused on integrating process variations with the actual changes in the context that motivate them. SPICCE provides a set of techniques with which to analyze and decompose each context into projects, organizations – denominated as organizational units-, and laws, and manage them according to their similarities and differences. Bearing in mind that these must later be represented or included in the (tailored) processes, making context changes evident and extending them to the process will deal with the process variations that configure the tailored process.

Implementing SPICCE involves knowing how the context is modeled, how it includes changes, and defining how to turn them into actual variations. Moreover, since this task is difficult owing to the high number of standards that are currently used to describe projects, organizational units and laws, the first step has been focused on analyzing the Business Motivation Model. The study has illustrated the potential usefulness of the proposal with an application example based on a real industrial case. It highlights that context changes that are barely evident sooner or later become process variations, and shows how it is possible to tailor the process from the context by managing these changes.

Our future work lies in complementary directions. First, it would be interesting to execute some extra case studies with which to validate the existing SPRINTT variability mechanisms in different domains, which might additionally provide feedback as regards *changes evidence*. The second direction is focused on completing SPICCE with the analysis of the standards and international regulations, as stated previously. We also plan to homogenize different regulations, such as the BMM and ISO 21500, as they are different viewpoints that represent the same context, and after that, to create an ontology to engrain projects, organizations and laws.

Acknowledgements. This work has been developed under the GEODAS-BC project (Min. de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER, TIN2012-37493-C03-01).

References

1. Zahran, S.: Software Process Improvement: Practical Guidelines for Business Success. Addison-Wesley, Harlow (1998)
2. Rombach, H.D.: Integrated Software Process and Product Lines. In: Li, M., Boehm, B., Osterweil, L.J. (eds.) SPW 2005. LNCS, vol. 3840, pp. 83–90. Springer, Heidelberg (2006)
3. Sutton Jr., S.M.: Aspect-Oriented Software Development and Software Process. In: Li, M., Boehm, B., Osterweil, L.J. (eds.) SPW 2005. LNCS, vol. 3840, pp. 177–191. Springer, Heidelberg (2006)
4. Simidchieva, B.I., Osterweil, L.: Categorizing and Modeling Variation in Families of Systems: a Position Paper. In: Gorton, I., Cuesta, C.E., Ali Babar, M. (eds.) Proc. of the 4th European Conf. on Software Architecture: Companion, pp. 316–323. ACM, Copenhagen (2012)
5. Martínez-Ruiz, T., García, F., Piattini, M.: Supporting Software Process Variability: The Variant Rich Process Modelling Paradigm Sent to JSEP (2012)
6. Brinkkemper, S.: Method Engineering: Engineering of Information Systems Development Method and Tools. Information and Software Technology 38(4), 275–280 (1996)

7. Martínez-Ruiz, T., García, F., Piattini, M., Münch, J.: Modeling Software Process Variability: An Empirical Study. IET Software 5(2), 172–187 (2011)
8. Martínez-Ruiz, T., García, F., Piattini, M.: Process Institutionalization Using Software Process Lines. In: Cordeiro, J., Fillipe, J. (eds.) 11th Int. Conf. on Enterprise Information Systems, ISAS, Milan, pp. 359–362 (2009)
9. OMG: Business Motivation Model. Object Management Group (2012)
10. Sutton, S., Osterweil, L.J.: PDP: Programming a Programmable Design Process. In: Int. Workshop on Software Specification and Desing, pp. 186–190. Schloss Velen (1996)
11. Sutton, S.M., Osterweil, L.J.: Product Families and Process Families. In: 10th International Software Process Workshop (ISPW 2006). IEEE CS (1996)
12. Martínez-Ruiz, T., Münch, J., García, F., Piattini, M.: Requirements and Constructors for Modeling Variability in Software Processes, a Systematic Review. Software Quality Journal 20(1), 229–260 (2012)
13. Araujo, F., Freire, M.A., Câmara dos Santos, W., Kulesza, U.: An Approach to Manage and Customize Variability in Software Processes. In: 24th Brazilian Symposium on Software Engineeering, SBES 2010, pp. 118–127. IEEE Digital Library, Salvador (2010)
14. Hurtado Alegría, J.A., Bastarrica, M.C., Quispe, A., Ochoa, S.F.: An MDE Approach to Software Process Tailoring. In: Proceedings of the 2011 International Conference on Software and Systems Process, pp. 43–52. ACM, Waikiki (2011)
15. Simmonds, J., Bastarrica, M.C.: Modeling Variability in Software Process Lines. Departamento de Ciencias de la Computación. Universidad de Chile (2011)
16. Martins, P.V., Silva, A.R.: ProPAM: Discussion for a New SPI Approach. Software Quality Professional 11(2), 4–17 (2009)
17. Martins, P.V., Silva, A.R.: ProPAM. SPI based on Process and Project Alignment. In: Khosrow-Pour, M. (ed.) Managing Worldwide Operations and Communications with Information Technology. IDEA Group Inc., Vancouver (2007)
18. Killisperger, P., Stumptner, M., Peters, G., Grossmann, G., Stückl, T.: Meta Model Based Architecture for Software Process Instantiation. In: Wang, Q., Garousi, V., Madachy, R., Pfahl, D. (eds.) ICSP 2009. LNCS, vol. 5543, pp. 63–74. Springer, Heidelberg (2009)
19. Silva Barreto, A., Murta, L., Rocha, A.R.: Software Process Definition: a Reuse-Based Approach. In: XXXIV Conf. Latinoam. de Informática, pp. 409–418 (2008)
20. Silva Barreto, A., Murta, L., Rocha, A.R.: Software Process Definition: a Reuse-Based Approach. Journal of Universal Computer Science 17(13), 1765–1799 (2011)
21. Lu, R., Sadiq, S.K.: Managing Process Variants as an Information Resource. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 426–431. Springer, Heidelberg (2006)
22. Lu, R., Sadiq, S.: A Reference Architecture for Managing Business Process Variants. In: 9th ICEIS 2007 (2007)
23. Liu, J.Y.-C., Chen, V.J., Chan, C.-L., Lie, T.: The impact of software process standardization on software flexilblity and project management performance: Control theory perspective. Information and Software Technology 50, 889–896 (2008)
24. Mezini, M., Ostermann, K.: Variability Management with Feature-Oriented Programming and Aspects. ACM SIGSOFT Software Engineering Notes 29(6), 127–136 (2004)
25. Charfi, A., Schmeling, B., Mezini, M.: Transactional BPEL Processes with AO4BPEL Aspects. In: 5th European Conf. on Web Services, ECOWS 2007, pp. 149–158. IEEE CS, Halle, Saale (2007)
26. Charfi, A., Mezini, M.: AO4BPEL: An Aspect-oriented Extension to BPEL. Journal of World Wide Web 10(3), 309–344 (2007)
27. Gröner, G., Boskovic, M., Parreiras, F.S., Gasevic, D.: Modeling and validation of business process families, Information Systems. Information Systems (2012)

28. Ploesser, K., Peleg, M., Soffer, P., Rosemann, M., Recker, J.: Learning from Context to Improve Business Processes. BP Trends (2009)
29. Rosemann, M., Recker, J., Flender, C.: Contextualization of Business Processes. Int. J. Business Process Integration and Management 1, 111–127 (2007)
30. Hartmann, H., Trew, T.: Using Feature Diagrams with Context Variability to Model Multiple Product Lines for Software Supply Chains. In: 12th International Software Product Line Conference, pp. 12–21 (2008)
31. Martínez-Ruiz, T., García, F., Piattini, M., de Lucas-Consuegra, F.: Process Variability Management in Global Software Development. A case study. In: ICSSP 2013. Springer, San Francisco (accepted, 2013)
32. Martínez-Ruiz, T., García, F., Piattini, M.: Framework to the Standardization and Institutionalization of Software Processes (2008)
33. Clements, P., Northrop, L.: Software Product Lines. Practices and Patterns. Addison-Wesley, Boston (2002)
34. Filman, R.E., Elrad, T., Clarke, S., Aksit, M.: Aspect-Oriented Software Development. Addison-Wesley, Boston (2004)
35. Dutoit, A.H., McCall, R., Mistrík, I., Paech, B.: Rationale Management in Software Engineering: Concepts and Techniques. In: Dutoit, A.H., McCall, R., Mistrík, I., Paech, B. (eds.) Rationale Management in Software Engineering, pp. 1–45. Springer, Heidelberg (2006)
36. OMG: Software Process Engineering Metamodel Specification. Object Management Group (2008)
37. Martínez-Ruiz, T., García, F., Piattini, M.: Managing Process Diversity by Applying Rationale Management in Variant Rich Processes. In: Caivano, D., Oivo, M., Baldassarre, M.T., Visaggio, G. (eds.) PROFES 2011. LNCS, vol. 6759, pp. 128–142. Springer, Heidelberg (2011)
38. Martínez-Ruiz, T., García, F., Piattini, M., Münch, J.: Applying AOSE Concepts for Modeling Crosscutting Variability in Variant-Rich Processes. In: Biffl, S., Koivuluoma, M., Abrahamsson, P., Oivo, M. (eds.) 37th Conf. on Software Engineering and Advanced Applications, SEAA, pp. 334–338. IEEE, Oulu (2011)
39. Martínez-Ruiz, T., García, F., Piattini, M.: Enhanced Variability Mechanisms to Manage Software Process Lines. In: 16th European Systems & Software Process Improvement and Innovation, pp. 12.13-12.23. Publizon, Alcalá de Henares, Madrid (2009)
40. Martínez-Ruiz, T., García, F., Piattini, M.: Towards a SPEM v2.0 Extension to Define Process Lines Variability Mechanisms. In: Lee, R. (ed.) Software Engineering Research, Management and Applications. SCI, vol. 150, pp. 115–130. Springer, Heidelberg (2008)
41. ISO/IEC: Guidance on Project Management. vol. ISO 21500:2012(E). ISO IEC, Switzerland (2012)
42. Calero, C., Ruiz, F., Piattini, M. (eds.): Ontologies for Software Engineering and Software Technology. Springer (2006)
43. Guide, P.A.: to Project Management Body of Knowledge. Proyect Management Institute (2000)
44. OMG: Semantics of Business Vocabulary and Business Rules Object Management Group (2008)
45. OMG: Business Process Modeling Notation Specification. OMG (2006)
46. OMG: Organization Structure Metamodel Specification. Object Management Group (2009)
47. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.: Feature-oriented domain analysis (FODA) feasibility study. SEI, Carnegie Mellon University (1990)