Managing Process Diversity by Applying Rationale Management in Variant Rich Processes

Tomás Martínez-Ruiz, Félix García, and Mario Piattini

Alarcos Research Group, Institute of Information Systems and Technologies,
Escuela Superior de Informática, University of Castilla-La Mancha,
Paseo de la Universidad 4, 13071 Ciudad Real, Spain
{tomas.martinez,felix.garcia,mario.piattini}@uclm.es
http://alarcos.esi.uclm.es

Abstract. Process diversity arises as software processes are influenced by organization, project and other contextual factors. Managing this diversity consists of considering how these factors actually modify the process. Variant rich processes offer support for process tailoring, but they do not currently link these changes with the business factors motivating them. The lack of decision traceability signifies that variant rich processes are not suitable for addressing process diversity. This article aims to fill this gap by applying rationale management to supporting decision-making when tailoring processes. Rationale management has become one of the main assets in variant rich process tailoring, since it handles how context-related factors are transformed into real variations in the tailoring process, as a consequence of well-reasoned and traceable steps. An application study shows how rationale provides useful mechanisms with which to tailor a process according to its context of enactment.

1 Introduction

Software processes are implemented in organizations whose characteristics, teams and people affect how the software process will actually be instantiated [1]. Software processes therefore diverge over time according to the projects or teams to which they are applied [2], while they are strongly based on international process standards [3]. Software process development organizations must therefore handle the diversity of processes by means of considering how they vary from the standard process, and more specifically, what exact variations standard processes experience to allow them to be adapted according each context.

Variability in software processes supports tailoring, and these therefore meet diversity in their enactment context. In previous works we have developed the Variant Rich Process Paradigm (VRP), which supports software process variability by following the Product Line Engineering (SPLEs) and Aspect Oriented Software Engineering (AOSE) approaches [4, 5]. The paradigm has been implemented in vSPEM notation [6, 7], which enhances SPEM 2.0 to support the modelling of Variant-Rich Processes. vSPEM is also the core notation of SPRINTT, a framework to support the institutionalization of software processes [8]. This framework can help users to define and institutionalize process variations and therefore supports the management of process

D. Caivano et al. (Eds.): PROFES 2011, LNCS 6759, pp. 128–142, 2011.

diversity, but the framework lacks support to link these variations with the real causes in the context motivating the change. This lack actually signifies two main disadvantages: variations are neither traceable nor justifiable, and knowledge cannot be extracted from them, since the decisions about variation in a process are human-based activities.

Human based decisions have been supported in software engineering by means of Rationale Management [9]. Rationale management deals with managing all the reasoning that goes into determining the design of an artifact [9]. In fact, stored and retrieved information about decisions during the creation of a software product is useful in other different phases of this development or maintenance. This information may also be useful in making similar decisions in similar projects. In fact, design Rationale has been successfully implemented in software engineering approaches such as product lines [10], and is also used in software process evolution [11]. As a result of both successful implementations, it may be applied to variant rich processes.

This paper tackles the inclusion of Rationale Management in the previously developed Variant Rich Process Paradigm and in the vSPEM notation. Rationale is considered as the cornerstone that connects the process' context, the variant rich processes and all the tailored processes created from them. As it contextualizes variations, it also supports the combination of context and project performance knowledge in institutionalizing processes with SPRINTT.

This paper is organized as follows. Section 2 summarizes the state of the art with regard to rationale and knowledge storing in software engineering. Section 3 shows how rationale can contribute to process institutionalization in the SPRINTT framework. The approach for managing rationale in variant rich processes is presented in Section 4. Section 5 shows the application of rationale in tailoring a software process in a company. Finally, conclusions and future work are presented in Section 6.

2 State of the Art

Rationale management is widely defined in [9]. Schneider [12] proposes to use it in a manner which is as embedded as possible in the actually tasks with the aim of minimizing effort, and considers the benefits to be higher than the effort involved in creating it. In addition, Lee [13] describes the six main issues concerned with managing rationale, including the different storing layers and elements. As a result, it has been used in software engineering [14], and has been specially applied in deciding variations of product lines. Knodel and Muthig [10] propose a five-step approach for documenting variations. However, previous to these works, there were some approaches for managing traceability in product lines [15, 16]. All of them have the same main objective: linking variations with the actual causes motivating them, and offering the best support with regard to deciding variations, but they do not provide mechanisms with which to support variation knowledge reuse.

On the other hand, some approaches apply rationale to software processes. Ocampo et al. [11] propose using rationale for process evolution. Nkwoka et al. [17] also use rationale in an environment for process improvement. Both approaches focus

on enhancing processes by means of rationale, so all of them evolve or improve processes by using well-defined variations from variant rich processes.

With regard to the management of Process Diversity, it is important to note that this is not a new topic in software processes, as it has already been applied to support development and maintenance activities. In fact, Sutton et al. suggested it in 1996 when they discussed the programming of software processes, since programmed processes diverge each other [18]. In the field of process tailoring some works can be considered such as Simidchieva et al. [19], Barreto et al. [20]. These works tackle diversity but they lack of specific support for linking variations with actual causes and learning from variations generated knowledge. Caivano et al. propose the use of patterns in making decisions about the processes and solving diversity [21], which strongly link modifications with the causes that originate them. Henninger [22] uses the experience factory approach from Basili [23] to design a process diversity knowledge storing system in which adaptations are strongly performed by means of rules.

In summary, the related works are mainly focused on applying variability mechanisms for process tailoring or applying rationale to processes, but they do not tackle both approaches in an integrated manner. Moreover, they do not consider that process tailoring could benefit from rationale management just as much as products do, since product and process tailoring are similarly managed. The work presented herein aims to enhance the VRP tailoring approach with rationale for driving variations with the intention of obtaining advantages such as traceability, justification of variations or knowledge reuse, and of therefore optimizing variations when tailoring variant rich processes.

3 Rationale in the SPRINTT Institutionalization Framework

Institutionalization constitutes the superlative degree of process tailoring. It implies process adaptation from the organization's standard processes. The SPRINTT environment was designed to provide support in process institutionalization [8], as it allows feedback in a continuous four step cycle (tailoring, execution, analysis and standardization), and thus helps to determine which variations are better than others and promote their re-use: they are *validated*.

A comparison of processes with cars may help the reader to understand the rationale needed. Hire car companies do not classify cars according to the elements of which they are composed (engine, battery…) or by names, since this would not be logical However, if we wished to make a long journey, a car hire company would offer us a private car, if we had a large family they would give us a mono van, and if we were moving house they would offer us a removal van. This does not mean that any of these models is always better than the others, but all of them fit particular usage circumstances better than the others.

Similarly, variants in a variant rich process need to be stored according to their context. SPRINTT proposes to validate the variations by comparing tailored process definition with execution logs, but a good variation in a process may be clearly bad in the case of readapting the same variant rich process in order to fit the characteristics of another totally different context. Rationale management therefore plays a principal role in linking and justifying variations with their application context, and then providing validated and *contextualized* variants. The context knowledge is therefore

stored with any other project or process knowledge. Moreover, tailoring new processes based on the knowledge base also takes into account the fact that *the variations were well enacted in processes tailored to similar contexts.*

Fig. 1 describes the main elements in the SPRINTT cycle, and how these interact with rationale in order to define the contextualized variants. If we start from *the process adaptation* step, both rationale and the tailored process are created by taking characteristics from the enactment context, and they are both stored. Process is enacted in the corresponding project (*process enactment* step), within its context, and is also stored. The post-mortem *analysis* step retrieves tailored processes and projects, and decides the validated variations, i.e. those whose changes were correct. At this point, the validated variants are those which are well enacted. As these are identified, they could be realized again, but *when* and *how* can they be satisfactorily reused?

The *standardization* step takes the validated variations and contextualizes them by using the previously stored rationale (including context and justification). That is, it specifies the context in which the validated variations were used, and supports the reuse of the variations in similar contexts.

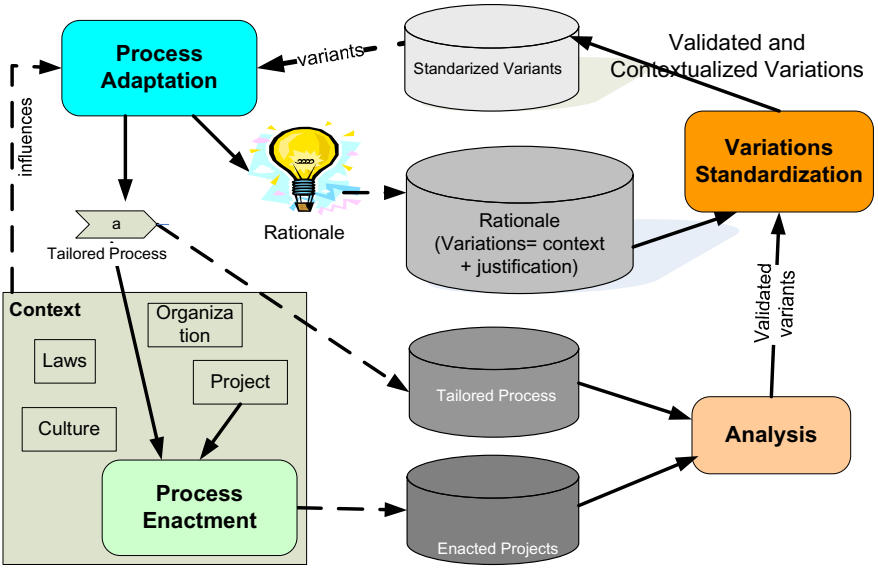

Fig. 1. SPRINTT Institutionalization Cycle with Rationale Management

Tailoring support in the sprint environment is based on controlled and scoped variations. In previous works we have developed the vSPEM notation, which adds the new package *Variations* to SPEM, supporting on-point and crosscutting variations in software processes. On-point variations are defined by means of variation points and variants of the process composing elements by applying the Software Product Lines (SPLEs). Crosscutting variations are defined by applying the Aspect Oriented Software Engineering (AOSE) approach, and they support the encapsulation of several of these on-point variations in process aspects, providing the capability of crosscutting tailoring software processes.

4 Rationale in Tailoring Variant Rich Processes

Rationale is the basis by which to optimally tailor software process from the viewpoint of variant-rich processes. vSPEM supports it through its inclusion in a new package, the *VRichProcess,* which is focused on tailoring software processes by means of variant rich processes (based on the variability elements in the *Variation* package) with rationale to handle how to carry out variations.

Fig. 2 shows the new elements defined in the *VRichProcess* package and their relationships. The elements, which inherit from the *Variations* package, are represented with white boxes. The new elements allow us to describe the context in which a tailored process is applied, which is used to tailor it (elements filled in grey), and to manage the rationale from the tailoring in order to determine how to transform the characteristics of a context into the optimal variations in the variant rich process (elements marked with bold border). These elements apply the storage layers defined by Lee [13], and their intention is to capture information as it is produced, from the problem description (project and variant rich process) to the solution (tailored process to that project), that is from the elements on the left to those on the right of the model shown in Fig. 2. They also elicit the main assets in consistently building the solution.

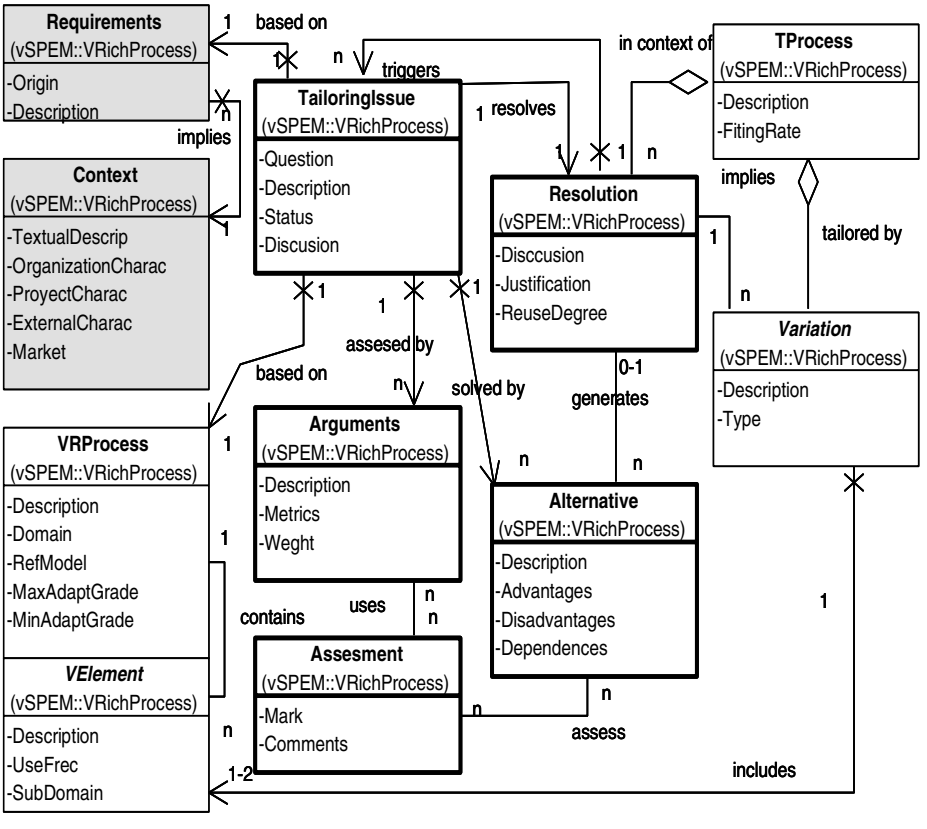

Fig. 2. Elements supporting rationale in the vSPEM notation

The white and non-marked classes in Fig. 2 inherit from homonymous classes in the *Variations* package and they support the process variability definition and their usage in tailoring processes. They also add some new attributes to support rationale, as Tables 1 and 2 show.

Table 1. Description of the elements defining variability in processes

Element	Description
VRProcess	Represents a process with variability (variants, variation points, process aspects). Attributes: *Domain,* which defines the domain the variant rich process is designed in (*real time software, systems, management, embodied, Artificial Intelligence*, etc); *Description* allows the broad definition of the variant rich process; *RefModel* specifies the reference models the variant rich process is based on, if any; *MaxAdaptGrade* and *MinAdaptationGrade* indicate the maximum or minimum adaptation grade of the variant rich process. The former corresponds with the maximum number of variation points defined into the variant-rich process plus the maximum number of process aspects that could be used in the VRP; the latter corresponds with the mandatory number of variation points included in the variant-rich process.
Velement	Variable element in a variant rich process. There are three types: variation points and variants (for on-point variations) and process aspects (for crosscutting variations). It includes three attributes: the S*ubDomain* it is designed for, if any; a long *description*; and a ratio of frequency of use, *UseFrec*.

Table 2. Description of the elements defining tailored processes

Element	Description
Variation	Describes the variations (both on-point and crosscutting). Attributes: The *Type* differentiating the on-point and crosscutting variations; a detailed *description* of the '*how to*' of the variation.
Tailored Process	Represents the tailored process created from the variant rich process with the characteristics of the context. It includes a detailed *Description*, and a *FittingGrade* signifying how well the tailored process fits the context.

The *VRichProcess* package include the *Context* and *Description* elements (filled in grey) with which to include knowledge about the *context* in which the processes are enacted, and supports in a high abstraction level of the goals involved in tailoring any process (Table 3).

Specific support for Rationale is provided by means of five elements detailed in Table 4. These elements are consistent with the explicit representation of the decision sub-layer proposed in [13]. They are also the *core* of the *VRichProcess* package (marked with a bold border).

In summary, the rationale elements included in the *VRichProcess* package specify the way in which the tailoring knowledge is created and used in the process institutionalization cycle (see Fig. 1). Three main objectives are fulfilled by means of these

Table 3. Description of the elements in the Context description area

Element	Description
Require-ment	Process tailoring requirements which the variant rich process need to achieve in order to fit the context as well as possible. Attributes: a *description* and *origin,* which indicates whether the requirement is owing to the project, the organization or external characteristics.
Context	Represents the context in which the process will be executed. Attributes: *TextualDescription* of the organization. *OrganizationChar* specifies any characteristic of the organization affecting the process enactment (such as the segment they work, the number of employees…); *ExternalChars* defines the context in which the organization is, but which cannot be controlled or changed (laws, culture etc); *ProjectChars* includes the project plan; *Market* indicates the type of market the organization uses to develop software (healthcare, army…).

Table 4. Description of the elements in the Rationale set

Element	Description
Tailoring Issue	Represents a tailoring issue of the variant rich process, which is a change in the variant rich process in order for the tailored process to meet one specific requirement. Attributes the *question* should satisfy; a complete *description*; the status (open or closed) of the tailoring issue; and any *discussion* needed to decide about the tailoring issue.
Alternatives	Proposals for resolving the tailoring issue. Attributes: a long Description; the set of *Advantages* and *Disadvantages*; and a list of the *Dependences* it could generate. Since variations are not independently executed, variability mechanisms also consider dependences between them [6, 7], and these dependences are therefore included in the documentation of all the alternatives. In fact, the dependences of an alternative are an important point when it is assessed.
Arguments	Criteria for deciding how good the alternatives are. Attributes: a long *Description*; a *Metric* for assessing and comparing the alternatives; and the *Weight*, defining the different significance of the arguments.
Assessment	Represents the assessment of each argument in each alternative. It includes two attributes: the *Mark* (how well the alternative to that argument is satisfied) and the *Comments* about the assessment.
Resolution	Selected alternative to resolve the tailoring issue by means of some variations. Attributes: the *discussion* about analyzing alternatives, and the *justification* of why the alternative must be selected; *ReuseDegree,* indicating from 0 to 1 to what extent the resolution is based on previous ones. When the alternative becomes a resolution, its dependences become new triggered tailoring issues to ensure consistence.

mechanisms: traceability from end to end (context and variations, respectively); justifiability, since through rationale it would be possible to ensure that each variation is

created as the best solution after assessing all the alternatives, making the variant rich process fit the process context; knowledge reusing since the approach allows the knowledge to be extracted from the variations and managed.

5 Application Study

An application study is being conducted on the use of the vSPEM notation (including the rationale management) as the core mechanisms for applying the MEDUSAS methodology [24] in a given project. MEDUSAS is an environment containing the processes, which aims to support the evaluation and improvement of the Usability, Security and Maintainability of software products. To achieve its purpose, MEDUSAS is composed of a methodology to guide organizations in carrying out the quality assurance process and metrics, indicators and guidelines to support the evaluation process.

The methodology includes the quality assessing and support processes, clearly defining *who*, *what* and *how* quality is assessed. It contains three processes, the *Quality Assessment Process*, which is the main process and is composed of four phases (*Planning* to establish the assessment contract and specify the assessment plan; *Specification* to determine the scope of the quality assessment project, and what to assess from the products; *Execution* which includes all the assessment activities; and *Ending* which is focused on realizing the assessment report); and the *Quality Assessment Management*, and the *Infrastructure Management* processes, which support the first process.

There are two ways in which to apply MEDUSAS: by considering the methodology as another asset in the client organization and guaranteeing alignment between its processes and those from the development cycle (on-site mode), or by applying MEDUSAS to ensure the quality of the artefacts that a third organization is developing for the client (external assessment mode). Table 5 shows a description of the variant rich process of MEDUSAS developed according to the rationale structure presented in Section 4. This VRP provides mechanisms for tailoring MEDUSAS methodology according to the different contexts in which it could be used. The defined variations are focused on two main points: the first is to provide the methodology with the profile, in accordance with the assessment type (internal or external); this is managed by means of two process aspects dealing with crosscutting variations; each of them varies some activities and tasks in all three processes, in accordance with the type of implementation. The second includes tailoring support depending on whether certain roles take part in the project, the standards they use, and the list of evaluating work products, among other aspects. All these are on-point variations and are defined by means of variation points and variants. Table 6 lists some of the variation elements included in the variant rich process, These are variants, variation points and process aspects. However, a detailed description of them is not provided here owing to space limitation (a definition of these concepts appears in [6, 7]). This study presents the first occasion on which the variants were used to tailor a process by applying the rationale steps.

Table 5. Description of the MEDUSAS Variant-rich process

Description	*Presented above*
Domain	Quality Assessment
Reference Model	ISO/IEC 29119, ISO/IEC 25020, ISO/IEC 25000
MaxAdapGrade	17 (15 *variation points* +2 *process aspects*)
MinAdaptGrade	8

5.1 Description of the Tailoring Context

The MEDUSAS Variant Rich Process is applied in the context of a Spanish *organization* focused on ensuring quality in the field of software testing, mainly to other enterprises or public institutions. It works in *nearshoring* mode since its offices are in Ciudad Real (Spain). It must apply some Spanish laws, such as the *Personal Data Protection Law* (LOPD), *or the Information Society Services Law* (LSSI).

The project in which the MEDUSAS will be applied is that of ensuring the quality of the software that a third organization is developing for a *client*. Table 7 resumes the information about the context, according to the context structure Section 4 presents. Based on the previously described context, some requirements for process tailoring appear, and these are listed in Table 8.

Table 6. Set of variability elements in the MEDUSAS variant rich process

Id	Description	Use Freq	Subdom.
1	Process Aspect "On-site".	First Use	n/a
2	Process Aspect "External".	First Use	n/a
3-5	Role variation point in some activities.	First Use	n/a
6-8	Role variants concerning the "*Client Team*", "*Quality Responsible*" and "*External Team*" roles.	First Use	n/a
9	Work product variation point related to the "*Deciding Assessment Model*" activity.	First Use	n/a
10	Work product variant "Standard X".	First Use	n/a
11,12	Activity variation points in the *Specification* Phase.	First Use	n/a
13,14	Variants of some optional activities.	First Use	n/a

Table 7. Characterization of the context in which the Variant Rich Process is applied

Textual Desc.	*Presented above*
Organization Characteristics	It works ensuring software quality in the field of software testing. It includes computer experts from the UCLM.
Project Characteristics	It focuses on assessing the quality of the software that a *third organization* is developing for the *client.* It is realized from the organization's offices (externalized mode).
External Char.	Projects must meet Spanish laws such as *LOPD* and *LSSI*.
Market	External Quality Assessment

Table 8. Set of requirements for a process meeting the organization's needs

Id	Origin	Description
1	Project	It must be adapted to be applied in an externalized context.
2	Organization	The complexity of the methodology must be simplified.
3	Organization	The number of roles involved must be minimized.
4	Project	The number of activities and tasks must be reduced, if possible.
5	Project	Any adaptation must affect the quality of the methodology.
6	Project & external	The project and external laws force the standard for quality assessment X to be used.

5.2 Rationale in Tailoring MEDUSAS to the *Organization*

If we consider the requirement the context of the *Organization* presents, then the variant rich process of MEDUSAS needs to be tailored. The matching of the tailoring requirements (Table 6) with the variability elements in the variant rich process (Table 8) generates certain *Tailoring Issues*, as Table 9 details. These tailoring issues propose the use of some of the variability elements in the MEDUSAS VRP in order to fit the requirements to the specific context that the executing project demands.

Satisfying the tailoring issues presented in the Table 9 generates some alternatives, as Table 10 lists; the resolution of the tailoring issue is obtained from these. Table 10 does not show certain details regarding the application of the variability elements owing to confidentiality constraints, but it depicts the different options that could be used when tailoring the software process. These alternatives are assessed according to the arguments presented in Table 11.

Table 9. Tailoring Issues for tailoring the variant rich process to the *Organization*

Id	Req.	Question	Description	Status	Discussion
1	1	The possibility of using the process aspects to adapt the methodology according to the execution mode.	The VRP contains some aspects to….	Open	It might be interesting to…
2	2, 3, 5	The possibility of reducing those optative roles	Some roles are optative...	Open	It might be a good idea…
3	5, 6	The possibility of assessment using the standard X	The standard used may…	Open	…
4	2, 4, 5	The possibility of reducing optative tasks and activities	Removing the optative activities…	Open	…

Table 10. Description of the alternatives for solving the previous tailoring issues

Id	Description	Adv. /Disad	Dep.
1.a	Uses the process aspect "*On-site*".	Confidential	None
1.b	Uses the process aspect "*external*".	"	"
1.c	Does not use the process aspects.	"	"
2.a	Considers the role, "*Client Team*".	"	"
2.b	Does not consider the role "*Client Team*".	"	"
2.c	Considers the role, "*Quality Responsible*".	"	"
2.d	Does not consider the role "*Quality Responsible*".	"	"
2.e	Considers the role, "*External Team*".	"	"
2.f	Does not consider the role "*External Team*".	"	"
3.a	Considers using the standard "X".	"	"
3.b	Considers using the standard "Y".	"	"
3.n	Considers not using standards.	"	"
4.a	Considers carrying out some optative tasks	"	"
4.b	Considers not using these optative tasks.	"	"

Alternatives are assessed by means of arguments, resulting in a matrix. Table 12 summarizes this matrix, showing alternatives in rows and arguments in columns. Because of confidentiality constraints, we show an estimation of how good they are. As a result, we assess them by using linguistic tags: *high*, *medium* and *low*.

As a result of the evaluation of alternatives, some are selected to solve each tailoring issue. These alternatives, and the tailoring issue they solve, are described in Table 13. Owing to space constraints, any of the presented resolutions triggers a new *Tailoring Issue*.

Table 11. Description of the Arguments for assessing the Alternatives

Id	Description	Metrics	Weight
a	The variations do not remove functionality.	The functionality is reallocated throughout the process	0,75
b	The Assessment's Quality and Confiability are not affected.	Variations do not affect the assessment quality	1
c	The process is complete.	No mandatory variation points are empty.	1
d	Alternatives fit the assessment client's necessities as much as possible.	The characteristics of the alternative meet the context requirements.	0,5

Table 12. Overview of the assessment of the alternatives by means of the arguments

	a	b	c	d		a	b	c	d
1.a	low	low	high	low	**3.a**	**med.**	**high**	**high**	**med.**
1.b	**med.**	**high**	**high**	**med.**	3.b	low	low	high	low
1.c	low	low	low	low	3...				
2.a	med.	high	high	low	3.n	low	low	high	low
2.b	**med.**	**high**	**high**	**med.**	4.a	med.	low	high	low
2.c	med.	high	high	low	**4.b**	**med.**	**high**	**high**	**med.**
2.d	**med.**	**high**	**high**	**med.**	**4.c**	**med.**	**high**	**high**	**low**
2.e	**med.**	**high**	**high**	**med.**	4.d	med.	low	high	med.
2.f	low	low	high	med.					

Table 13. Resolutions created after assessing the alternatives

Tailoring Issue	Alternatives Selected	Justification
1	1.b	The process aspect meets the context requirements
2	2.b, 2.d, 2.e	The "Client Team" and "Quality Responsible" optative roles can be deleted, but the "External Team" cannot, because it ensures the quality…
3	3.a	The variant rich process allows precisely the standard the context requires to be used.
4	4.b, 4.c	One optative activity is not included, but the other is included, as it ensures the quality of the assessments.

5.3 Tailored Process from the MEDUSAS Variant Rich Process

As a consequence of the Tailoring issues described in Table 9, and the solution described in Table 13, some variations are realized. The *Organization's* tailored MEDUSAS process effectively considers the enactment context. The principal asset is that the "*External* aspect" is used, signifying that the methodology has been explicitly configured to carry out external validations, which is the main characteristic of the enactment context. In addition, some roles, such as the "*External Team*" have been configured according to factors such as the quality assurance level, or the human resources available in the *Organization*. Both of them clearly influence the enactment, and not taking them into account may compromise the entire project execution. Some other adaptations regarding the standards used, or skipping certain activities, influence whether the resulting process is or is not in accordance with external laws and the budget.

All of the above signifies that the tailored process fits the enactment context to a high degree. It is therefore possible to ensure that the enactment consistence is higher than if the process to be enacted was the standard (and strict) version of MEDUSAS. The variations are listed in Table 14 and configure the process described in Table 15.

Table 14. Variations realized as a consequence of the resolutions

Resolution	Variable Elements	Type	Description
1	2	Crosscut.	The External Aspect is activated.
2	5,8	On-point	The role variant "*External Team*" is placed in the role variation point, in the corresponding activity.
3	8,10	On-point	The work product variant "Standard X" is placed in the work product variation point related to the "*Deciding Assessment Model*" activity.
4	12,14	On-point	The optative activity variant is placed in the corresponding activity variation point.

Table 15. Tailored process configured by using the MEDUSAS VRP and the context

Textual Description	*Described above.*
Fitting Rate	High.

5.4 Lessons Learned from the Proposal of Application Study

Tables 6 to 16 show an excerpt of the use of rationale in tailoring a software process. These details meet the three main objectives that rationale management supports. Firstly, the excerpt shows easy tracing from the inputs in the process tailoring (the variant rich process and the context) to the resulting tailored process. Secondly, after eliciting alternatives and comparing them, it justifies each variation as the best solution for solving each tailoring issue. Thirdly, rationale allows knowledge to be stored and permits us to learn from what the variations realize based on the context and the variant rich processes.

The application study also provides us with some other advantages we had not previously considered. Firstly, as rationale elements are built based on others, rationale offers the sequence steps with which to consistently tailor a software process. These steps really define a process tailoring activity. The sequence of steps also provides another advantage in that they are well defined and could be easily included in a tool giving support to process tailoring. This could transparently manage rationale from the user viewpoint.

In addition, since elements supporting rationale are well defined and well related, the tailoring steps may be easily recomputed in the case of input change. New variations are then obtained after any improvement or update is made in the context or the variant rich process. To sum up, we have shown how rationale management optimizes variant rich process variations, and software processes therefore meet process diversity challenges.

6 Conclusions and Future Work

Process diversity is as real in software processes as they enact and it is as important as using a capable process. Managing process diversity is therefore essential in attempting to perform good process enactments, and this therefore implies ensuring that processes meet the characteristics of their context. That implies tailoring the process by considering several context factors. Variability has typically been used for the consistent and easy tailoring of software processes, but totally exploiting variability mechanisms is not possible unless they are suitably managed. Tailoring not only implies realizing certain variations in the software process, but also needs to ensure that the variations make the process fit its context; these variations actually come up to and meet the tailoring objectives. Moreover, this knowledge needs to be used again in tailoring other processes and in improving the process tailoring activity.

Designing a process variability notation, such as vSPEM or the Variant Rich Process (VRP) paradigm, supports half of the problem of tailoring processes. But this notation is not fully usable since it does not relate tailoring causes to tailoring results, from end to end. Rationale Management can be used to solve this problem since it supports relationships between the process context and the actual variations, it allows each variation to be justified not only as a good variation, but ensures that all the variations are the best ones for tailoring the whole process. It also implies that knowledge about the tailoring is stored when it is produced, and retrieved again if it is necessary. Applying rationale in process tailoring implies executing several well-defined steps and documenting them. This extra work might initially be viewed as a disadvantage; however it forces structured software process tailoring in the same way that software processes structure software development. The advantages of applying this rationale therefore compensate for the extra work that it involves.

Rationale management has been merged with previous variability mechanisms in the VRP paradigm and notation. This also qualitatively improves the process institutionalization framework, as tailored processes are linked with the problem they actually solve. The application study shows that rationale management supports the achievement of all the advantages that were assumed, and offers other advantages. In addition, it structures and makes the whole tailoring activity consistent by dealing with improving this activity itself.

Future work is focused on validating the vSPEM language, considering that rationale is now the cornerstone of the variability mechanisms, and the most important asset in tailoring processes. This will provide us with feedback in order to improve our approach. We are carrying out some other case studies in the context of real variant rich processes tailoring and we hope to develop a tool to support the vSPEM language, as was mentioned in the section concerning the lessons learned. This tool will helps us to carry out the case studies automatically, thus reducing the effort needed in documentation and assisting during the decision making process.

Acknowledgments

This work is partially supported by the Program FPU of the Ministerio de Educación, and by the INGENIOSO (JCCM, PEII11-0025-9533), MEDUSAS (CDTI (MICINN), IDI-20090557), PEGASO/MAGO (MICINN and FEDER, TIN2009-13718-C02-01), and ALTAMIRA (JCCM, Fondo Social Europeo, PII2I09-0106-2463) projects.

References

1. Lindvall, M., Rus, I.: Process Diversity in Software Development. IEEE Software, 14–18 (2000)
2. Siebel, N.T., Cook, S., Satpathy, M., Rodríguez, D.: Latitudinal and Longi-tudinal process diversity. Journal of Software Maintenance and Evolution: Research and Practice 9, 9–25 (2003)
3. Deck, M.: Managing Process Diversity While Improving Your Practices. IEEE Software, 21–27 (2001)
4. Filman, R.E., Elrad, T., Clarke, S., Aksit, M.: Aspect-Oriented Software Development. Addison-Wesley, Boston (2004)
5. Clements, P., Northrop, L.: Software Product Lines. Practices and Patterns. Addison-Wesley, Boston (2002)
6. Martínez-Ruiz, T., García, F., Piattini, M.: Towards a SPEM v2.0 Extension to Define Process Lines Variability Mechanisms. In: Lee, R. (ed.) SERA. SCI, vol. 150, pp. 115–130. Springer, Praga (2008)
7. Martínez-Ruiz, T., García, F., Piattini, M.: Framework to the Standardization and Institutionalization of Software Processes. (2008)
8. Martínez-Ruiz, T., García, F., Piattini, M.: Process Institutionalization using Software Process Lines. In: Cordeiro, J., et al. (eds.) ICEIS 2009. ISAS, pp. 359–362 (2009)
9. Dutoit, A.H., McCall, R., Mistrík, I., Paech, B.: Rationale Management in Software Engineering: Concepts and Techniques. In: Dutoit, A.H., et al. (eds.) Ra-tionale Management in Software Engineering, pp. 1–45. Springer, Heidelberg (2006)
10. Knodel, J., Muthig, D.: The Role of Rationale in the Design of Product Line Architectures – A Case Study from Industry. In: Dutoit, A.H., et al. (eds.) Rationale Management in Software Engineering, pp. 297–312. Springer, Heidelberg (2006)
11. Ocampo, A., Münch, J.: Rationale Modeling for Software Process Evolution. Software Process Improvement and Practice 14, 85–105 (2008)
12. Schneider, K.: Rationale as a By-Product. Rationale Management in Software Engineering, 91–110 (2006)
13. Lee, J.: Design Rationale Systems: Understanding the Issues. IEEE Expert 12, 78–86 (1997)

14. Dutoit, A.H., mcCall, R., Mistrík, I., Paech, B.: Rationale Management in Software Engineering. Springer, Heidelberg (2006)
15. Mohan, K., Ramesh, B.: Ontology-based Support for Variability Management in Product and Service Families. In: Proc. of the HICSS 2003 (2003)
16. Kumar Thurimella, A., Wolf, T.: Issue-based Variability Modelling. In: GREW 2007, pp. 11–22 (2007)
17. Nkwoca, A., Hall, J., Raspanotti, L.: Design Rationale Capture for Process Improvement int he Globalised Enterprise: An Industrial Study. Faculty of Mathematics and Computing. The Open University, Milton Keynes (2010)
18. Sutton, S., Osterweil, L.J.: PDP: Programming a Programmable Design Process. In: 8th Int. Workshop on Software Specification and Desing, pp. 186–190 (1996)
19. Simidchieva, B.I., Clarke, L.A., Osterweil, L.J.: Representing Process Variation with a Process Family. In: Wang, Q., Pfahl, D., Raffo, D.M. (eds.) ICSP 2007. LNCS, vol. 4470, pp. 121–133. Springer, Heidelberg (2007)
20. Silva Barreto, A., Murta, L., Rocha, A.R.: Software Process Definition: a Reuse-Based Approach. In: XXXIV Conferencia Latinoamericana de Informática, Santa Fe, Argentina, pp. 409–418 (2008)
21. Caivano, D., Visaggio, C.A.: Process Diversity and how Practitioners Can Manage It. Novatica V (2004)
22. Henninger, S.: An Environment Supporting Software Process Standardization and Innovation
23. Basili, V., Caldiera, G., Rombach, D.: The Experience Factory. In: Marciniak, J. (ed.) Encyclopedia of Software Engineering, pp. 469–476. John Wiley, Chichester (1994)
24. Torre, D., Blasco, B., Genero, M., Piattini, M.: CQA-ENV: An integrated environment for the continuous quality assessment of software artifacts. In: The 8th International Conference on Software Methodologies, Tools and Techniques (SoMeT), Praga, vol. 199, pp. 148–164 (2009)