RiSE-DA: A Lightweight Domain Analysis Process And Its Industrial Evaluation

Tassio Vale
Federal University of Recôncavo da Bahia
Cruz das Almas - Brazil
Email: tassio.vale@ufrb.edu.br

Iuri Santos Souza
and Eduardo Santana de Almeida
Federal University of Bahia
Salvador - Brazil
Email: {iurisin,esa}@dcc.ufba.br

Ivonei Freitas da Silva
State University of West Paraná
Cascavel - Brazil
ivonei.silva@unioeste.br

***Abstract*—In the Brazilian software development scenario, most software development companies focus on building products in the same business domain. However, they tend to perform ad-hoc reuse practices and show interest on the adoption of agile methods. However, the combination of systematic reuse activities and agile practices is not trivial. In this paper, we present a lightweight domain analysis process called RiSE-DA by describing its activities, tasks, roles, and artifacts of the process. This process is evaluated through two case studies performed in companies with different sizes, constraints, and business knowledge. As results, we can conclude the proposed process provides a set of comprehensive domain assets and addresses management problems such as poor feedback, adaptability, and iterativeness.**

I. Introduction

Although the integration of domain analysis with Agile Software Development (ASD) can provide benefits, there is a lack of studies combining both practices [1]. A process for domain analysis with ASD is a challenge, since these approaches can present conflicts regarding tasks, activities, artifacts, and roles. Aiming to decrease conflicts between both approaches, companies which are applying ASD need to choose and adapt domain analysis activities for their scenarios.

This paper presents a lightweight domain analysis process called RiSE-DA by describing its activities, tasks, roles, and artifacts of the process, since the related work [2][3] do not address iterativeness, adaptability, and feedback in a lightweight domain analysis process. In addition, this process was evaluated through two case studies performed in two Brazilian companies with different sizes, constraints, and business knowledge. These results are synthesized in a cross-case analysis that investigated similarities and differences among the findings.

The remainder of this paper is structured as follows: Section II describes the activities, tasks, roles, and artifacts of the process; Section III presents two case studies performed to evaluate this proposal; Section IV discusses lessons learned from the research and development projects; and Section V summarizes the study findings and presents future work.

II. RiSE-DA

We define the roles, activities, and work products of RiSE-DA, a software modeling process combining domain analysis and Scrum practices. The process description is based on the Software Process Engineering Metamodel (SPEM) [1]. The entire RiSE-DA specification (activities, tasks, steps, roles and work products) is available on the web[2].

DOI reference number: 10.18293/SEKE2015-003

Figure 1 shows an overview of the process activities and work products (assets). The main activities are: *Pre-Analysis*, *Product Map Development*, *Major Features Priorization*, *Sub-Features Definition*, and *Commonality and Variability Analysis*. Furthermore, the main work products are: *Market Analysis Document*, *Product Map*, *Domain Backlog*, *Sprint Backlog*, *Features List* and *Feature Model*. Given space constraints, .

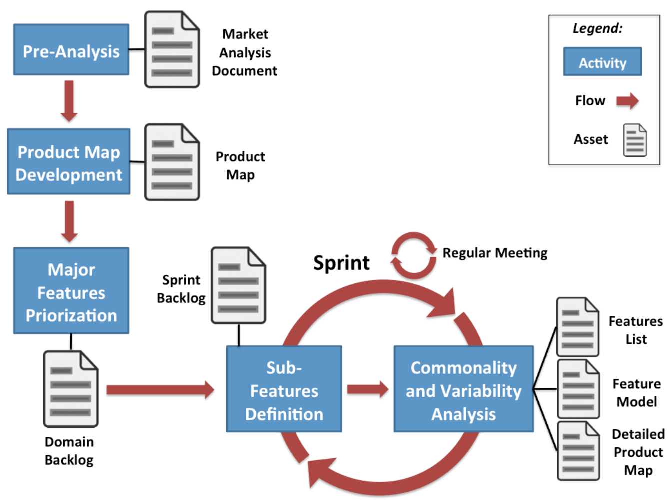

Fig. 1. RiSE-DA Overview.

Additionally, the process roles are (the definitions are exclusively available on the web): *Scrum Master*, *Business Expert*, *Domain Expert*, *Inspector*, *Legacy Systems Engineer*, *Product Expert*, *Domain Analysis Expert*, *Domain Engineer*, *Scrum Team* and *Domain Owner*.

The RiSE-DA activities and work products are described next.

A. Pre-Analysis

This activity identifies marketing strategies and other marketing details that are important to define proper features and

[1]Software & Systems Process Engineering Metamodel Specification (SPEM) - http://www.omg.org/spec/SPEM/2.0

[2]RiSE-DA is available at http://tassiovale.com/rise-da/

products to the customers. *Pre-Analysis* is divided into two tasks: *Market Analysis* (evaluates the market regarding its business goals, user profile, legal and cultural constraints, business opportunities, competitors, and other factors that the *Business expert* and *Domain expert* define) and *Marketing Strategies Identification* (defines how the products will be released to the customers considering the sale units, integration methods, installation, product maintenance, and user skill level).

B. Product Map Development

The *Product Map Development* consists of tasks to build a matrix comprising features and products, indicating, for each product, which features it implements. Kang et al. [4] define feature as "*a prominent user visible aspect, quality, or characteristic of a software system or systems*". As result, there is a common understanding and sharing of the major features and products. This activity comprises four tasks: *Products Identification* (identifies the products available in the company portfolio that share business activities in common) and *Major Features Identification* (identifies major features and provides a description of them), *Features Grouping* (arranges the fine-grained feature into major features) and *Product Map Building* (*Domain Analysis Expert* builds a map indicating, for each product, which major features it implements).

C. Major Features Prioritization

This activity prioritizes the major features using classification criteria relevant for reuse. *Major Features Prioritization* is divided into two tasks: *Domain Potential Assessment* (assessment is performed by applying on *Domain expert* a questionnaire using a set of criteria, such as domain maturity, reuse potential and risks) and *Major Features Ranking* (defines the final ranking of major features based on assessment previously applied).

D. Sub-Features Definition

Sub-features express the details of major features. The *Sub-Features Definition* specifies descriptions and properties of the sub-features for each major feature in the domain. Since the major features involve a considerable number of functionalities, an iterative process is necessary to optimize the company effort. This activity comprises four tasks: *Sub-Features Identification* (*Domain Analysis Expert* captures reusable sub-features from the *Domain Expert* or *Product Expert* through workshops), *Legacy Assets Mining* (*Domain Analysis Expert* captures reusable sub-features by studying the system-as-is, their flows, work procedures, business rules, reports about defects, user manuals, screenshots of the products, and prototypes), *Sub-Features Inspection* (*Inspector* inspects the features list in terms of non-conformities that consider aspects such as feature granularity, understanding, and duplication), and *Sub-Features Validation* (*Domain Analysis Expert* collects opinions from the *Domain Expert* and *Product Expert* to adapt the features specification whether necessary, and integrate them with the other available features).

E. Commonality and Variability Analysis

This activity specifies the commonalities and variabilities through feature model and product map work products. It defines which features are directly linked (father-son) and how they are classified (mandatory, optional, and variant). Three tasks support the *Commonality and Variability Analysis*: *Product Map Updating* (updates the product map created during the *Product Map Development* and *Sub-Features Definition* activities), *Feature Modeling* (organizes hierarchically the feature model [4] based on the features list, commonalities and variabilities provided by the *Domain Expert*, *Product Expert* or *Legacy Assets Expert*), and *Models Inspection* (verifies the *Product Map* and *Feature Model* inconsistencies).

F. Agile Practices

For the proposed process, the activities *Sub-features Definition* and *Commonality and Variability Analysis* incorporates certain practices, artifacts, and roles of the Scrum framework. These practices are: *Sprint Planning*, *Regular Meeting* and *Sprint Review and Retrospective*.

Sprint Planning: aims to find the most appropriate major features for a specific sprint that provide reuse potential for the organization. Based on the Scrum framework, the planning is divided into two parts: part one, where the major features are selected (from the domain backlog) for the next sprint (sprint backlog); and part two, where the major features selected are broken into Scrum tasks (in the sprint backlog).

Regular Meeting: adapted from the daily meeting from Scrum, the *Regular Meeting* is a practice that intends to aid the stakeholders with fast feedbacks about the sprint. The frequency of this meeting is defined according to the effort spent in the domain analysis activities (such as daily or every two days). Issues related to the process, artifacts and team are raised by the *Scrum Team*. From this meeting, some small adjustments can be performed during the sprint, since the sprint goal does not change.

Sprint Review and Retrospective: after finishing the sprint, the *Sprint Retrospective* and *Sprint Review* practices adjust and improve what is necessary for the next sprints (e.g. process and team adjustments). An initial cause analysis also is performed to explore the root of the issues during the process. In addition, the team should provide a new effort estimation for each major feature in the scope backlog in order to be more precise when performing the *Sprint Planning* part one. The *Domain Analysis Expert*, *Domain Engineer*, and the *Legacy Systems Engineer* would consider the technical issues to perform these new estimation. Technical issues can be associated to the variability implementation complexity, structuring the common and variable components or using new technologies. The *Scrum Team*, for instance, would estimate how long it will take to perform the *Commonality and Variability Analysis* for the major features.

III. Case Studies

We evaluated the RiSE-DA process qualitatively, through the case study technique [5]. Our assumption is that the proposed domain analysis process achieves acceptable results in terms of work products quality and applicability in the organizations. However, companies moving to agile principles and practices are looking for software reuse processes providing mechanisms to support faster changes in volatile business domains. In this context, process iterativeness, feedback among stakeholders, and process adaptability are essential [1].

In order to evaluate iterativeness, feedback among stakeholders, and process adaptability in our proposal, we defined the research question "*How do the stakeholders characterize the iterativeness, adaptability and feedback of the process?*". In order to enable the triangulation of information [5], the data collection procedure involved the following techniques: survey, field observation, document analysis, and focus group.

A. *Case #1: An Oil, Gas and Energy Company*

The *Case #1* was applied in an Oil, Gas and Energy company, working with software development for more than thirty years and is spread over several cities, whose main locations are Rio de Janeiro and Salvador, in Brazil. In this project, a set of applications was selected in a pilot.

A training was applied to the company employees (project participants). Presentations were prepared by the RiSE members and company members were grouped by availability. It was a training of eighteen hours divided into three days. Support materials and practical examples were used to provide a better learning.

Despite the company size, the pilot was performed with a small team, considering time restrictions for many employees participate in this project. Then, the roles were played by more than one member, i.e., the *Domain Expert* and *Business Expert* roles were performed by the same participant.

The resulting product map and feature model described three different domains, 17 major features and 96 sub-features. It took three weeks with three sprints to describe the features. Time variations occurred in the three performed sprints, with an average of four hours and thirty minutes per sprint. Twenty percent of the time (forty minutes) was spent with the agile management tasks. The company participants did not provide details about the features and product functionalities, since they were not business experts. It might have influenced the spent time.

1) Case #1 Findings: RiSE-DA fostered the iterativeness only when defining sub-features and analyzing commonalities and variabilities. There was not change for the sprint time box. The company participants argued RiSE-DA fostered the iterativeness in the project, since the sprints were short and the domain was incrementally defined.

Adaptability was also fostered when defining sub-features as well as analyzing commonalities and variabilities. The Scrum practices (retrospective, planning, review, and daily meeting) detected needs of changes in the process activities and artifacts.

Furthermore, RiSE-DA fostered continuous feedback, when defining sub-features and analyzing commonalities and variabilities. Workshops and model storming were good strategies to achieve it. The feedback among stakeholders were effective, and iterativeness provided a considerable impact on feedbacks.

Therefore, all evaluation aspects had a significant impact. Frequent feedbacks and small (but frequent) changes were possible because of the iterativeness. Feedbacks supported the changes detection, providing insights for next sprints, and adaptability supported process adjustments such as changes on the daily meeting practice (daily meeting questions changed since the *Scrum Team* worked together in many sprints.

For each new sprint with a different group of company members, the *Scrum Team* analyzed previous models and evolved them, then, obsolete features were detected and removed for next sprints. The decision of removing obsolete features occurred after the *Scrum Team* evaluate the retrospective and review results, during the sprint planning, and since the features were in the target domain of the current sprint.

Iterativeness and feedback enabled the *Scrum Team* to anticipate problems regarding the requirements, technical constraints, staff, or other external factors such as demands for product development. Adaptability had an important impact on the process performance when the adjustments on activities optimized the way as the company performed them.

B. *Case #2: An Educational Management Systems Company*

The Educational Management Systems company works with software development for the educational/scholar domain for eighteen years and is located at Salvador, Bahia, Brazil. A domain analysis project was set up to organize the customizations as derived products from a common reuse platform. Seven participants (four from the company) participated in the project.

The company members were trained before the domain analysis. Basically, the domain analysis concepts were communicated, then the case participants applied them in the company context.

As results, product map and feature model described one domain, 8 major features, and 159 sub-features (from two of the major features). The company participants were involved in other demands and the project took about six months (the amount of features also influenced the spent time). The meetings duration varied in the eleven sprints, with an average time of fifteen hours for each one. Sixteen percent of the time, approximately, was spent on sprint planning, regular meetings, retrospective, and review.

1) Case #2 Findings: RiSE-DA fostered iterativeness in the *Sub-Features Definition* and *Commonality and Variability Analysis* activities. There were changes in the sprint time box during the project due to external factors such as the legacy systems maintenance. Defects in the software caused the variations in time boxes. The company participants considered the iterativeness fostered flexibility in the project in terms of changes and reflections.

In addition, the process fostered adaptability, mainly when defining sub-features and analyzing commonalities and variabilities. The Scrum practices (retrospective, planning, review, and daily meeting) detected needs of changes in the process activities and artifacts. The participants argue the process adaptability supported the flexibility in the project, and the adaptability was frequent because of the iterativeness.

RiSE-DA also supported continuous feedback, mainly when defining sub-features and analyzing commonalities and variabilities. Workshop was a good strategy to achieve it.

Although the *Scrum Team* members were in different locations (they used tools for remote communication), the feedback among them and other team members was effective. Scrum and model storming practices supported the feedbacks. The feedbacks were continuous due to iterativeness.

Frequent feedbacks and small (but frequent) changes were possible because of the iterativeness. Feedbacks detected changes and provided it for the next iteration. Adaptability encouraged adjustments to improve feedback (e.g. the daily meeting questions changed to a simpler format, since the *Scrum Team* worked together in many sprints).

IV. Lessons Learned

In order to report the experience from the application of RiSE-DA in real-world scenarios, we have present the lessons learned as follows:

Duration of RiSE-DA Sprints. Sprints with one or two weeks were considered appropriate, because the team was learning how to perform domain analysis and understand the domain (both companies). This lesson learned favor the iterativeness (maximum of two weeks), frequent feedbacks and adaptability.

Presence of the *Domain Expert* in workshops. The *Domain Expert* has deep knowledge about the domain and he can provide frequent feedbacks. This lesson enables the identification of some obsolete features during the sprints and the changes were made earlier.

Presence of *Business Expert*. The *Business Expert* aligns business goals and market strategies of companies. In the larger company, we faced the absence of business experts and it impacted on the process results. The involved employees were from the software development sector, and provided information based the product development they are responsible for. They did not have an overall knowledge on the domain. As consequences, they provided fine-grained features, revealed difficulties to identify major features and relationships among features implemented by different products. To deal with these consequences, we combined the participants' information with legacy systems analysis.

Early changes detection. Changes detection impacted on the performance, effort, obsolete features control, risks management, and problems during the project. As a final result, the motivation in the projects was considered beyond expectations.

Deal with complex organizational structure. The complex organizational structure of the larger company influenced the results, since important products could not be included in the case, and important experts were not available. The process adaptability allowed the participants to build comprehensive domain models considering only the available information.

Feature identification. Feature was a new concept for most participants. During the initial activities, they had problems by defining the products features. We adopted two strategies to mitigate this problem: for the small company, we presented practical examples of features and investigated the functionality description provided by the participants in order to verify whether it was a feature or not; for the larger company, the participants used the activity diagram notations during model storming practices, and the domain analysis experts extracted the candidate features from these diagrams.

Sprint scope. During the sprints, participants proposed to add activities that meet other companies' objectives (e.g. combining the domain analysis process with the development of a framework). This focus shift impacted on the process results in terms of time and effort. Therefore, as lesson learned, the process must focus on defined activities and possible adaptations, without adding different ones.

Tool support for domain analysis. The available commercial tools to model domain variability provide more functionalities than needed for domain analysis, since they are expensive. Thus, it was not feasible use these tools in the context of this work. We used an unstable academic tool which caused problems during the project such as missing information. In addition, they do not address traceability among product map and feature model. It still remains an open problem in our research work.

V. Conclusion

In this paper we described activities, tasks, roles, and artifacts of RiSE-DA, a lightweight domain analysis process adapted to the Brazilian software development scenario. According to our evaluation, RiSE-DA provides appropriate domain analysis activities considering the evaluated cases.

In addition, the stakeholders manifested motivation with management aspects (iterativeness, feedback, and adaptability) of the process. Scrum practices helped domain analysis activities to find obsolete features and make changes in a faster way, decreasing effort and increasing the motivation to use the process.

Although the qualitative studies are hard to replicate and generalize, in future work, new evaluations should be performed in different scenarios to reinforce the findings in this study. We also intend to define new activities for the process (e.g. requirements, architecture, and testing) in order to build a complete domain engineering process.

References

[1] I. F. da Silva, P. A. da Mota Silveira Neto, P. O'Leary, E. S. de Almeida, and S. R. de Lemos Meira, "Agile software product lines: A systematic mapping study," *Software Practice and Experience*, vol. 41, no. 8, pp. 899–920, Jul. 2011.

[2] K. Schmid, "A comprehensive product line scoping approach and its validation," in *Proceedings of the 24th International Conference on Software Engineering*, ser. ICSE '02. New York, NY, USA: ACM, 2002, pp. 593–603. [Online]. Available: http://doi.acm.org/10.1145/581339.581415

[3] J. Bosch and P. M. Bosch-Sijtsema, "Introducing agile customer-centered development in a legacy software product line," *Software: Practice and Experience*, vol. 41, no. 8, pp. 871–882, 2011.

[4] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-oriented domain analysis (foda) feasibility study," Carnegie-Mellon University Software Engineering Institute, Tech. Rep., November 1990.

[5] R. K. Yin, *Case Study Research: Design and Methods (Applied Social Research Methods)*, 4th ed. Sage Publications, 2008.