A Method for Service Agile Development Using XP

Felipe Carvalho\textsuperscript{1,2}(Student), Leonardo Guerreiro Azevedo\textsuperscript{2}(Advisor)

\textsuperscript{1}Petrobras – Petróleo Brasileiro S/A
\textsuperscript{2}Graduate Program in Informatics (PPGI)
Federal University of the State of Rio de Janeiro (UNIRIO)
Rio de Janeiro, Brazil

\{felipe.carvalho,azevedo\}@uniriotec.br

\textbf{M.Sc. Year of Enrollment: 2012}

\textbf{Expected Conclusion: Dec/2013}

\textbf{Finished phases: Proposal Defense}

\textbf{Abstract.} Agile Methods and SOA (Service-Oriented Architecture) have emerged with similar concerns, such as: flexibility, fast response to changes, business understanding. Despite this similarity, there is no consensus about how to combine them in a method that provides a set of steps to be followed to implement flexible solutions, built incrementally and adherent to their principles. The goal of this work is to develop a method (and test it in a real-world scenario) that addresses both SOA and Agile principles, and can be easily used in organizations interested in such combination.

\textbf{Resumo.} Métodos ágeis e SOA (Service-Oriented Architecture ou Arquitetura Orientada a Serviços) surgiram em resposta a preocupações semelhantes, tais como: flexibilidade, resposta rápida para mudanças, entendimento do negócio. Apesar desta similaridade, não há consenso na literatura em como combinar estas abordagens em um método que defina de forma clara um conjunto de passos a serem seguidos para implementar soluções flexíveis, construídas incrementalmente e adherente aos seus princípios. O objetivo deste trabalho é desenvolver um método (e testá-lo em um projeto real) que trate os princípios ágeis e de SOA e que possa ser facilmente utilizado em organizações interessadas nesta combinação.

\textbf{Keywords:} Methods and Approaches for Information Systems Engineering, SOA, eXtreme Programming, Agile Methods, XP

1. Theoretical grounding

Dynamism, communication channels, global competition, and deregulated markets shape a scenario where companies worldwide communicate and negotiate (Kohlborn \textit{et al.}, 2009). Organizations are in a context where it is required to respond rapidly to market changes.

According to Lankhorst (2012), economies around the world evolved to “service economies” where the business models of traditional production-oriented organizations are repositioning and adapting themselves to a service-driven model. Kohlborn \textit{et al.}
(2009) go beyond by stating that organizations have been restructuring according to their core competencies, which demands them to establish a collaboration network with partners to deliver products to their clients, and to find agile ways of working. In this context, “agility” means the ability to handle and control unstable requirements and environments in a way that improve organization competitiveness (Lankhorst, 2012).

In this sense, Pressman (2012) states that organizations have realized that traditional ways of working cannot meet the changes demanded by clients. He also states that project development based on frozen requirements, long deadlines and distant stakeholders are not able to deliver products properly.

During 2000-year, many articles were published highlighting the issues related to documentation driven development and heavyweight processes (Beck et al., 2001). Seventeen developers and consultants met in 2001 to discuss better ways for software development and published the Agile Manifesto, which corresponds to four statements that express the main concerns for a development process, along with related principles. The methods included in the Agile Manifesto were named as “agile methods”.

SOA (Service-Oriented Architecture) is an strategy to reorganize an initially isolated portfolio of applications into an interconnected set of services, accessible by standard interfaces and communication protocols (Papazoglou, 2003). The construction of applications is largely simplified through the composition of existing services (Erl, 2005).

Gu & Lago (2007) state that service-oriented systems are developed differently from traditional ones. There are greater challenges, such as, reuse of assets, stakeholder involvement, deep understanding of business model, service distribution across enterprise boundaries, and business-IT alignment.

Given the contexts where service-orientation and agility emerged, as presented by Strode (2006), Abrahamsson (2002), Erl (2007) and Gu & Lago (2007), it is remarkable that both paradigms share similar concerns, such as: new ways of working, flexibility, responsiveness to changes, stakeholder involvement, and business understanding. Nevertheless, despite these similarities, there is no consensus about the applicability of agile methods to the development of systems employing the SOA principles. Krogdahl et al. (2005) point that some researchers consider the combination of these approaches equivalent to combining “oil & water”.

The purpose of our research is to propose a method that uses XP’s principles, values and practices to address SOA (Service-Oriented Architecture) challenges in a service development lifecycle. The proposed method aims to provide service-oriented solutions built incrementally with continuous feedback and flexibility to rapidly meet changes in business requirements while being adherent to SOA principles. As a result, by the end of this research, we aim to provide a process model, making explicit the involved roles, responsibilities, guidelines, activities and their input and output information, so that the process can be used in practice in real scenarios.

The focus of this work is on the service construction phase in a service development lifecycle. Such focus derives from issues found after analyzing the related works, where many authors discuss the usage of Agile Methods to build SOA solutions in a high level of abstraction. Most of them present principles and ideas without concrete steps. Hence they do not discuss how services should be constructed to result
in flexible solutions, adherent to SOA and Agile principles. The terms “construction” and “development” are used interchangeably as pointed by Brittenham (2001). We consider this phase to be responsible for the service design and implementation.

Among the agile methods we have chosen the eXtreme Programming (XP) based on the results of an extensively analysis of which agile practices (Abrahamsson, 2002; Strode, 2006) are employed by each agile method and in which software development lifecycle phase the methods work presented at <http://www.uniriotec.br/~azevedo/UNIRIO/Agile_Methods_Practices_Analysis.pdf>. XP is the method that uses most of the agile practices for "Software construction”.

2. Contributions

The main contribution of this work is the method for agile service development using XP in a Service-Oriented scenario. Other important contribution is the results of a systematic literature review presenting the state of the art of approaches for XP and SOA combination, along with related open issues.

The systematic review will enable us to analyze existing methods, evaluating whether there is already a concrete method combining SOA and XP for service development, i.e., a method that clearly presents practices that can be used in practice. As a result, this assessment will allow us to decide on extending an existing method or to propose a new method handling existing complexities and gaps.

3. Research status & evaluation

Currently, we have already reviewed SOA principles, general concepts and examples of their use, focusing on the works of Papazoglou (2007), Lankhorst (2012), Kohlborn et al. (2009), Erl (2007) and Gu & Lago (2007). We have also reviewed Agile Methods literature based on the studies published by Strode (2006) and Abrahamsson (2002).

We have started the method conception, and all results accomplished were presented in the paper "Service Agile Development Using XP" published at the 7th International Symposium on Service Oriented System Engineering (SOSE).

The next steps are planned in the following roadmap:

- Conduct a deeper literature review, looking for an existing method that combines SOA principles and XP practices. The research performed so far has not found any method that fits the requirements.

- Afterwards, we plan to assess the methods found in regards of combining SOA principles and XP practices looking for a concrete method that can be used in real projects, i.e., the method has to provide well-defined practices that can be easily followed in professional scenarios.

This assessment may conclude that there is no method. In this case, a new method will be developed in this work. On the other hand, the assessment can result in an existing method that cannot be followed in practice. In this case, this work will evolve this method in order to accomplish such requirement.
• Once a properly method is defined, SOA specialists and Agile specialists will assess it, and a proposal of initial refinements will be produced.

• Afterwards, the method will be used in a real project at Petrobras IT department. This project will be conducted using 3-weeks sprints. At the end of each sprint, a review meeting will take place in order to the team evaluates the method and proposes further refinements. All proposed modifications will be logged for further analysis.

• Finally, the method will be described in the dissertation, making explicit the method and the rationale used in its development and its evaluation in the real scenario.

6. Related work

The literature review executed so far highlights researches on the combination of Agile Methods and SOA which are presented as follows. It is important to emphasize that a more systematic review is going to be executed.

Karsten & Cannizzo (2007) report their experience using XP and Scrum in a distributed team developing a Software Development Kit (SDK) and a set of services, adherent to SOA concepts to expose traditional telecom features. The authors provided important information about the usage of XP practices in an SOA development initiative, and also address one of the main challenges of SOA: distributed teams. Nevertheless, it does not mention how or whether SOA principles were pursuit by the team. There is no guideline or step-by-step proposal on how to combine agile and SOA. It also lacks information about how business-IT alignment was accomplished, considering the absence of a business stakeholder to provide guidance and feedback.

Ivanyukovich et al. (2005) analyze how XP practices can be used in service development within an SOA context. They believe that not all practices are relevant for service development, although considering advantages in the usage of other practices. They provide useful information on how to map SOA concepts to XP concepts, as well as adapting XP practices to work with service development. Nevertheless, such work does not address SOA challenges and complexities, and does not specify how XP practices could be used during service development.

Roy & Debnath (2010) aim to show how to develop services using XP practices in an SOA context. They depict a scenario of services development in a “traditional” team, where each member is responsible for its part and not communicating or collaborating properly with others. They suggest the usage of XP practices to turn this team into an agile one, but provide a shallow analysis of such conversion and applicability of the tools. Likewise, they do not refer to SOA complexities or how to address these complexities using XP practices.

Schatz (2006) considers the combination of agile methods to SOA as the next stage of evolution to deliver performing results to business. He focuses on the advantages brought by XP practices to SOA-compliant systems. Nevertheless, the work just provides insights on how SOA could take advantage of XP practices to deploy system faster and with more quality. It does not discuss SOA complexities and their relationship with XP practices.
Callahan (2006) aims to assess the applicability of XP practices to SOA development, considering the relationship between those practices and SOA principles and complexities. He considers the use of small releases, on-site customer and pair programming practices to be beneficial to SOA. However, the lack of an up-front design and minimal documentation are considered to be harmful to SOA applications. Although highlighting that XP core practices need adaptations to fit SOA complexities properly, he does not provide an analysis of what modifications should be made to XP core practices to combine them with SOA principles and complexities.

By analyzing these related works, we have identified that none of the authors have defined clear steps for implementing their proposals in real scenarios. There are many insights and considerations about how applicable is the combination of SOA and Agile principles, but there is not a clear definition of steps or practices to be followed by organizations aiming to implement such combination. None of the authors have either provided guidelines about how to use XP practices to address SOA complexities. For example, SOA teams are often geographically distributed, whereas XP emphasizes collocated teams. Besides, Pair Programming is a core practice, but in SOA usually teams are scattered. This is the kind of gap we are trying to fulfill with our proposal.

7. Conclusion
The work has already produced good results: definition of concepts, clarification where agile methods work in a development lifecycle, identification of gaps, development of an initial approach to combine SOA and XP. All results are described in the paper "Service Agile Development Using XP" published at SOSE 2013. The agile practices are listed and cross-referenced to agile methods in the document <http://www.uniriotec.br/~azevedo/UNIRIO/Agile_Methods_Practices_Detection.pdf>, resulting in an analysis of which agile practices are common among methods, and which method is the most comprehensive.

However, there are many tasks yet to be accomplished. For instance, a systematic review is required to confirm if a method that combines SOA and XP really does not exist. The result of this review will be used to define the direction of the work: to evolve an existing method or to create a new one. The evaluation of existing methods has to be presented in more systematic way, making explicit the assessment criteria.

Some future work we have already identified are: the application of XP practices to other phases of an SOA development lifecycle, instead of the construction phase; and the evaluation of the usage of other agile methods combine with SOA.

References
Information Systems: Service-Oriented Architecture and Software Engineering, Helsinki University of Technology, Finland.


