

Evolução de Requisitos na Metodologia Ágil

Thiago Cabral^{1,*}, Rafael Soares¹, Fernanda Alencar^{1,2}

¹ Programa de Pós Graduação em Engenharia da Computação, Universidade de Pernambuco,
Rua Benfica, 455 – Madalena – Recife/PE, Brasil

{tclm, rhas, fernandaalenc}@ecomp.poli.br

² Universidade Federal de Pernambuco, DES-CTG, Av. de Arquitetura, s/n., CDU - Recife/PE,
Brasil.

fernanda.ralencar@ufpe.br

Abstract. Nowadays, agile methodologies are widely accepted mainly by industry, by being iterative approaches that keep the dynamics of software development environments. Numerous researches indicate that these methodologies are useful and effective for successful development. However, very little has been about efforts made in the requirements phase in agile methodologies. The evolution of requirements, in particular, brings risks to the software project. However, when well managed can improve software quality and satisfaction of the stakeholders. The evolution of requirements is a themed deserving further examination of these methodologies. In the engineering community requirements, the i* approach is a very rich ontology and has been intensively studied and utilized. Thus, the paper proposes the use of the i* approach to the analysis of the factors necessary for the management of the evolution of requirements in agile methodologies.

Resumo. Atualmente, as metodologias ágeis são amplamente aceitas, principalmente pela indústria, por serem abordagens iterativas que mantêm a dinâmica dos ambientes de desenvolvimento de software. Inúmeras pesquisas apontam que essas metodologias são úteis e eficazes para o sucesso do desenvolvimento. No entanto, muito pouco se tem sobre esforços empreendidos na fase de requisitos nas metodologias ágeis. A evolução de requisitos, em particular, traz riscos ao projeto de software, mas quando bem gerenciada pode melhorar a qualidade do software e a satisfação dos stakeholders. A evolução de requisitos é uma temática que merece ser aprofundada nessas metodologias. Na comunidade de engenharia de requisitos, a abordagem i* é uma ontologia muito rica e tem se mostrado bastante estudada e utilizada. Assim, o artigo propõe o uso da abordagem i* para a análise dos fatores necessários ao gerenciamento da evolução dos requisitos em metodologias ágeis.

1 Introdução

A engenharia de requisitos é o processo de analisar e documentar a funcionalidade pretendida de um sistema de software. Muitas vezes considerada como uma atividade inicial na construção de um sistema [1], acaba por ter amplo ciclo de vida. Dentre as

atividades do processo de engenharia de requisitos merece destaque a evolução dos requisitos ao longo do processo de desenvolvimento do sistema. A maioria das empresas que utilizam os métodos tradicionais de engenharia de requisitos, no tocante à evolução dos requisitos, não sabe lidar com essa questão. Essas empresas se utilizam de estratégias *ad-hoc* para tratar essa questão.

Em se tratando de pequenos projetos de sistemas, a evolução dos requisitos pode não levar a grandes problemas, uma vez que uma pessoa pode coordenar o sistema em sua totalidade. Porém, em grandes projetos, o mau gerenciamento da evolução de requisitos leva a problemas indesejáveis, tais como, retrabalho, atraso para entrega do sistema, baixa qualidade e aumento dos custos do sistema de software. Em [6] é apresentado o estado da arte da evolução de requisitos. Segundo essa pesquisa existem muitos sinônimos para o termo “evolução” usados nas literaturas dentre os cerca de 120 trabalhos identificados com relação à temática.

Outros pontos relacionados a mudanças nos requisitos dizem respeito à volatilidade dos mesmos e às alterações no ambiente organizacional. A volatilidade dos requisitos leva a que funcionalidades descritas na especificação do sistema sejam também instáveis, dificultando a evolução natural do sistema pretendido. As mudanças no ambiente organizacional levam, não só, a problemas com a documentação, mas também com o gerenciamento e a evolução do sistema como um todo. Fato é que as mudanças serão constantes e os stakeholders devem estar preparados para isso [2].

Os métodos ágeis surgem com o objetivo de melhorar a comunicação entre os indivíduos no ambiente de desenvolvimento (comunicação face a face) e de minimizar as falhas no gerenciamento das mudanças. Assim, quatro valores fundamentais foram propostos: os indivíduos e suas interações devem ser considerados acima de procedimentos e ferramentas; o funcionamento do software deve estar acima de documentação abrangente; a colaboração dos clientes deve estar acima da negociação de contratos; e, a capacidade de resposta imediata às mudanças em detrimento de um plano pré-estabelecido. Não se trata de um desprezo aos métodos e ferramentas tradicionais do desenvolvimento de software, mas sim do estabelecimento de uma escala de valores, na qual a flexibilidade e a colaboração são mais relevantes do que a rigidez de processos e planejamentos clássicos.

O desenvolvimento ágil quebrou o paradigma de que não se pode planejar o desenvolvimento de um software como se planeja a construção de um prédio. Nos métodos ágeis os requisitos são desenvolvidos de forma incremental, de acordo com as prioridades do cliente. Com isso, como o setor de projetos de desenvolvimento de software está sujeito a constantes mudanças. Os métodos ágeis objetivam atender às rápidas mudanças[3].

Práticas de software ágeis obtêm uma maior agilidade através da auto-organização de equipes, colaboração do cliente, maior qualidade da documentação, menos tempo reduzido para o mercado. Suportam iterações que podem integrar qualquer mudança nos requisitos do usuário, durante a fase de implementação [4]. Todavia, após a entrega do produto o trato da evolução dos requisitos não é algo trivial. Faz-se necessário monitoramento de todo o conjunto de requisitos que permanece consistente, bem como a relação entre os requisitos, com as decisões tomadas junto à equipe de teste. Nos métodos ágeis, a evolução dos requisitos começa na fase de monitoramento, após a entrega do produto.

Apesar de tudo, alguns pesquisadores afirmam que os métodos ágeis se aplicam melhor a contextos específicos [2]. Alguns trabalhos apontam as diversas fragilidades desses métodos [5], sobretudo no tocante a engenharia de requisitos. Dentre os problemas apontados, destacamos a gerência de requisitos e a evolução desses.

Na Seção 2, são apresentados os objetivos da pesquisa. Na Seção 3, são descritas as contribuições científicas deste artigo. As conclusões são apresentadas na Seção 4. Os trabalhos futuros são apresentados na Seção 5 e por último, na Seção 6, as referências.

2 Objetivos da Pesquisa

A evolução de requisitos continua sendo um grande problema quando nos referirmos a uma metodologia ágil. Ainda não se sabe qual o impacto da evolução de requisitos é na prática nos métodos ágeis. Procuramos entender melhor as abordagens atuais utilizadas no gerenciamento de requisitos e os problemas que surgem na evolução de requisitos em uma metodologia ágil.

De acordo com Jaqueira et al.[5], Espinoza e Garbajosa [9] a diferença entre o desenvolvimento de software no modelo ágil e tradicional não se encontra apenas na forma como os processos são realizados, mas também nos artefatos produzidos como saídas de cada processo.

Para tentar solucionar a fragilidade dos métodos ágeis em tratarem a gerência de requisitos e da evolução desses, faz-se necessário identificar mecanismos de como tratar da evolução dos requisitos nas metodologias ágeis sem ferir, fundamentalmente, seus princípios.

Desta forma, este artigo propõe a utilização da técnica i* [8], para tratar a análise do processo da evolução dos requisitos em metodologias ágeis, tais como, o Scrum.

3 Contribuições Científicas

Evolução de requisitos é um processo de mudança contínua de requisitos em uma determinada direção. No entanto, pesquisas mais atuais se concentram em como lidar com a evolução depois que acontece. Evolução de requisitos acontece depois que um produto é entregue [6]. Este processo continua a ser um fenômeno pouco compreendido a partir das perspectivas de evolução.

No gerenciamento dos requisitos, o acompanhamento da evolução é necessário com o objetivo de se evitar problemas inesperados durante todo o ciclo de vida do sistema. O monitoramento de requisitos geralmente é realizado através da captura de informações específicas sobre os requisitos e é, especialmente, importante quando em sistemas de software de grande porte, com longo ciclo de vida.

Os sistemas de grande porte são muitas vezes baseados em produtos previamente lançados e muitos de seus componentes são reutilizados e /ou prolongados [1] fazendo com que a qualidade não fique em um nível desejado. Além disso, esses sistemas

estão sendo desenvolvidos por muitas equipes diferentes em diferentes localizações físicas o que deve levar a um maior cuidado não só em seu desenvolvimento, mas também em sua manutenção e na evolução de seus requisitos.

Esse assim chamado desenvolvimento global de software é introduzido por muitas empresas e essa tendência deverá crescer cada vez mais. Essa forma de desenvolver softwares com a terceirização globalizada faz com que o gerenciamento de requisitos se torne ainda mais difícil de organizar e implementar em um projeto [1]. Contudo, sabe-se que, ao gerenciar a evolução dos requisitos, evita-se que problemas futuros como custo elevado, falhas no gerenciamento de mudanças e baixa qualidade dos artefatos produzidos possam ocorrer no desenvolvimento. Nos métodos ágeis, a gerência de requisitos e a evolução não são tratadas como parte do desenvolvimento de software. Isso representa uma fragilidade desse tipo de metodologia.

Com isso, em foco, a utilização da técnica *i** pode proporcionar um mecanismo de visualização dos impactos na evolução do sistema pretendido. Metas podem deixar de ser satisfeitas, impactando diretamente nas mudanças dos requisitos. Da mesma forma que metas flexíveis, que representam requisitos não funcionais, podem também serem prejudicadas e impactarem na qualidade do sistema que esteja sendo evoluído.

A técnica *i** é uma ontologia rica que trata das relações entre atores e que vem sendo bastante utilizada nos dias atuais [8] [12]. Salienta-se ainda que a técnica *i** captura requisitos organizacionais e metas modelando sistemas e seus ambientes em termos dos relacionamentos intencionais entre atores estratégicos [10]. Da mesma forma, procura-se saber o que o ator quer, como ele consegue o que quer e de quem depende para conseguir o que quer. Contudo, esta técnica tem por objetivo articular a noção de intencionalidade distribuída. O foco nos stakeholders e seus relacionamentos para descrição de requisitos é uma característica da técnica *i**, onde os atores dependem uns dos outros para atingir suas metas. Os métodos ágeis também se concentram em fatores humanos e apostam na entrega de valor para o cliente. No grupo outras experiências com Scrum e *i** já tiveram êxito [5], [11].

A partir da avaliação do uso da técnica *i** em conjunto com métodos ágeis, em particular com o Scrum, através de casos exemplos será possível: (i) verificar como o uso dos modelos *i** enriquecem a análise da evolução dos requisitos em métodos ágeis, contribuindo para a melhoria dos projetos de desenvolvimento ágil de software e (ii) avaliar a viabilidade de seu uso como forma de documentação da evolução dos requisitos.

4 Conclusões

A gerência de requisitos engloba todas as partes que contribuem para a geração de um documento de requisitos e sua manutenção ao longo do tempo. A evolução é considerada uma das questões mais críticas no desenvolvimento de sistemas.

Por outro lado, têm-se os métodos ágeis inseridos em ambientes de desenvolvimento de software dinâmicos, onde os requisitos estão em constante modificação e são construídos a partir do *feedback* das partes interessadas (stakeholders) no decorrer do desenvolvimento. O não gerenciamento dos requisitos e o trato com sua evolução

afeta não só a confiabilidade do sistema, mas também as atividades de engenharia e do ambiente.

Em particular, a técnica ágil Scrum apresenta dificuldades não só em capturar requisitos, mas, sobretudo em gerenciá-los e, portanto, na evolução dos sistemas. Em se tratando de requisitos organizacionais de dependência e intencionalidade, que promovem o reuso e agilidade, poucas são as técnicas que os consideram. Em Scrum a dependência não é explicitamente considerada.

A técnica i^* por sua vez, é utilizada para o mapeamento organizacional, sendo amplamente pesquisada na academia. Possui uma série de vantagens, dentre elas a derivação sistemática de requisitos a partir de objetivos, o alcance da completude das especificações de requisitos, além de terem os objetivos mais estáveis do que os requisitos. Sua notação gráfica propicia a visibilidade dos relacionamentos de dependência e intencionalidade entre entidades que se relacionam (atores). Essa característica pode vir a contemplar às técnicas ágeis de recursos gráficos, pois o foco, tanto na técnica i^* e quanto nos métodos ágeis, está em todos os agentes envolvidos no processo. Assim, estamos investigando como usar o modelo i^* , para tratar a evolução dos requisitos nos métodos ágeis.

5 Trabalhos Futuros e em Andamento

Apesar do reconhecido papel da engenharia de requisitos, a evolução de requisitos continua sendo um fenômeno pouco compreendido [7]. Dessa forma, abordagens que tratam dessa questão são sempre importantes, pois da evolução depende o sucesso de um projeto.

Atualmente, está sendo realizado um trabalho de revisão sistemática acerca das principais fontes de discussão sobre como a evolução de requisitos está sendo abordada pela comunidade acadêmica e na indústria. O foco será com relação, aos métodos ágeis, em particular o Scrum, e aos desafios associados a esses. Será analisado se as relações de dependências dos atores da organização podem promover o reuso do conhecimento da organização, para os times, em vários projetos. Para tanto, será considerada a inclusão de conceitos avançados da técnica i^* , tais como papéis, agentes e posições. Outro ponto a ser investigado diz respeito ao uso do mapeamento organizacional muito além dos projetos de desenvolvimento e manutenção de software, mas com relação à análise de riscos desses projetos. Precisar-se-á avaliar, contudo, a compatibilidade com a característica de agilidade do Scrum.

Esta proposta surgiu para proporcionar aos stakeholders e toda equipe que faz parte do desenvolvimento a importância da evolução dos requisitos nos dias de hoje e um mecanismo gráfico de análise da evolução dos requisitos. Para validar a proposta, serão conduzidos exemplos com a participação da academia e da indústria.

6 Referências

1. Lormans, M.: Managing Requirements Evolution using Reconstructed Traceability and Requirements Views. Department of Software Technology. Delft University of Technology, (2009).
2. Puntel, M. and Prass, F.: Um Método Ágil Híbrido. Sistemas de Informação. Universidade Luterana do Brasil, Cachoeira do Sul, (2010).
3. VIANA, Antônio Geraldo Gonçalves.: Gerenciamento de projetos em processo ágil de desenvolvimento de software. Instituto de Educação Tecnológica – IETEC, Belo Horizonte.[20-] Disponível em: <www.techoje.com.br/site/techoje/categoria/abrirPDF/393>, Acesso em: abr. 2013.
4. Ahmed, A., Ahmad, S., Ehsan, Dr. N., Mirza, E. and Sarwar, S. Z.: Agile Software Development: Impact on Productivity and Quality, *Proceedings of the 2010 IEEE ICMIT*, Singapore, 2-5 June (2010), pp. 287-291.
5. Jaqueira, A., Andreotti, E., Lucena, M., Aranha E.: Desafios de Requisitos em Métodos Ágeis: Uma Revisão Sistemática. 3rd Brazilian Workshop on Agile Methods, São Paulo (2012).
6. Li, J., Zhang, H., Zhu, L., Jeffery, R., Wang, Q. and Li, M.: Preliminary Results of A Systematic Review on Requirements Evolution, In: 16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012), Institute of Software, Chinese Academy of Sciences, China, (2012), pp. 12–21
7. Anderson, S., Felici, M.: Quantitative aspects of requirements evolution, In: Computer Software and Applications Conference, 2002. COMPSAC 2002. Proceedings. 26th Annual International, (2002), pp. 27- 32.
8. Yu, E.: Modeling Strategic Relationships for Process Reengineering. In: Social Modeling for Requirements Engineering. Cambridge, Massachusetts, London, England: MIT Press, (2011) p. 11-152.
9. Espinoza, A. and Garbajosa, J.: Study to Support Agile Methods More Effectively through Traceability, Computer Science Innovations in Systems and Software Engineering, Vol. 7, No. 1, (2011), pp. 53-69.
10. Alencar, Fernanda M. R. de: Mapeando a Modelagem Organizacional em Especificações Precisas. 302 fls. Tese (Doutorado em Ciência da Computação). Universidade Federal de Pernambuco, Recife, Centro de Informática, Programa de Pós-Graduação em Ciência da Computação, (1999).
11. Scheidegger, Ma. E. S. de: Integrando Scrum e a Modelagem de Requisitos Orientada a Objetivos por meio do SCRUM i*. Dissertação (em Ciência da Computação). Universidade Federal de Pernambuco, Recife, Centro de Informática, Programa de Pós-Graduação em Ciência da Computação, (2011).
12. i*Wiki “IstarWiki community”. Disponível em: <<http://istar.rwth-aachen.de/tiki-index.php?page=i%2A+Wiki+Home>>. Acesso em: abr. 2013.