

Integrando Requisitos Ágeis com Modelos i*

Aline Jaqueira, Bernardo Gurgel, Márcia Lucena

Departamento de Informática e Matemática Aplicada – UFRN
alinejaqueira@ppgsc.ufrn.br, bernardogfilho@gmail.com,
marciaj@dimap.ufrn.br

Resumo. Histórias de usuário são um artefato de uso limitado. O contexto geral do sistema, seus objetivos, as relações de dependência dos requisitos são muitas vezes omitidos ou mal entendidos com o uso somente desse artefato. Este trabalho apresenta uma abordagem para complementar os requisitos ágeis (histórias de usuário) através de sua integração com modelos i*. As histórias de usuário são mapeadas para modelos i* e, por meio da visualização dos modelos, busca-se contribuir dando suporte aos *stakeholders* através da representação visual e mais abrangente dos requisitos. Além disso, as histórias de usuário poderão ser analisadas de acordo com a carga semântica da técnica i*. Ao utilizar os modelos, a característica de agilidade dos projetos poderá ser mantida pelo fato da visualização através de modelos ser mais simples e rápida.

1 Introdução

Nos métodos ágeis os requisitos são desenvolvidos de forma incremental, de acordo com as prioridades do cliente. A elicitação é realizada com clientes que fazem parte da equipe de desenvolvimento. Os clientes escrevem histórias do que o sistema precisa fazer (história de usuário) e priorizam as mesmas de acordo com o valor do seu negócio. Essas histórias são os artefatos de saída da atividade de elicitação dos requisitos nos métodos ágeis.

Uma história de usuário descreve a funcionalidade que tem valor para o cliente do software e é usada para planejamento do projeto, funcionando como um lembrete para a equipe já que conversas posteriores sobre a história são essenciais para transmitir os detalhes das mesmas [3]. Nos métodos ágeis as histórias de usuário são usadas por serem consideradas uma abordagem flexível, de baixa sobrecarga e centrada no usuário [1]. São amplamente utilizadas pelo desenvolvimento ágil [2] e são consideradas neste trabalho como um artefato de requisito ágil.

As histórias de usuário geralmente são escritas em linguagem natural. Um formato para a escrita das histórias vem sendo bastante utilizado: “como <papel>, eu quero <ação> para <meta>” [3]. Portanto, para aplicação das histórias de usuário neste trabalho, parte-se do princípio que as mesmas utilizam esse formato.

As histórias de usuário constituem artefatos muito restritos para contemplar determinadas questões como o entendimento global do sistema e as relações de dependências entre as funcionalidades [4]. Algumas limitações desse artefato [5]: (i) dependências entre as histórias são geralmente “escondidas”; (ii) o entendimento do contex-

to social no qual o sistema a ser desenvolvido está inserido não está, nem pode ser contido numa história de usuário; (iii) interpretar alguns detalhes e implicações do sistema torna-se um processo difícil a partir somente das histórias de usuário. Essas limitações sobrecarregam a equipe ao exigirem rigorosas operações mentais.

Além disso, contar com a participação do cliente comunicando os requisitos através das histórias, das conversas e dos testes de aceitação é um desafio e algumas decisões se tornam arriscadas principalmente para sistemas complexos. Para a equipe, torna-se muito difícil controlar o sistema unicamente com histórias de usuário [6].

2 Objetivos da Pesquisa

Este trabalho propõe uma abordagem que visa melhorar alguns desafios citados para as histórias de usuário. Busca-se dar suporte aos *stakeholders* fornecendo uma visão gráfica e abrangente das histórias de usuário e dos seus relacionamentos com o sistema através de modelos *i**.

Tanto os métodos ágeis quanto a técnica *i** ressaltam os *stakeholders* no ambiente de desenvolvimento, o que justifica a escolha da técnica *i** para representar as histórias de usuário [13]. 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 reconhecem as pessoas como peças fundamentais e como principais impulsionadores do sucesso do projeto [7].

Portanto, neste trabalho, os conceitos das histórias de usuário são empregados juntamente com os conceitos da técnica *i** [8]. O papel das histórias de usuário são mapeados como atores nos modelos *i**, as metas das histórias de usuário são mapeadas como metas nos modelos *i** e as ações das histórias de usuário são mapeadas como tarefas do Ator Sistema, pois as mesmas serão operacionalizadas pelo sistema. Foram utilizados conceitos e notações da técnica *i** de sua versão simplificada [9] e somente alguns elementos da técnica *i** são utilizados de acordo com a necessidade de mapeamento deste trabalho. Dessa forma, na técnica *i**, para construir o modelo SD (*Strategic Dependency*), os elementos utilizados são o *ator*, as *metas* e a *associação IS_A*. Para o modelo SR (*Strategic Rationale*) são utilizados os elementos *tarefas*, *recursos* e a ligação de *decomposição*.

Ao utilizar os modelos da técnica *i** a partir das histórias de usuário, a característica de agilidade dos projetos poderá ser mantida pelo fato da visualização através dos modelos ser mais simples e rápida. Modelos proporcionam agrupamentos visuais que permitem analisar rapidamente grandes quantidades de informações, além de ajudarem a organizar e apresentar as informações e darem contexto aos detalhes [6]. Uma vez que os modelos *i** são representações gráficas dos requisitos, ter-se-á uma maneira abrangente e visual de enxergar as histórias de usuário, melhorando a visualização do contexto do sistema, facilitando o acesso aos requisitos, contribuindo para a tomada de decisão no ambiente de desenvolvimento.

Para simplificar o entendimento, foram estabelecidas heurísticas como recurso para se chegar à solução do mapeamento [13]. Vale ressaltar que as heurísticas esta-

belecidas devem ser executadas na ordem em que foram expressas para que o mapeamento ocorra de maneira mais correta e objetiva.

Heurísticas para mapear as histórias de usuário para o modelo SD:

SD-H1: Criar o Ator Sistema;

SD-H2: Criar um Ator no modelo i^* para cada diferente papel das histórias de usuário;

SD-H3: Criar uma meta no modelo i^* para cada meta das histórias de usuário. Se houver metas repetidas as mesmas serão definidas uma única vez no modelo;

SD-H4: Se houver metas repetidas para atores diferentes, criar um Ator genérico;

SD-H4.1: Criar um relacionamento IS_A do Ator genérico para os demais atores específicos que compartilham a mesma meta;

SD-H5: Relacionar as dependências de cada ator com suas metas.

Heurísticas para mapear as histórias de usuário para o modelo SR:

SR-H1: Criar uma Tarefa dentro do Ator Sistema para cada ação das histórias de usuário;

SR-H2: Se houver ações diferentes para a mesma meta, criar uma tarefa genérica;

SR-H2.1: Decompor a tarefa genérica em sub tarefas que representem as ações associadas à mesma meta;

SR-H3: Relacionar as dependências de cada meta com as tarefas correspondentes de acordo com as histórias de usuário.

SR-H4: Se houver Tarefas que dependem do próprio Ator a que estão relacionadas, gerar um recurso com o nome da tarefa;

SR-H5: Relacionar o recurso criado dependendo do Ator.

Para demonstrar uma aplicação da abordagem, utilizou-se como exemplo um sistema de *login* simples [10] considerando a perspectiva de um usuário e de um administrador. A tabela 1 apresenta as histórias de usuário do sistema de *login*.

De acordo com as heurísticas propostas, o modelo SD será mapeado primeiramente criando o Ator Sistema. Posteriormente, cria-se os atores para os papéis das histórias de usuário, que, nesse caso, são ator Usuário e ator Administrador. As metas das histórias de usuário são criadas como metas no modelo SD e ligadas como dependências saindo dos atores a que estão associadas e chegando ao Ator Sistema.

Table 1. Histórias de Usuário do Sistema de Login (Fonte: IBM [10])

	Papel	Ação	Meta
1	Usuário	Ter nome de usuário	Acessar conteúdo seguro
2	Usuário	Ter senha	Acessar conteúdo seguro
3	Usuário	Escolher nome de usuário	Personalizar a conta
4	Usuário	Alterar senha padrão	Personalizar senha
5	Administrador	Atribuir senha ao usuário	Registro ser automatizado
6	Administrador	Enviar email de registro	Confirmar ativação da conta no email
7	Administrador	Solicitar login ao usuário	Garantir segurança do

			conteúdo
8	Usuário	Cadastrar lembrete de senha	Lembrar a senha
9	Administrador	Solicitar lembrete de senha	Confirmar usuário

Para gerar o modelo SR, cada ação das histórias de usuário foi gerada como tarefa dentro do Ator Sistema, pois o mesmo é que irá operacionalizá-la, realizando a tarefa de maneira particular para atender às metas dos atores. Como existem, nesse exemplo, ações diferentes para a mesma meta, foi criada uma tarefa genérica no modelo SR que foi decomposta nas ações em forma de sub-tarefas. As tarefas que dependem do próprio ator geraram um recurso que depende do ator com o nome da tarefa. A figura 1 mostra o resultado do mapeamento.

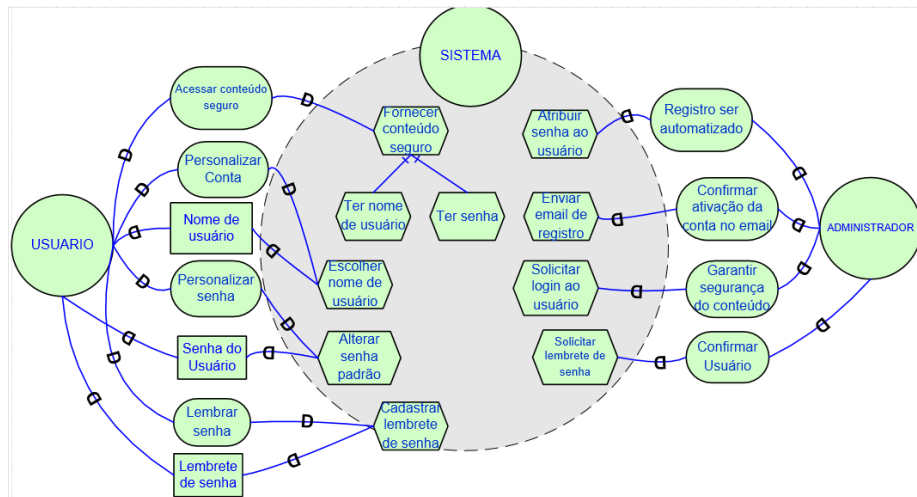


Fig. 1. Modelo SR do Sistema de Login

3 Contribuição Científica

Mesmo em um ambiente ágil é necessário desenvolver algum modelo antes da implementação para garantir uma compreensão compartilhada pela equipe, de modo que fique sincronizada com os objetivos do negócio, valor e contexto do projeto [6]. Modelos visuais auxiliam nesse entendimento e no entendimento de como os usuários precisarão usar o sistema. São eficazes para os *stakeholders* entenderem a solução proposta e também para mantê-los interessados e envolvidos e até mesmo para deixar claro o que a solução não vai entregar.

A contribuição deste trabalho é a elaboração de uma abordagem que utiliza modelos visuais propiciados pela técnica *i** para amenizar algumas limitações das histórias de usuário. Além disso, essa abordagem poderá funcionar como uma forma de documentação visual dos requisitos do software, contribuindo assim para amenizar também o desafio da falta de documentação no ambiente de desenvolvimento ágil [12].

Outras contribuições podem ser citadas como a melhoria no entendimento do contexto do sistema a ser desenvolvido, o uso e o acesso mais fácil à informação das histórias de usuário a partir da visualização do modelo visual, melhoria do processo de tomada de decisão de acordo com a análise das histórias, uma vez que elas estão descritas em modelos i^* .

Além disso, para aplicar a abordagem deste trabalho, está em desenvolvimento a ferramenta *StoryTeller*, cujo objetivo é fornecer um meio para armazenar e gerenciar as histórias de usuário dos projetos de software. A ferramenta está sendo desenvolvida como um projeto *open source* de aplicativo *web*. O *link* para o repositório no *Github* é <https://github.com/bernardogfilho/storyteller> e o *link* para onde a ferramenta se encontra hospedada no *Heroku* (<http://storyteller-beta.herokuapp.com/>).

Na *StoryTeller*, os usuários poderão se registrar e criar projetos vinculados à sua conta além de cadastrar as histórias de usuário dos projetos. Assim, será possível realizar operações básicas como editar ou excluir as histórias de um projeto. Será possível também pesquisar por histórias que possuam o mesmo ator ou a mesma meta. Além de fornecer uma base para organizar e gerenciar as histórias de usuário, a ferramenta será estendida para automatizar a abordagem deste trabalho, ou seja, mapear as histórias de usuário para os modelos i^* , a fim de fornecer o modelo visual de maneira automatizada. Pelo fato de a ferramenta tratar as histórias de usuário de modo independente e flexível, as mesmas se tornam fáceis de manipular, sendo possível exportá-las de diversas maneiras e relações através de um arquivo *.xml* ou *.json*, por exemplo.

4 Conclusão

Uma abordagem baseada em modelos visuais pode fornecer ligações mais diretas e rastreáveis para o desenvolvimento do sistema, promovendo uma análise com maior impacto na concepção e implementação do software. Funciona também para facilitar a comunicação, compreensão, a detecção de problemas ou explorar cenários hipotéticos e potenciais soluções. Ao visualizar modelos, possíveis erros e/ou negligências são reconhecidos mais facilmente [11]. Tudo isso facilita o processo de análise e discussão a respeito do sistema a ser desenvolvido.

A partir do mapeamento das histórias de usuário para os modelos SD e SR da técnica i^* , é possível organizar e representar todas as histórias em um modelo que fornece uma visualização geral das histórias e seus relacionamentos. Além disso, todas as histórias de um mesmo ator são apresentadas no mesmo modelo, permitindo encontrá-las com maior facilidade. Dessa maneira, é possível entender melhor o contexto do sistema, seus principais atores e suas metas.

A visualização através dos modelos torna mais fácil a identificação de dependências de cada ator com o Sistema bem como a identificação de tarefas do Sistema para atender a cada ator específico envolvido com o software.

Portanto, a abordagem deste trabalho possibilita melhor análise, comunicação, discussão e melhor entendimento do sistema a ser desenvolvido.

5 Trabalhos Futuros

Com o objetivo de dar continuidade à pesquisa desenvolvida por este trabalho algumas sugestões de trabalhos futuros podem ser citadas: (i) o desenvolvimento de diretrizes ou uma ferramenta para realizar a transformação das histórias de usuário para o formato utilizado neste trabalho, de modo a poder utilizar esta abordagem para todas as histórias de usuário; (ii) o desenvolvimento de diretrizes para realizar o mapeamento “de volta”, dos modelos i* para as histórias de usuário; (iii) o tratamento da escalabilidade para o ator Sistema; (iv) a possibilidade de representar as tarefas nos demais atores do modelo; (v) a possibilidade de representar o relacionamento entre os demais atores no modelo.

Referencias

1. O’heocha, C., Conboy, K.: The Role of the User Story Agile Practice in Innovation. In: P. Abrahamsson, N. Oza (Eds.), *Lean Enterprise Software and Systems* (pp. 20-30). Springer, New York (2010)
2. Cockburn, A.: *Agile Software Development: The Cooperative Game*. Pearson, Boston (2007)
3. Cohn, M.: *Agile Estimating and Planning*. Prentice Hall, Massachusetts (2006)
4. Sharp, H., Robinson, H., Segal, J., Furniss, D.: The Role of Story Cards and the Wall in XP Teams: a Distributed Cognition Perspective. *Proceedings of Agile 2006*, IEEE Computer Society Press, pp. 65-75. Washington (2006)
5. Sharp, H., Robinson, H., Petre, M.: The Role of Physical Artefacts in Agile Software Development: Two Complementary Perspectives. *Interacting with Computers* 21 pp. 108–116. New York (2009)
6. Beatty, J., Chen, A.: *Visual Models for Software Requirements*. Microsoft Press, Washington (2012)
7. Highsmith, J., Cockburn, A.: *Agile Software Development: The Business of Innovation*. *IEEE Computer*, vol. 34, pp. 120-122 (2001)
8. Yu, E.: *Modelling Strategic Relationships for Process Reengineering*. PhD Thesis. University of Toronto, Department of Computer Science (1995)
9. i* Wiki Home, <http://istar.rwth-aachen.de/tiki-index.php>
10. IBM Data sets : Example of User Stories, <http://www-958.ibm.com/software/analytics/manyeyes/datasets/example-of-user-stories/versions/1>
11. Horkoff, J., Yu, E.: A Qualitative, Interactive Evaluation Procedure for Goal- and Agent-Oriented Models. In: *CAiSE Forum. CEUR Workshop Proceedings* (2009).
12. Jaqueira, A., Andreotti, E., Lucena, M., Aranha, E.: Desafios de Requisitos em Métodos Ágeis: Uma Revisão Sistemática. In: *3rd Brazilian Workshop on Agile Methods*, São Paulo (2012)
13. Jaqueira, A.: *Uso de Modelos i* para Enriquecer Requisitos em Métodos Ágeis*. Dissertação de Mestrado. Departamento de Informática e Matemática Aplicada – UFRN. Natal (2013)