

MÉTODO LRCF PARA LEVANTAMENTO DE REQUISITOS DE PROJETO BASEADO EM CARTÕES DE FUNCIONALIDADE

Letícia Pavesi Mansano, MSc.
José Marcelo Almeida Prado Cestari, PhD.

RESUMO

A tecnologia tem evoluído cada vez mais rápido, por isso o tempo que pode ser gasto no desenvolvimento de novos produtos está reduzindo. Isto exige adaptação contínua da metodologia de desenvolvimento utilizada. Contudo, os anseios do cliente em relação ao produto se mantêm. Estes desejos são refletidos na satisfação do cliente ao ter contato com o produto e impactam financeiramente no retorno do investimento do projeto. Esses fatores justificam a necessidade de uma metodologia que enumere um maior número de requisitos no início do projeto. Assim, problemas como estimativa imprecisa de tempo, recursos humanos mal dimensionados e mudanças de requisitos são minimizados. O método de Levantamento de Requisitos com Cartão de Funcionalidade tem como objetivo identificar os desejos do cliente, mapeando as prioridades. Além disso, o método propõe que clientes e desenvolvedores interajam, evitando a criação de requisitos conflitantes e interpretação incorreta de especificações. Esta metodologia utiliza Cartões de Funcionalidade, que são preenchidos pelo cliente. Após o preenchimento dos cartões, os mesmos são entregues aos desenvolvedores. O método foi aprovado pelas pessoas que trabalham em uma empresa brasileira de produtos eletrônicos.

Palavras-chave: requisitos, análise de requisitos, cartão de funcionalidade

ABSTRACT

The technology has evolved increasingly faster, so time spent developing a new product is reducing. This requires the evolution of the development methodology. However, customer's concerns and expectations about product is maintained. These desires are reflected in the customer's satisfaction, during contact with the product and it also impacts financially on project's return of investment. These factors justify the need for a methodology that enumerates the largest possible number of requirements at the beginning of the project. Thus, problems such as inaccurate time estimation, amount of human resources and impact of requirement changes are minimized. The use of the Requirement Gathering with Functionality Cards - LRCF method aims to identify the customer's wishes, by mapping the priorities. Furthermore, the method makes customers and developers interact, preventing the creation of conflicting requirements and specifications misinterpretation. This methodology uses Functionality Cards, which are client filled. Then cards are delivered to developers. The method was approved by Brazilian people who design electronic products.

Keywords: Requirements, Requirements Analysis, Functionality Cards

1. INTRODUÇÃO

O aumento da complexidade dos projetos fez com que o ser humano desenvolvesse metodologias para organizar e monitorar as atividades durante a concepção de um produto. Como é descrito por Haughey (2014), houveram consideráveis avanços no gerenciamento de projetos para que o homem fosse capaz de construir as Pirâmides do Egito, a Muralha da China. Em 1917 é apresentado o gráfico de Gantt, que é utilizado até os dias de hoje para controlar a evolução das atividades. Em meados dos anos cinquenta, a Dupont passou a utilizar o método do Caminho Crítico. Nesta época a marinha americana desenvolveu a técnica Program Evaluation Review Technique (PERT).

Haughey (2014) apresenta a WBS, conhecida como EAP - Estrutura Analítica de Projeto, desenvolvida nos anos sessenta pelo departamento de defesa americano. Em 1975 a empresa Simfact Systems Limited lançou o PROMPTII, visando projetos de software, seguida pelo Scrum em 1986. A década de noventa foi marcada pela publicação do método PRINCE pelo governo do Reino Unido e pelo Critical Chain de Eliyahu M. Goldratt. Nos anos 2000 foi elaborado o Manifesto Ágil e o PMBOK consolidou-se como referência na área de gerenciamento de projetos. Recentemente foi lançada a ISO 21500:2012 que padroniza o gerenciamento de projetos.

Além da melhoria do processo de desenvolvimento de novos produtos, houve uma mudança no comportamento dos consumidores. Com isso, a satisfação em consumir um produto levou a maior popularidade do fabricante e lealdade do consumidor. Para Beard (2014) a satisfação do cliente é uma definição criada pelo marketing para definir as expectativas dos clientes ao adquirirem um produto ou serviço. Este sentimento é importante para prever a possibilidade de uma nova compra e a lealdade do cliente. Este sentimento é utilizado como ponto de diferenciação e interfere na taxa de rotatividade de clientes do negócio. Ao entregar produtos que atendam as expectativas do cliente, o fornecedor aumenta a vontade do cliente de continuar adquirindo produtos da mesma procedência e reduz a propaganda negativa que é feita pelos consumidores insatisfeitos com a aquisição. Do ponto de vista da empresa, Beard (2014) também afirma que a manutenção de clientes é muito mais barata que a busca por novos.

Assim como é importante para o consumidor que suas expectativas sejam atendidas, para os fornecedores de um produto ou serviço é importante avaliar a rentabilidade. A sigla ROI, muito utilizada em gerenciamento de projetos, significa Return of Investment (em tradução livre Retorno do Investimento). Frequentemente, este retorno é apresentado em porcentagem. Este índice representa o benefício econômico de um determinado investimento, em relação ao montante de dinheiro investido.

O desenvolvimento ágil é focado no princípio econômico de que, ao entregar uma funcionalidade ao cliente mais rapidamente, antes será realizado o pagamento pelo cliente (Leffingwell, 2012).

Para Dunn (2008), os projetos desenvolvidos com a metodologia ágil têm a satisfação dos clientes maior, porque as funcionalidades mais importantes para o cliente são entregues em um tempo menor, em comparação com a metodologia em cascata de desenvolvimento.

2. CONTEXTUALIZAÇÃO

2.1 SCRUM

Dentre as metodologias de gerenciamento de projetos disponíveis, a metodologia ágil Scrum tem ficado cada vez mais evidente nos projetos. A história do Scrum inicia em 1986, com Takeuchi e Nonaka (1986). Nesta pesquisa foi apresentado como as empresas Honda, Canon e Fuji-Xerox produziram bons resultados desenvolvendo projetos com equipes autônomas, multidisciplinares e organizando as atividades do projeto de forma diferenciada. O termo Scrum é emprestado do rugby e faz alusão a forma como a equipe se organiza após uma infração no jogo ou a bola sai de campo.

Em 1993, Jeff Sutherland e sua equipe desenvolveram um processo de desenvolvimento de software que combinava os conceitos apresentados por Takeuchi e Nonaka e conceitos do desenvolvimento orientado a objetos. Com isso, foi criada uma forma iterativa e incremental de desenvolvimento de software, que permitia adaptar o desenvolvimento e aumentar a produtividade. Desde então, o Scrum vem sendo aprimorado por diversos colaboradores. O Scrum foi criado para o desenvolvimento de software, porém vem sendo utilizado com sucesso no desenvolvimento de hardware, programas de marketing e iniciativas de vendas (Rubin, 2012).

O desenvolvimento de um projeto utilizando Scrum pode ser visualizado na Figura 1. No início de um projeto é desenvolvida uma lista ordenada com todo o escopo do projeto. Esta lista é denominada Product Backlog e apresenta o conteúdo do produto expresso em forma de Funcionalidades. Uma forma de descrevê-las é preencher o modelo: “As a <type of user>, I want <some goal> so that <some reason>”.

Este modelo de descrição de funcionalidade é defendido por Cohn (2005), pois expressa o desejo do cliente em primeira pessoa e, durante a elaboração da narrativa, é imaginada a execução da ação. Este modelo também facilita a priorização do desenvolvimento.



Figura 1. Fluxo de desenvolvimento utilizando o Scrum. Disponível em <https://www.scrumalliance.org/why-scrum>

Schwaber e Sutherland (2013) apresentam um manual com os conceitos do Scrum. Segundo os autores, o desenvolvimento é dividido em janelas temporais que variam de duas a quatro semanas, denominados Sprints. Este processo é iterativo e

acontece até que todo o Product Backlog seja executado, ou seja, todo o escopo do projeto seja desenvolvido.

Fazem parte do desenvolvimento de um produto o Scrum Master, o Product Owner e o Development Team. O Scrum Master é o responsável por garantir que não existam impedimentos ao desenvolvimento do produto, treinar a equipe para que siga a metodologia e auxiliar tanto o Development Team quanto o Product Owner. A proximidade entre o cliente e o desenvolvimento do produto cria o papel do Product Owner. Esta pessoa representa todos os anseios do cliente em relação ao produto e deve estar sempre disponível para que dúvidas sobre o produto possam ser respondidas. A execução do trabalho é feita pelo Development Team, que é uma equipe de desenvolvedores de habilidades variadas. É sugerido que esta equipe seja composta por três a nove desenvolvedores.

Os Sprints têm uma reunião de planejamento, chamada de Sprint Planning. Nesta reunião são selecionados os itens do topo da lista do Product Backlog. O volume de trabalho que será executado é determinado pelo Development Team, baseado na dificuldade das atividades.

Depois de planejado o trabalho que será executado no Sprint, segue a execução das atividades. Diariamente, é feita uma reunião denominada Stand Up Meeting. O objetivo desta reunião é apresentar ao restante da equipe o progresso realizado pelo desenvolvedor, respondendo de forma rápida (em aproximadamente trinta segundos) o que foi desenvolvido desde a última reunião e o que será feito até a próxima. Devido ao tempo reduzido para expressão individual do desenvolvedor e as equipes reduzidas, esta reunião é realizada com os participantes de pé e em uma roda. O Scrum Master também participa desta reunião.

Ao término do Sprint, é realizada a reunião de Sprint Review. Neste momento é feita a apresentação do trabalho executado ao Product Owner e, é indicado que sejam entregues uma versão do produto parcial para que o Product Owner seja capaz de experimentar a funcionalidade.

O processo de desenvolvimento utilizado pela equipe também é avaliado a cada Sprint, na reunião denominada Sprint Retrospective. Neste momento o Development Team se reúne com o Scrum Master e as sugestões de melhoria são selecionadas para serem implementadas no Sprint seguinte.

A utilização de Scrum permite ao cliente alterar o escopo do projeto com mais flexibilidade, mesmo porque durante a confecção do produto o cliente pode ter contato com produtos concorrentes e identificar outras necessidades que o produto poderia vir a atender. Para Leffingwell (2011), este contato e a descoberta de novas necessidades leva o cliente a modificar alguns requisitos. Por isso, o autor salienta que os desenvolvedores do produto devem iniciar os trabalhos assumindo que os requisitos apresentados pelo cliente possuem erros e incertezas. Esta mudança de pensamento torna necessário o acompanhamento constante com o cliente, verificando suas necessidades e possíveis alterações de escopo. Este acompanhamento é favorecido com a utilização de metodologias ágeis de desenvolvimento.

2.2 FUNCIONALIDADES E REQUISITOS

A formalização dos anseios do cliente em um projeto é feita na forma de requisitos. Segundo Leffingwell (2011), é desafiante construir requisitos válidos, mesmo que se preste atenção nas solicitações do cliente. Muitas vezes, durante o levantamento de requisitos, é ignorado o fato de que o cliente não sabe o que

realmente deseja. Ainda, que o cliente não é capaz de se expressar de forma que os requisitos possam ser compreendidos.

Uma das formas para identificar os requisitos de um projeto é solicitar do cliente que sejam listadas as funcionalidades desejadas. Isto significa solicitar que seja descrito um comportamento ou função que o produto deve apresentar em determinado cenário ou situação.

Uma funcionalidade bem descrita permite que sejam extraídos os requisitos do produto. Sommerville (2011) afirma que os requisitos de um sistema não devem especificar apenas os serviços ou funcionalidades que o mesmo deve apresentar. Os requisitos devem listar as necessidades para que o sistema entregue a funcionalidade corretamente.

A partir da descrição de uma funcionalidade, a equipe de desenvolvimento do projeto deve ser capaz de extrair os Requisitos Funcionais e os Não-Funcionais do produto. Um Requisito Funcional é uma frase que contém um serviço que o produto deve apresentar, como o sistema deve reagir ao receber uma entrada, como o mesmo deve se comportar em situações particulares e o que não deve ser executado pelo sistema (Sommerville, 2011).

Requisitos Não-Funcionais são partes do sistema, que geralmente afetam o produto como um todo, não apenas uma funcionalidade. Este tipo de requisito tem como exemplo o controle temporal, restrições de funcionamento regulamentadas por padrões, restrições do processo de desenvolvimento.

Leffingwell (2011) afirma que, ao elaborar os requisitos de um sistema, devem ser consideradas necessidades técnicas, de usuário, dos operadores do sistema e viabilidade financeira, mesmo que o produto seja solicitado por um único cliente. Ao manter o foco apenas no cliente, podem ser descobertos requisitos não-funcionais.

2.3 CARTÃO DE FUNCIONALIDADE

A identificação dos requisitos de um produto utilizando o método proposto requer a preparação de Cartões de Funcionalidade. Estes cartões são preenchidos pelo cliente e pela equipe de desenvolvimento. Cada cartão deve apresentar espaço suficiente para a descrição de uma única funcionalidade, o espaço para um índice de importância, o volume de trabalho necessário à implementação desta funcionalidade e uma palavra-chave.

Como usuário eu gostaria que o PRODUTO...

Ordem: Tam.:

Palavra-Chave:

Figura 2. Cartão de funcionalidade proposto para utilização do método LRCF

O tamanho deste cartão varia de acordo com a quantidade de pessoas que irá descrever estas funcionalidades, pois todos os participantes das reuniões devem ser capazes de ler o conteúdo dos cartões.

A disposição das informações no cartão é feita de forma que o espaço destinado a descrição da funcionalidade é de aproximadamente dois terços do mesmo. Além disso, esta descrição inicia com o texto padrão: “Como usuário, eu gostaria que o [nome do projeto]...”. Esta descrição tem como objetivo conduzir o processo de formalização de requisitos com a perspectiva do cliente.

Na parte inferior do cartão, deve ser incluída a importância da funcionalidade, considerando todos os requisitos identificados. Este registro pode ser feito enumerando ordenadamente os cartões.

Outro campo importante do Cartão de Funcionalidade é o tamanho do requisito. Esta classificação tem granularidade baixa e serve apenas para informar aos envolvidos no projeto o volume de trabalho empregado. É sugerido utilizar elementos que desassocie o *Tamanho* e tempo de desenvolvimento. Por isso, no cartão exemplo da Figura 2 são utilizados cinco tamanhos: *PP*, *P*, *M*, *G* e *GG*. Estes são associados com tamanho de camisetas, mas podem ser utilizadas outras unidades de medida como baldes de tamanhos distintos (Cohn, 2005).

O último campo do Cartão de Funcionalidade é Palavra-Chave. Dependendo da quantidade de funcionalidades ou tipo de aplicação de um projeto, pode-se utilizar palavras-chave para auxiliar o agrupamento de funcionalidades semelhantes. O espaço de preenchimento deste campo deve ser pequeno, porque devem ser utilizadas no máximo duas palavras para referenciar a funcionalidade.

3. DESCRIÇÃO DO MÉTODO LRCF - LEVANTAMENTO DE REQUISITOS UTILIZANDO CARTÃO DE FUNCIONALIDADE

A obtenção dos requisitos de um projeto de pequeno e médio porte consiste em reuniões que promovem a interação entre os clientes e os envolvidos no processo de confecção do mesmo. A quantidade de reuniões varia de acordo com a complexidade do projeto e a quantidade de requisitos que precisam ser formalizados e discutidos. Em todas as reuniões de levantamento de requisitos, devem ser utilizados e manipulados os Cartões de Funcionalidade.

Inicialmente, o cliente é encorajado a preencher os cartões com as funcionalidades que o produto deveria possuir. Ao listar estas características, é utilizada a perspectiva do cliente e deve-se procurar obter a maior quantidade de detalhes sobre cada funcionalidade, ou seja, requisitos da funcionalidade.

Para não tornar o preenchimento dos cartões cansativo, é feito um processo iterativo de descoberta de requisitos. Neste processo, cada participante preenche um Cartão de Funcionalidade por iteração. Quando todos terminarem o preenchimento, a leitura aleatória das funcionalidades descobertas é realizada.

Quando o cliente considera terminado o processo de listagem das funcionalidades, deve ser iniciada a priorização. Isto demonstra como deve ser organizado o cronograma do produto, visando a maior satisfação do cliente. Ainda, é mostrado quais funcionalidades são *vitais* ao projeto, o que *deve* ser incluído no produto e as funcionalidades que o cliente *gostaria* que o produto possuísse.

A elaboração da lista de funcionalidades do produto ordenada pode ser obtida em uma ou várias reuniões com o cliente. Sendo que a quantidade exata de reuniões

vai ser definida pelo tempo gasto durante a interação dos envolvidos e pelo porte do projeto.

A lista de funcionalidades do produto deve ser apresentada aos desenvolvedores na sequência. Então, em uma reunião com os desenvolvedores e o cliente, é feita uma apresentação das funcionalidades e do motivo da ordenação. Ao compreender a lista com as funcionalidades priorizadas, a equipe de desenvolvimento se despede do cliente e a complementa incluindo os requisitos não-funcionais. Um exemplo de funcionalidade é apresentado na Figura 3. A funcionalidade apresentada foi quebrada em requisitos do produto e complementada com os requisitos não-funcionais intrínsecos e obrigatórios ao correto funcionamento da solicitação do cliente. Estes requisitos são apresentados na Tabela 1.

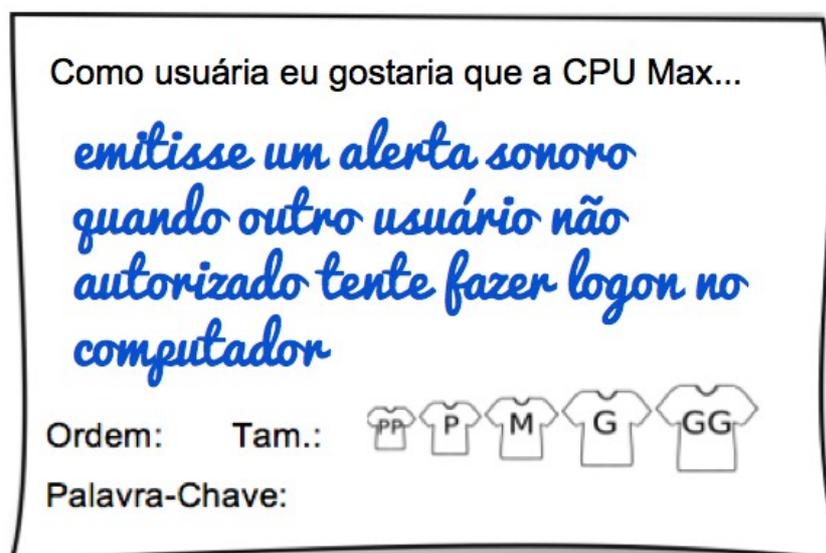


Figura 3. Exemplo de funcionalidade fictícia de um produto preenchida em um Cartão de Funcionalidade

Tabela 1. Requisitos funcionais e não-funcionais da funcionalidade descrita na Figura 3.

A CPU deve possuir circuito com buzzer para emissão de alerta sonoro
A CPU deve comportar mecanicamente a inclusão do circuito de buzzer
A CPU deve acomodar o buzzer próximo à superfície da carcaça para que o som não seja abafado
A CPU deve possuir um script para identificar tentativas de logon no computador
A CPU deve ser capaz de consultar os usuários cadastrados para poder identificar usuários não cadastrados
A CPU deve permitir que o sistema operacional controle o circuito de buzzer
A CPU deve possuir um circuito de buzzer compatível com a forma de alimentação da placa mãe

O próximo passo da metodologia é mensurar a dificuldade para a implementação da funcionalidade. A equipe de desenvolvimento deve escolher a funcionalidade mais fácil e a mais difícil, independentemente da prioridade. Estas funcionalidades são mensuradas com a menor e a maior unidade de medida adotadas, respectivamente. Utilizando o Cartão de Funcionalidade apresentado na Figura 2, as funcionalidades são rotuladas como PP e GG.

O último campo a ser preenchido nos Cartões de Funcionalidade é a palavra-chave, que resume a funcionalidade em uma ou duas palavras. Ao definir um rótulo curto para a funcionalidade, o diálogo entre a equipe é padronizado desde o início do projeto.

A forma como um projeto é gerida não interfere na utilização do método LRCF, pois sua utilização ocorre em um momento específico do ciclo de vida de um produto. A contribuição do método LRCF no levantamento de requisitos de um projeto de pequeno e médio porte é sumarizada na Figura 4.



Figura 4. Resumo do Método LRCF, que auxilia no levantamento de requisitos de um projeto utilizando Cartão de Funcionalidade.

4. ESTUDO DE CASO

O método proposto foi empregado no desenvolvimento de produtos compostos por hardware e software em uma empresa de pequeno porte, da cidade de Curitiba. A equipe de desenvolvedores é composta por dez pessoas, sendo a metodologia ágil Scrum empregada para a construção do produto.

No início de dois mil de dezessete, foram solicitados dois produtos de médio porte. Para a etapa inicial de levantamento de requisitos, o método foi empregado com o cliente, que passou a ser o Product Owner do projeto, e os desenvolvedores (Development Team).

Durante a escrita deste artigo, os dois projetos em que a metodologia foi empregada encontravam-se em aproximadamente cinquenta e oitenta por cento de desenvolvimento. Os resultados da utilização do método LRCF na empresa pesquisada foram obtidos ao longo das reuniões de feedback, realizadas quinzenalmente. Foi considerada a opinião do cliente e dos desenvolvedores.

Na opinião dos entrevistados a utilização do método produziu um maior número de requisitos para o produto e os cronogramas de desenvolvimento estão sendo seguidos de forma mais precisa, em comparação com os projetos desenvolvidos anteriormente.

Devido a singularidade e confidencialidade de cada produto, não é possível quantificar a melhoria na identificação de requisitos. Porém, erros de estimativa de data de entrega foram reduzidos. As alterações de escopo solicitadas pelo cliente impactaram no projeto de forma mais sutil, isto porque, na visão dos desenvolvedores, a modificação do produto foi menos traumática.

5. CONCLUSÃO

O método LRCF proposto neste artigo descreve como realizar o levantamento dos requisitos de um projeto utilizando Cartões de Funcionalidade. O LRCF foi desenvolvido baseado na metodologia ágil de gerenciamento de projetos, mas também pode ser utilizado nas metodologias em cascata.

O objetivo do desenvolvimento de um produto utilizando o LRCF é identificar os anseios do cliente no início do projeto, convertendo-os em requisitos de projeto. Neste processo deve ser considerada a importância de cada funcionalidade para o cliente. A ordenação das funcionalidades de acordo com sua prioridade permite direcionar o processo de desenvolvimento. Outra vantagem da priorização é identificada nas situações em que é necessária uma redução do escopo de projeto.

Um projeto que possui requisitos bem definidos gera um produto que satisfaz as expectativas do cliente. Além da maior satisfação do cliente, quando um grupo de pessoas está envolvido em um projeto e têm expectativas sobre o produto, podem existir requisitos conflitantes. Ao unir todas estas pessoas em momentos oportunos, as ideias antagônicas são convertidas em uma solução que atende as necessidades do grupo e não de apenas alguns dos interessados que serão beneficiados pelo produto.

A integração entre cliente e desenvolvedor proposta pelo LRCF aumenta a produtividade e o produto é desenvolvido com mais qualidade. Dunn (2008) afirma que o relacionamento do cliente com os desenvolvedores durante o planejamento do projeto e as entregas parciais do software tornam o trabalho mais produtivo e deixam a equipe mais engajada e motivada.

A construção de um produto alinhado às expectativas do cliente gera percepção maior de retorno de investimento, pois o cliente está recebendo algo dentro das especificações que realizou.

Definir melhor os requisitos de um projeto pode auxiliar na identificação de riscos, pois o trabalho que deve ser desenvolvido está melhor definido e podem ser identificados caminhos críticos do desenvolvimento. Ainda, resultado de um melhor detalhamento de requisitos pode ser empregado para organizar os casos de uso do produto e, conseqüentemente, refinar o processo de testes. Por isso, podem ser realizados trabalhos futuros que estudem o impacto da utilização do método LRCF no gerenciamento de riscos de um projeto e nos testes de funcionamento.

REFERÊNCIAS

Haughey, Duncan. A Brief History of Project Management. Disponível em: <<https://www.projectsart.co.uk/brief-history-of-project-management.php>>. Acesso em 01 jun. 2016

Beard, Ross. Why Customer Satisfaction is Important (6 Reasons). 20 jan. 2014. Disponível em <<http://blog.clientheartbeat.com/why-customer-satisfaction-is-important/>>. Acesso em 01 jun. 2016.

Dunn, Roger. Staying Agile in an uncertain economy. IBM Developers Work. Edição de 15 mai. 2008. Disponível em: <<http://www.ibm.com/developerworks/rational/library/edge/08/may08/dunn/index.html>>. Acesso em 11 jun. 2016.

Takeuchi, Hirotaka e Nonaka, Ikujiro. The New New Product Development Game. Harvard Business Review, Jan. 1986

Rubin, Kenneth S., Essential Scrum: A Practical Guide to the Most Popular Agile Process. Pearson Education, Inc. 2012, 452 p.

Leffingwell, Dean. Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise. Pearson Education, Inc. 2011, 524 p.

Leffingwell, Dean. Agile software requirements : lean requirements practices for teams, programs, and the enterprise. Pearson Education, Inc. 2011, 518 p.

Schwaber, Ken e Sutherland, Jeff. The Definitive Guide to Scrum: The Rules of the Game.

Jul. 2013, Scrum Alliance. disponível em: <<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>>. Acesso em 12 jun. 2016

Cohn, Mike. Agile Estimating and Planning. Prentice Hall. 2005, 368 p.

Sommerville, Ian. Software Engineering. Pearson Education, Inc., 2011 9 Ed. 773 p.