



A Utilização de Histórias de Usuários no Levantamento de Requisitos Ágeis

Hugo Estevam Romeu Longo*

Madalena Pereira da Silva**

Resumo

A partir da popularização dos métodos ágeis, surgem discussões sobre o que são histórias de usuário, porque utilizar histórias de usuário e como se deve descrever da melhor maneira o comportamento de um sistema através da engenharia de requisitos. O objetivo deste artigo é analisar se a utilização de histórias de usuário é suficiente durante a fase de levantamento de requisitos, abordando as diferentes maneiras de descrever, compreender e utilizá-las atualmente. Para ilustrar as fases de desenvolvimento da história de usuário é apresentado um estudo de caso que visou aplicar técnicas deste método comparadas às técnicas de levantamento de requisitos dirigidas por casos de uso.

Palavras-Chave: Métodos Ágeis. Desenvolvimento de Software

* Professor do Curso de Sistemas de Informação na Universidade do Planalto Catarinense, Arquiteto de Software da Empresa NDD Digital. hugoestevamrl@hotmail.com

** Professora da Universidade do Planalto Catarinense e Orientadora de monografias da Pós Graduação em Engenharia de Software da referida instituição. madalena.silva@posgrad.ufsc.br

1 Introdução

Durante décadas a proposta de utilizar casos de uso, introduzida em 1986 por Ivar Jacobson, conhecido como o principal contribuidor para o surgimento da UML (*Unified Model Language*) e o UP (*Unified Process*) foi amplamente utilizada para descrever requisitos funcionais.

A ideia de Jacobson (1992) de introduzir o caso de uso foi muito apreciada, tendo como principais virtudes a simplicidade e utilidade. Os cenários originais de Jacobson para utilização dos casos de uso foram intencionalmente informais. Ele constatou que quando as pessoas precisam escrevê-los, criam uma resistência sempre que se tornam mais formais.

Porém ao deixar os casos de uso completamente informais, Jacobson (1992) constatou que apresentavam-se outras dificuldades. Surgiam questionamentos das pessoas sobre como fariam para vincular um grande número de casos de uso ou então, como saberiam se aquela escrita informal estava sendo executada da maneira correta.

Outras pessoas, mais especificamente ligadas ao desenvolvimento de ferramentas CASE (*Computer Aided Software Engineering*), consideram esta forma de trabalho informal como incompleta, necessitando de uma base matemática e de apoio de uma ferramenta CASE. Eles criam notações, relacionamentos e artefatos para fazer casos de uso mais "rigorosos".

Mesmo com as definições acima e com bom número de aceitação, os casos de uso ainda não eram formais o suficiente para os formalistas, que continuaram a procurar por um modelo totalmente formal. Ao mesmo tempo em que era muito formal para aqueles que gostavam de pensar em casos de uso informais, sem estrutura.

A atividade de pesquisa por décadas tentou resolver o impasse na definição dos casos de uso. Cada nova abordagem era apontada como uma *bala de prata*, para solucionar a indefinição. Porém, pouco a pouco houve um convencimento geral de que não havia tal solução mágica. Ferramentas CASE, especificação formal, processos, componentes etc. eram boas técnicas que ajudaram a Engenharia de Software a evoluir, mas

correntemente não se acredita mais em uma solução única e salvadora para os complexos problemas envolvidos com a produção de software.

Nos anos 2000, o crescimento da demanda por software em organizações de pequeno e médio porte levou ao surgimento de soluções mais simples e efetivas para o desenvolvimento de software para essas organizações. Assim surgiram os métodos ágeis, que são caracterizados como informais e minimamente documentados. Além disso, esses processos dão mais destaque à comunicação verbal e social, na equipe de desenvolvimento. Em contraste, os processos tradicionais que são sequenciais, possuem fases que enfatizam a formalidade, o trabalho de escrita e a comunicação.

Nos últimos anos, os processos com base no Manifesto Ágil explicado por Highsmith e Fowler (2001) foram ganhando aceitação entre os praticantes. Os princípios por trás deste manifesto sugerem que a mudança deve ser acolhida em todas as fases do ciclo de desenvolvimento de software, que o software de trabalho deve ser entregue com frequência, e que a transmissão de informações por meio de uma conversa face a face é mais eficiente do que através de documentação escrita.

O processo ágil de software, Extreme Programming (XP), Beck (2000) introduziu a prática de expressar requisitos de software na forma de histórias de usuário, descrições curtas de funcionalidade contadas a partir da perspectiva de um usuário. Quando os métodos ágeis são utilizados, nota-se que os requisitos são baseados nos valores da simplicidade, comunicação, coragem e feedback. Funciona, trazendo toda a equipe para trabalhar unida na utilização de práticas simples, com comunicação suficiente para permitir que a equipe veja quais são as melhores práticas para cada situação. Esta disciplina é o ponto chave para compreender o conceito de histórias de usuários e comunicar esses conceitos para os usuários de negócios, tendo a chance de entregar um produto que estes usuários corporativos vão aprovar dentro do tempo e orçamento previstos.

No processo de design iterativo e ágil de software, as histórias de usuário são fundamentais para a criação de requisitos nas subdisciplinas ágeis de XP e SCRUM, mas as histórias de usuário também são úteis como ferramenta para provocar o envolvimento do cliente. Nesse sentido, este artigo visa demonstrar, através de um

estudo de caso, de que forma as histórias de usuário podem ser usadas nos processos ágeis de software, bem como se as mesmas são promissoras e suficientes para substituir os casos de uso durante a etapa de levantamento de requisitos.

O artigo está organizado em seções, sendo que a seção 2 descreve os requisitos ágeis e as histórias do usuário. A seção 3 apresenta o estudo de caso. A seção 4 cita os trabalhos correlatos, seguida pela conclusão e referências bibliográficas.

2 Requisitos Ágeis Através de Histórias do Usuário

Em um ambiente ágil, a história de usuário é um instrumento de escrita utilizado no processo de levantamento de requisitos para descrever a especificação de uma funcionalidade do software, afirmam Beck e Fowler (2001). Normalmente, uma história de usuário é breve, pois consiste em apenas uma ou duas frases.

2.1 Protocolos da História de Usuário

A história de usuário é uma declaração informal de um requisito de usuário em vez de um grande documento de requisitos. A intenção real da história de usuário é fornecer à equipe uma capacidade para responder rapidamente o que o usuário quer e precisa. A história de usuário cria menos sobrecarga de documentação e mostra de forma rápida a evolução das necessidades do mundo real ou a descoberta de novos requisitos baseados no trabalho em andamento. Não é especificamente a descrição de um requisito de software, o problema do mundo real subjacente é que o componente de software é projetado para resolver adversidades enfrentadas pelo usuário final.

As descrições da história do usuário são tradicionalmente escritas à mão, em cartões de papel, Jeffries (2001) nomeou estes três aspectos para esta descrição: o cartão, a conversa e a confirmação. O cartão pode ser a manifestação mais visível de uma história do usuário, mas não é a mais importante. Enquanto o cartão pode conter o texto de uma história, os detalhes são trabalhados na conversa e, em seguida, registrada e verificada através da confirmação.

Uma história de usuário contém tudo o que é necessário para movimentar as memórias e inspirações da equipe de desenvolvimento, que mais tarde pode explorar a história com

maior profundidade. Durante o processo, uma série de conversas terá lugar entre o cliente e a equipe de desenvolvimento. Essas conversas capturadas como documentação adicional, serão anexadas ao cartão juntamente com todos os critérios descobertos nos testes de aceitação, servindo para concretizar os detalhes e o aspecto da conversa.

Testes de aceitação segundo Cohn (2004) estão associados às histórias do usuário para especificar os requisitos de software. São descritos como os comportamentos que o sistema em desenvolvimento necessita para atender às regras de negócio do cliente. É um requisito do sistema de software formulado como uma ou duas frases na linguagem cotidiana do usuário de negócios. Idealmente, as histórias de usuário são escritas pelo cliente e são o seu principal método de influência no desenvolvimento dos sistemas. Os testes de aceitação fazem parte do aspecto de confirmação e dão origem aos casos de teste do sistema.

A escrita da história do usuário deve ocorrer sem o uso de qualquer jargão técnico. Elas devem ser compreensíveis pelos homens de negócios e seu conteúdo deve caber em um cartão de índice. Deve ser possível explicá-los em 30 segundos e implementá-los em menos de uma semana, de acordo com Cockburn (2002).

As histórias de usuário no XP de acordo com Jeffries (2001) têm três componentes: Cartões, que são o meio físico no qual as histórias são escritas; Conversação, que é a discussão em torno das histórias; e Confirmação, que são os testes que verificam as histórias.

Histórias de usuário são escritas em um cartão de nota, com um nome e uma descrição. Se mais tarde, a história de usuário está muito grande, complicada ou imprecisa, ela será reescrita, até que fique satisfatória para todas as partes. No entanto, salienta-se em Beck (2000) que as histórias de usuário não devem ser definidas já na primeira vez que elas foram escritas, devendo-se evoluir à medida que as necessidades surgirem.

O formato simples de história de usuário apresentado por Cohn (2004) auxilia o começo da comunicação no âmbito de um projeto ágil. Devendo-se estar atento para não tratar a história como uma especificação de um mini requisito focado na palavra escrita, mas

sim utilizar histórias como ferramenta para a condução de uma conversa em um momento posterior.

A sintaxe de uma história de usuário sugerida por Cohn (2004) deve conter:

- **Como um...** (papel ou ator) (Quem)
- **Eu quero...** (capacidade ou funcionalidades necessárias) (O que)
- **de modo que...** (por que é de valor do negócio ou benefício) (Por que)

Cohn (2004) sugere a escrita de uma história de usuário como se estivesse na pele do usuário que deseja a história. O uso na perspectiva de primeira pessoa torna a história mais compreensível. Nesse caso, deve-se identificar o papel que está assumindo dentro da história:

"Como um tipo de usuário eu quero capacidade ou funcionalidade de modo que o valor do negócio ou benefício"

O "Quem" e "O que" são essenciais para a história, já o "Por que" só ajuda com clareza na configuração dos testes de aceitação. Histórias de usuário ajudam a fazer as perguntas sobre o contexto e razão para o pedido da pessoa que solicita o requisito. Também é possível verificar o nível de prioridade e a estimativa inicial de cada história de usuário conforme o modelo formal apresentado por Cohn (2004). Abaixo na Figura 1 é demonstrado um exemplo de cartão de história.

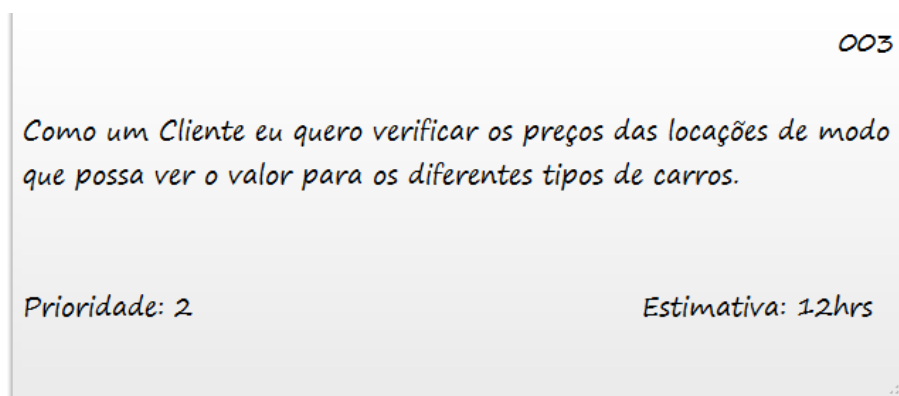


Figura 1. Exemplo de um cartão de história de usuário adaptado da fonte Kniberg (2008)

No modelo do cartão de história mostrado no trabalho de Kniberg (2008), o autor menciona que um cartão de história deve possuir alguns campos obrigatórios que permitam ao menos servir aos objetivos propostos, de maneira geral, o cartão de história pode ser alterado de acordo com as necessidades, mas recomenda-se inicialmente tentar utilizar apenas os campos descritos abaixo:

- **ID** - Um número sequencial, apenas para identificar a ordem de criação dos cartões de história e certificar que nenhum se perdeu.
- **Nome** - Título de poucas palavras, relevante para que o cliente e a equipe entendam bem do que se trata e distinguir das outras histórias.
- **Importância** - O mesmo que prioridade. Quanto maior o número, mais relevante é a história. Porém, uma história considerada mais importante não é 2, 3 ou 4 vezes mais primordial que outra, ela é apenas mais importante. É interessante que sejam escolhidos valores bem espaçados, de uma história para a outra, para que seja possível acrescentar histórias com valor de importância intermediário.
- **Estimativa inicial** - Calculada em pontos por história.
- **Como demonstrar** - Descrição em alto nível de como a história será demonstrada ao fim da iteração, isto é, como uma especificação de teste de aceitação, "Faça isso, então faça aquilo e então isso deverá acontecer."
- **Notas** - Outras informações, esclarecimentos, referências a outras fontes de informação etc. Normalmente breve.

2.2 Distinção da História de Usuário

As histórias de usuário segundo Cohn (2004) apresentam algumas das características dos casos de uso ou das declarações tradicionais de requisitos, ele afirma que é importante olhar para o que distingue as histórias de usuário das técnicas anteriores, e que estas diferenças podem resultar em inúmeras vantagens.

Uma das principais diferenças das histórias de usuário está na comunicação verbal. A linguagem escrita é muitas vezes imprecisa, e não há nenhuma garantia de que um cliente e/ou desenvolvedor irá interpretar uma declaração da mesma forma. Neste sentido é enfatizada a conversa com o cliente em contraste com as palavras escritas.

A segunda vantagem das histórias de usuário é que elas podem ser utilizadas prontamente no planejamento do projeto. Histórias de usuário são escritas de modo que cada uma possua estimativa de quão difícil ou demorado será o desenvolvimento, em contra ponto aos casos de uso, que por outro lado, são geralmente maiores para ser atribuídas estimativas úteis. Além disso, uma história é implementada inteiramente em determinada iteração de um projeto ágil.

Cohn (2004) apresenta como problema, o estilo de especificação de requisitos proposto pelo IEEE (1998), segundo o autor, quando se consideram as milhares ou dezenas de milhares de declarações do tipo (“o sistema deve [...]”) em uma especificação de requisitos de software (e as relações entre eles) para um sistema típico, é fácil perceber a dificuldade inerente em priorizá-los. Se os requisitos não podem ser priorizados para além do comum médio, alto e baixo, eles são altamente inadequados para um processo iterativo e de desenvolvimento incremental que vai entregar software funcionando a cada duas a quatro semanas.

Outra vantagem adicional das histórias de usuário destacada por Beck (2001) é que as histórias de usuário incentivam a equipe a adiar detalhes de coleta. Uma história inicial pode ser escrita e, em seguida, substituída com histórias mais detalhadas, uma vez que torna-se importante conhecer os detalhes. Esta técnica faz com que as histórias de usuário se encaixem em projetos com tempo limitado.

Uma equipe pode muito rapidamente escrever algumas histórias para dar-lhes uma sensação geral do sistema. Podem mergulhar nos detalhes sobre algumas das histórias e iniciar a codificação muito mais cedo do que uma equipe que se sente compelida a completar uma especificação de requisitos de software no estilo apresentado pela IEEE (1998).

2.3 Histórias de Usuário não são Casos de Uso

Os casos de uso introduzidos por Jacobson (1992) são hoje mais comumente associados com o Processo Unificado. Um caso de uso é uma descrição generalizada de um conjunto de interações entre o sistema e um ou mais atores, no qual um ator ou é um utilizador ou outro sistema. Os casos de uso podem ser escritos em textos não

estruturados ou para se conformar com um modelo estruturado. Os modelos propostos por Cockburn (2000) estão entre os mais usados.

Uma das mais óbvias diferenças entre as histórias de usuário e casos de uso é o seu alcance. Ambos são dimensionados para agregar valor ao negócio, mas as histórias são mantidas num tamanho menor, porque no desenvolvimento ágil, conforme descrito em Cockburn (2002) são definidas restrições ao seu tamanho (nenhuma história pode levar mais de 10 dias de trabalho de desenvolvimento) para que possam ser utilizadas na programação. Um caso de uso quase sempre abrange um escopo muito maior do que uma história.

As histórias de usuário e casos de uso também diferem no nível de completude. Grenning (2011) observou que os textos em um cartão de história de usuário que descrevem os testes de aceitação possuem os mesmos dados que um caso de uso. Por isso, Grenning (2011) sugere que a história do usuário corresponde ao cenário do caso de uso principal, e que os testes de aceitação da história correspondem aos fluxos alternativos do caso de uso.

Outra importante diferença entre casos de uso e histórias de usuário citada por Cohn (2004) é a sua longevidade. Os casos de uso são muitas vezes artefatos permanentes que continuam a existir, desde que o produto mantenha-se em desenvolvimento ativo ou manutenção. Histórias de usuário, por outro lado, não se destinam a sobreviver à iteração em que eles são adicionados ao software. Embora seja possível guardar os cartões de histórias em arquivo, muitas equipes simplesmente inutilizam os cartões.

Os casos de uso e as histórias de usuários são escritos para diferentes fins. Os casos de uso são escritos em formato aceitável para os clientes e desenvolvedores, para que cada um possa ler e concordar com o caso de uso. O propósito do caso de uso é documentar um acordo entre o cliente e a equipe de desenvolvimento. Histórias de usuário, por outro lado, são escritas para facilitar a liberação e planejamento da iteração, e para servir como espaços reservados para conversas detalhadas sobre as necessidades dos usuários.

2.4 Histórias de Usuário não são Requisitos

A Sociedade de Computação do Instituto de Engenheiros Elétricos e Eletrônicos (IEEE) publicou um conjunto de orientações sobre como escrever especificações de requisitos de software em IEEE (1998). Este documento, conhecido como padrão IEEE 830, traz recomendações que abrangem temas como a forma de organizar o documento de especificação de requisitos, o papel dos protótipos, e as características dos bons requisitos. A característica mais distintiva do padrão IEEE 830 na especificação de requisitos é o uso da frase "O sistema deve [...]", que é a maneira recomendada pelo IEEE para escrever requisitos funcionais.

Enquanto os requisitos sugerem o que deve ser feito, histórias de usuário focam nos objetivos, e isso torna a visão do produto completamente diferente. Ao se concentrar nos objetivos do usuário para o novo produto, ao invés de uma lista de atributos do novo produto, segundo Beck (2001) pode-se projetar uma melhor solução para as necessidades do usuário.

A diferença final entre as histórias de usuário e o padrão IEEE 830 de requisitos que foi abordada em Cohn (2004) é em relação ao custo do requisito que não é visível até que todos os requisitos sejam escritos. O cenário típico é que um ou mais analistas passam dois ou três meses escrevendo um documento de requisitos. Este documento é então entregue aos programadores, que detectam que o projeto levará 24 meses, invés dos seis meses previstos.

Neste caso, o tempo de escrita do documento foi desperdiçado por que a equipe não terá tempo para desenvolver, e mais tempo será dispendido com os desenvolvedores e analistas, que deverão iterar com o cliente sobre quais funcionalidades podem ser desenvolvidas no tempo desejado. Com histórias de usuário, uma estimativa está associada com cada história imediatamente. O cliente sabe a velocidade da equipe e o custo de cada história.

2.5 Envolvendo o Cliente no Processo de Desenvolvimento

Um dos poucos requisitos de XP apresentado por Beck (2001) é ter o cliente disponível para ajudar a equipe de desenvolvimento e ser uma parte do mesmo processo. Todas as

fases de um projeto XP requerem comunicação com o cliente, de preferência face a face *in loco*.

Ao fazer o processo de levantamento das histórias de usuário da forma mais simples possível, fornece-se aos usuários de negócios uma ferramenta para reunir histórias de usuários e, portanto, requisitos do sistema e decidir quais serão os requisitos que terão maior benefício com a implementação imediata no projeto e quais poderão ter implementação posterior. Segundo Cockburn (2002) o usuário de negócio e a equipe de desenvolvimento podem, então, priorizar as histórias e implementar uma solução dentro do prazo estipulado. Esse nível de envolvimento de negócio dá ao cliente um sentimento de confiança no projeto e um forte senso de que sua contribuição é vital para uma implementação bem sucedida.

Com pequenas entregas e histórias facilmente digeríveis em cartões ao invés de um documento de requisitos de cem páginas, faz com que o cliente sinta-se como parte do negócio e que a sua colaboração contribui para a entrega final do projeto.

2.6 Testes de Aceitação

Como parte do planejamento, histórias de usuário definem o que se quer construir no projeto. As histórias de usuário são priorizadas para indicar quais são as mais importantes para o sistema, como são divididas em tarefas e estimadas pela equipe de desenvolvimento. Quando uma história de usuário é implementada, de acordo com Kniberg (2008), um teste de aceitação mais formal, deverá ser escrito para assegurar que os objetivos da história sejam cumpridos.

Testes de aceitação são criados a partir de critérios de aceitação que segundo Cohn (2004) são descritos no verso do cartão da histórias de usuário. Os critérios de aceitação definem os limites de uma história de usuário, são usados para confirmar quando uma história é concluída e está funcionando conforme o esperado e são escritos em linguagem simples, assim como a história do usuário. Quando a equipe de desenvolvimento concluir a história do usuário deverá demonstrar a funcionalidade para o cliente, mostrando como cada critério está satisfeito. A Figura 2 apresenta um exemplo dos critérios de aceitação.

Critérios de Aceitação	
	Descrição do comportamento
Dado	Pré-condições
Quando	Ator + Ação
Então	Resultado esperado

Figura 2. Exemplo de template dos Critérios de Aceitação. Fonte Cohn (2004)

Cada teste de aceitação representa algum resultado esperado a partir do sistema. Deve-se verificar a correção dos testes de aceitação e resultados, para decidir quais são os de maior prioridade que falharam. Uma história de usuário não é considerada completa até que seus testes de aceitação tenham sido aprovados.

3 Estudo de Caso

Esta seção descreve o escopo do sistema que será usado como fonte de estudo para aplicação dos métodos ágeis no levantamento de requisitos e apresenta os diagramas de caso de uso, atividade e classe, utilizando-se a UML. A seção apresenta também a aplicação dos requisitos ágeis no sistema proposto, com o objetivo de verificar a possibilidade de substituição dos casos de uso, através da utilização das histórias de usuário durante o levantamento de requisitos, identificando e mapeando as funcionalidades.

3.1 Descrição do Projeto

O estudo de caso trata de um sistema de rede de locação de veículo, que tem como principal objetivo controlar a locação dos veículos da empresa. Os veículos são divididos em grupos, o que possibilita a escolha por parte do cliente. Para a realização da reserva, os funcionários da rede de locação fazem suas respectivas análises tendo como base o cadastro dos seus clientes. Caso haja restrição de acordo com as regras, o funcionário preenche formulário na Internet com todas as informações da restrição da

reserva. Na rede de locação, o funcionário responsável pela entrega e recebimento dos veículos, cadastra nos dados da reserva as informações obtidas durante a locação.

Para realização da reserva é necessário que o cliente tenha habilitação para dirigir o veículo de acordo com o grupo escolhido, deve também ser observado o histórico de outras locações do cliente. Caso exista alguma restrição, o responsável pela reserva deve notificar o cliente através do sistema.

O estudo foi iniciado com a fase de exploração, conceituada anteriormente na seção 2, cujo objetivo foi compreender o que o sistema precisava fazer, bem como analisar o suficiente para que pudesse ser estimado. Essa estimativa foi preliminar, e partiu dos primeiros requisitos obtidos no estudo.

Em paralelo às histórias dos usuários, foram avaliadas as diferentes formas de registrar os requisitos levantados no desenvolvimento de um sistema. Nesta etapa, foram utilizados os modelos de casos de uso propostos por Cockburn (2000).

O contexto da aplicação sugere que o sistema fique disponível por meio de serviços *on-line*, neste sentido foi identificada a necessidade de utilizar os conceitos de SOA (*Service Oriented Architecture*), desenvolvendo um Provedor, seguindo os conceitos de distribuição destes serviços e posteriormente o seu uso através dos Consumidores.

Diante das informações supracitadas, os casos de uso apresentados neste estudo de caso foram criados com base em dados obtidos através de entrevistas não estruturadas. Foi escolhido este formato, devido a geração espontânea das perguntas no fluxo natural de uma interação. Este tipo de entrevista é apropriado quando o entrevistador quer manter o máximo de flexibilidade na entrevista para direcionar o questionamento no sentido que lhe parecer apropriado, explorando as informações contidas nas respostas ou em conversas com os indivíduos participantes, que descrevem as regras de negócio.

Foram utilizadas algumas atividades básicas no preparo da entrevista: estudar o domínio do problema, determinar objetivos e lista de questões, planejar e agendar os encontros. A Tabela 1 apresenta a aplicação da entrevista não estruturada no estudo de caso.

Tabela 1. Dados obtidos na aplicação da entrevista não estruturada

Objetivos da Entrevista <ul style="list-style-type: none"> • Obter informações relevantes dos clientes, funcionários e veículos; • Entender o processo de reserva, identificar informações necessárias; • Analisar o processo de reserva do cliente;
Entrevistados <ul style="list-style-type: none"> • Funcionários; • Clientes;
Perguntas <ul style="list-style-type: none"> • Quais campos são necessários para o cadastro do cliente? • Os carros são categorizados por quais grupos? • As reservas serão informadas ao cliente e aos funcionários? • As reservas possuem descontos ou multas? • Para efetuar a reserva é necessário verificar se o cliente tem alguma restrição? • Na reserva, o cliente deve estar previamente cadastrado? • Quais informações são necessárias para efetuar a reserva?
Resumo da Entrevista <ul style="list-style-type: none"> • Informações dos clientes: nome, cpf/cnpj, inscrição municipal, inscrição estadual, endereço, bairro, complemento, cidade, estado e cep; • Os grupos são fixos: Econômicos, Utilitários e Luxo; • O funcionário necessita visualizar as reservas preenchidas pelos clientes e marcar como lida e não lida via formulário na web; • O desconto ou multa é informado no momento da efetuação do pagamento, com base nos dados da locação; • O controle de reserva, inicia com a verificação dos dados do cliente, avaliando se existe alguma restrição, como carteira de habilitação e histórico de multas;

O modelo de caso de uso proposto foi obtido através das informações fornecidas em entrevistas com o cliente. É um modelo que tenta representar os atores envolvidos no domínio da aplicação, porém sem maiores detalhes, serve para aprimorar o entendimento e discussão dos requisitos. Foram utilizados inicialmente neste trabalho para descobrir e registrar os primeiros requisitos de sistemas que podem ser vistos na Figura 3.

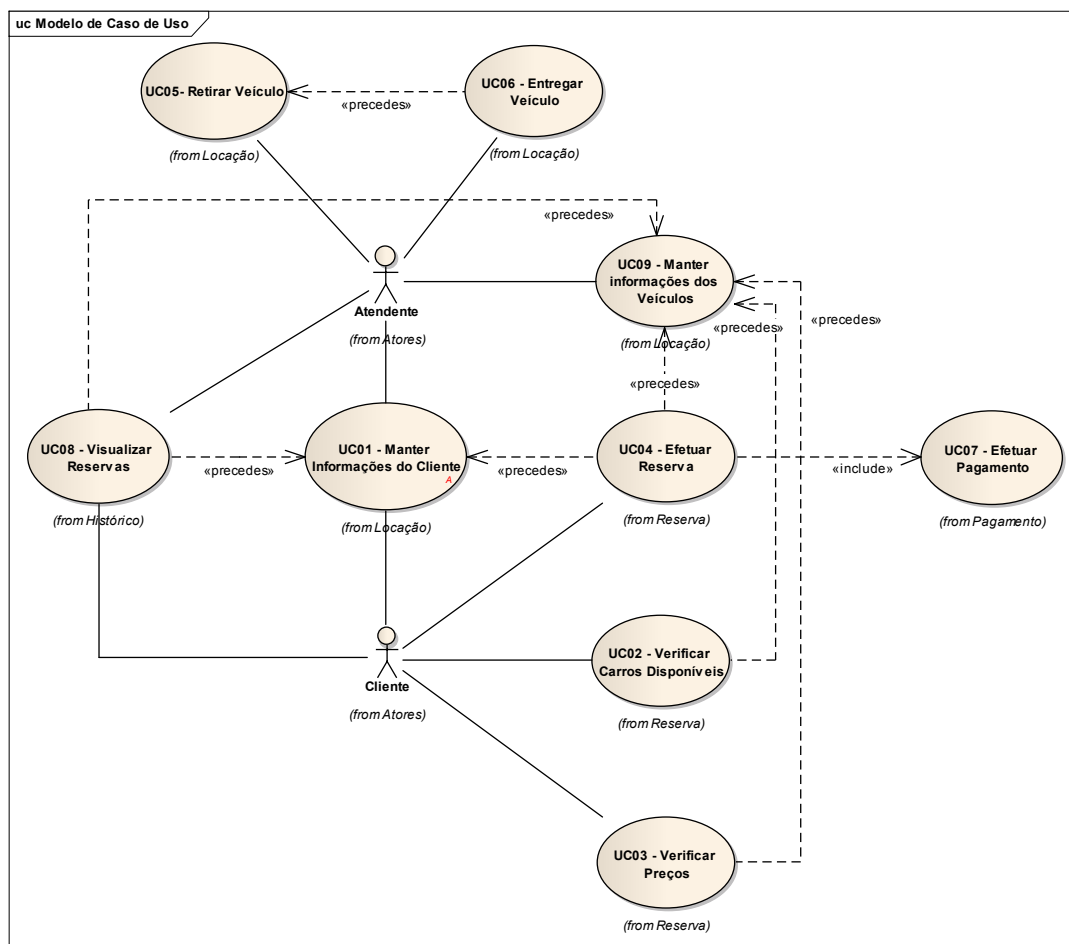


Figura 3. Modelo de caso de uso do sistema proposto no estudo de caso

É preciso ressaltar que as listas de regras de negócio não são equivalentes as de requisitos. Kulak e Gurney (2005) destacam que os casos de uso não podem retratar todas as sutilezas de como a empresa é gerida. Para isso, precisa-se das regras de negócio. As regras de negócios são as regras escritas e não escritas que ditam como uma empresa ou agência conduz seus negócios. Requisitos relacionam-se com uma aplicação específica a ser considerada ou desenvolvida.

A obtenção das regras de negócio do sistema (Figura 4) foi facilitada pelo fluxo de negócio descrito pelo cliente. De acordo com a orientação de Ambler (2012), projetos ágeis devem focar nos requisitos com maior nível de prioridade em cada iteração, sabendo que é preciso considerar as regras de negócio como requisitos separados. Ambler (2012) ainda afirma "as equipes de desenvolvimento de software ágeis devem abraçar as mudanças, aceitando a ideia de que as exigências vão evoluir ao longo de um projeto".

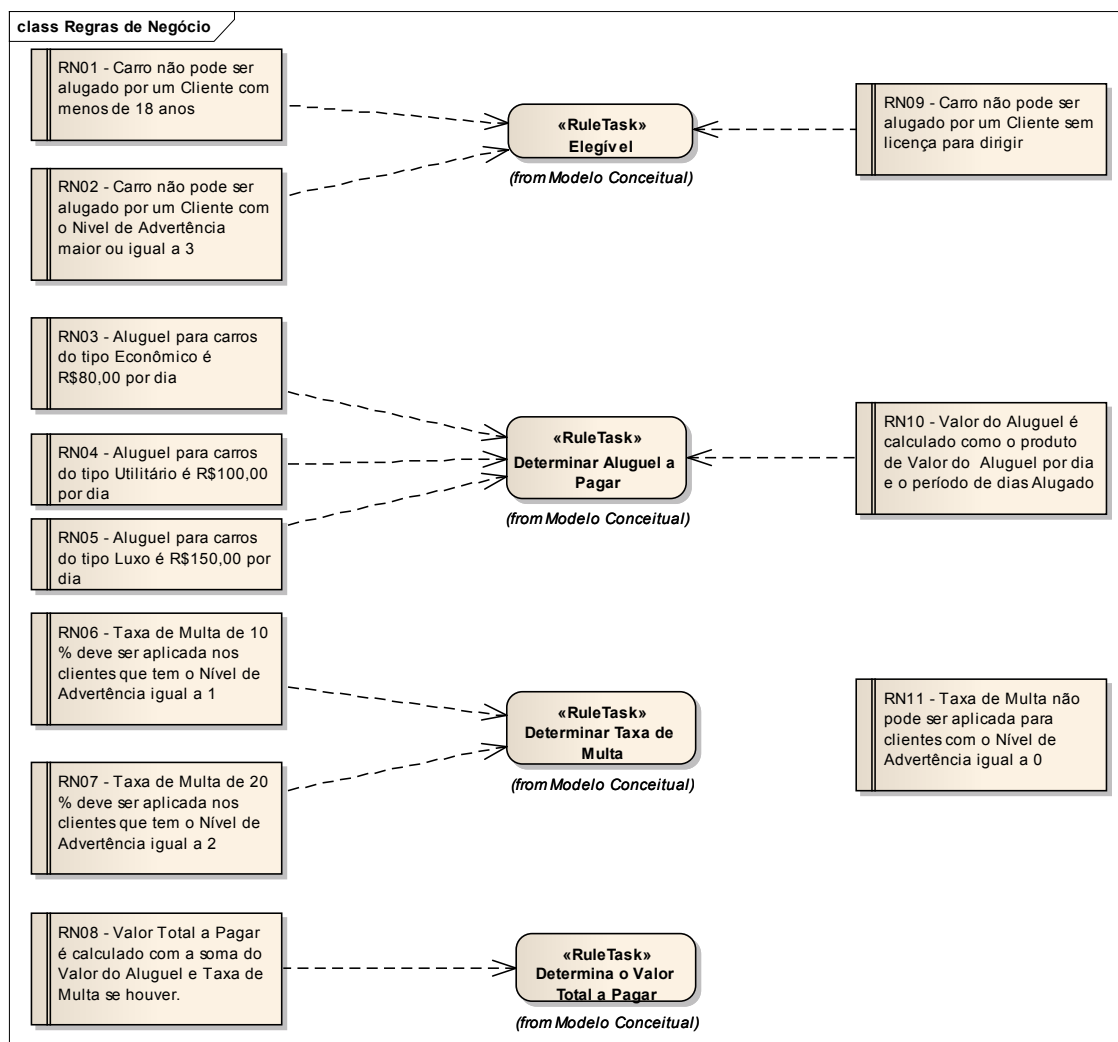


Figura 4. Regras de negócio do sistema proposto no estudo de caso

3.2 Aplicação do Caso de Uso

Os casos de uso são amplamente utilizados em processos de software baseados em uma abordagem disciplinada, como o *Rational Unified Process* (RUP) segundo Kruchten (2003). Este formato tem o objetivo de descrever o conjunto de interações e eventos entre os usuários ou sistemas externos (também conhecidos como atores).

Essas descrições incluem as funcionalidades do sistema, necessárias para satisfazer as exigências do solicitante. Há diferentes diretrizes para escrever casos de uso e a eficácia de um caso de uso depende da capacidade do autor para escrevê-los. Um exemplo de um caso de uso é apresentado na Tabela 2.

Tabela 2. Descrição do caso de uso "UC04 - Efetuar Reserva"

Nome:	UC04 - Efetuar Reserva
Sumário:	O sistema deve permitir que os clientes que desejam realizar locações de carros efetuem uma reserva. Este caso de uso recebe solicitações de inclusão, alteração, exclusão de reserva.
Ator:	Cliente
Entrada:	Dados do Cliente, Informações do Veículo, Dados de pagamento, Data da reserva.
Pré-condições:	Para efetuar reserva o cliente deve possuir um cadastro ativo no sistema.
Fluxo Principal:	<ol style="list-style-type: none"> 1. O Cliente insere as informações de entrada e solicita a reserva [RF07]; 2. O Sistema verifica as informações de acordo com as regras de negócio [RN01], [RN02], [RN09]; Se a avaliação das regras de negócio não aprovar a reserva vai para o [Fluxo Alternativo 1]; 3. O sistema registra uma reserva para o Cliente; 4. O Cliente visualiza o número da sua reserva gerada pelo sistema; 5. O Cliente pode alterar a sua reserva [Fluxo Alternativo 2]; 6. O Cliente pode cancelar a sua reserva [Fluxo Alternativo 3].
Fluxo Alternativo 1:	O sistema retorna uma mensagem avisando que não é possível efetuar a reserva pois o Cliente não passou na validação da regra de negócio (Exibe as informações das regras que o Cliente não atendeu).
Fluxo Alternativo 2:	O Cliente solicita a alteração dos dados da reserva [RF08]. Devem ser executados os passos do Fluxo Principal novamente.
Fluxo Alternativo 3:	O Cliente solicita o cancelamento da reserva. Após uma mensagem de confirmação, o sistema deve cancelar a reserva [RF09].
Requisitos Relacionados:	<p>RF07 - Realizar a reserva de veículo</p> <p>O sistema deve permitir que o cliente efetue a reserva de um veículo de acordo com as regras de negócio [RN01], [RN02] e [RN09]</p> <p>RF08 - Alterar as informações da reserva</p> <p>O sistema deve permitir que o Cliente altere as informações de uma reserva.</p> <p>RF09 - Excluir a reserva de veículo</p> <p>O sistema deve permitir que o cliente exclua uma reserva.</p>

Este caso de uso apresentado corresponde às histórias do usuário que serão apresentadas na subseção posterior. Como pode ser visto na Tabela 1, um caso de uso contém o nome, o sumário, as condições, os dados de entrada, os atores, a descrição de cada passo no cenário principal, fluxos de exceção e os requisitos relacionados.

Desta forma, o caso de uso contém informações específicas sobre os passos que o usuário deve seguir para usar com sucesso a nova funcionalidade do sistema. Também estão incluídos os detalhes, tais como as regras de negócio, validações específicas (por exemplo, o carro não pode ser alugado por um cliente com menos de 18 anos) e as mensagens de erro.

Na Figura 5 é apresentada uma imagem utilizando a notação UML para ilustrar os processos em ação.

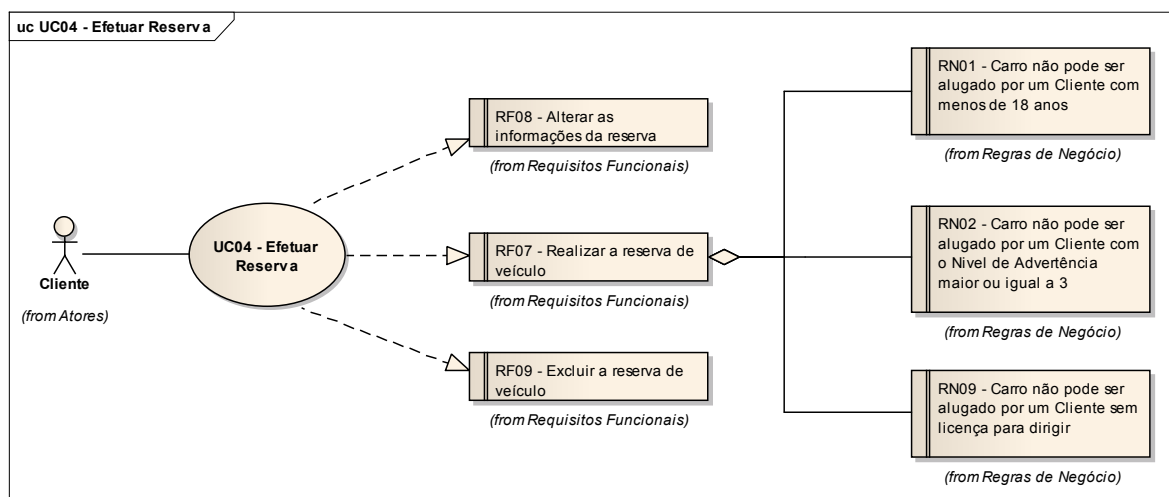


Figura 5. Digrama de caso de uso "UC004 - Efetuar Reserva"

3.3 Aplicação da História de Usuário

Segundo Cohn (2004) as histórias de usuário são criadas pelo cliente, ou com participação dele e apoio da equipe, no momento do levantamento de requisitos. Com base na entrevista descrita na seção 3.1 foram descobertas as funcionalidades a serem implementadas neste estudo de caso. Essas funcionalidades (requisitos ou histórias de usuário) são mantidas em uma lista chamada de *product backlog*.

Um dos princípios do manifesto ágil é usado aqui: adaptação ao invés de planejamento. O *product backlog* não precisa então ser completo no início do projeto. Pode-se iniciar apenas com as funcionalidades mais evidentes para depois, à medida que o projeto avança tratar novas funcionalidades que vão sendo descobertas. Isso, porém, não significa fazer um levantamento inicial excessivamente superficial.

Desta forma, foi necessário junto ao cliente obter o maior número possível de informações sobre suas necessidades. Aquelas que efetivamente surgiram nessa interação tiveram maior relevância, em relação a outras que forem descobertas mais adiante. A Tabela 3 apresenta as histórias de usuário no *product backlog*, estruturadas segundo definição de Kniberg (2008) detalhada na seção 2.

Tabela 3. Lista de histórias de usuário que representam o caso de uso utilizado no estudo

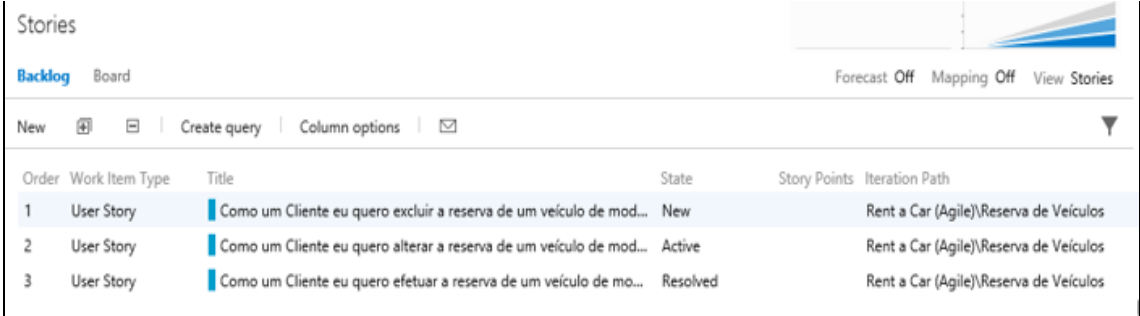
<i>Product Backlog</i>					
ID	Nome	Imp.	PH	Como demonstrar	Notas
1	Cadastrar Reserva	30	5	Como um Cliente eu quero efetuar a reserva de um veículo de modo que possa realizar a locação posteriormente.	Necessário ter um diagrama de sequência.
2	Alterar os dados da reserva	20	5	Como um Cliente eu quero alterar a reserva de um veículo de modo que possa alterar as informações da reserva.	Usar bloqueio de persistência otimista.
3	Excluir Reserva	10	8	Como um Cliente eu quero excluir a reserva de um veículo de modo que possa cancelar a reserva anteriormente criada.	Manter um log com as informações de quem excluiu a reserva.

O ideal é que seja apenas uma pessoa a criar as histórias e priorizá-las para repassar à equipe de desenvolvimento. As histórias representam a materialização das necessidades do cliente em relação ao software. Ou seja, são as especificações detalhadas das funcionalidades desejadas.

O principal objetivo de serem escritas com o cliente sempre presente é manter o foco em objetivos do negócio. Ainda que o cliente possua algum conhecimento técnico, é papel da equipe determinar como resolver um problema tecnicamente. O cliente deve focar nos objetivos do negócio.

Os registros das histórias de usuário foram elaborados utilizando o software *Team Foundation Server* (TFS), juntamente com um conjunto de templates para a utilização do SCRUM (*product backlog*, *sprint*, *user story*, *task*, *acceptance test* etc.) desenvolvidos por Microsoft (2013) e disponibilizados gratuitamente.

A Figura 6 apresenta lista de histórias de usuário registradas no *product backlog* da ferramenta TFS.



The screenshot shows the 'Stories' backlog in TFS. It includes a 'Forecast' chart at the top right and a table of user stories below. The table has columns for Order, Work Item Type, Title, State, Story Points, and Iteration Path.

Order	Work Item Type	Title	State	Story Points	Iteration Path
1	User Story	Como um Cliente eu quero excluir a reserva de um veículo de mod...	New		Rent a Car (Agile)\Reserva de Veiculos
2	User Story	Como um Cliente eu quero alterar a reserva de um veículo de mod...	Active		Rent a Car (Agile)\Reserva de Veiculos
3	User Story	Como um Cliente eu quero efetuar a reserva de um veículo de mo...	Resolved		Rent a Car (Agile)\Reserva de Veiculos

Figura 6. Lista de histórias de usuário cadastradas no TFS (Team Foudation Server)

Foi elaborada uma estimativa com base nos cartões de história, na metáfora e em uma solução simples (ASTELS et al., 2002). O objetivo desta fase foi estimar o menor tempo e o maior número de histórias para a primeira versão (BECK, 2000). O projeto também poderia ser dividido em várias iterações, sendo necessário identificar os seus respectivos entregáveis.

Com base nessas estimativas, a equipe se comprometeu a desenvolver as funcionalidades, e o cliente se comprometeu a não trazer novas funcionalidades durante o período estipulado. Se novas funcionalidades fossem descobertas, seriam abordadas em períodos posteriores. Este ciclo de desenvolvimento, com poucas semanas de duração sobre o qual se estrutura o SCRUM são os chamados *sprints*.

No início de cada *sprint* foi feito uma reunião de planejamento, na qual a equipe priorizou os elementos do *product backlog* a serem implementados, e transferiu estes elementos do *product backlog* para o *sprint backlog*, ou seja, a lista de funcionalidades a serem implementadas no ciclo que iniciou.

Durante o *sprint*, coube ao responsável manter o *sprint backlog* atualizado, indicando as tarefas já concluídas e as ainda por concluir. Neste estudo de caso foi escolhida a ferramenta TFS para atualizar as tarefas diariamente à vista de todos. Um exemplo do quadro de andamento de atividades é apresentado na Figura 7.

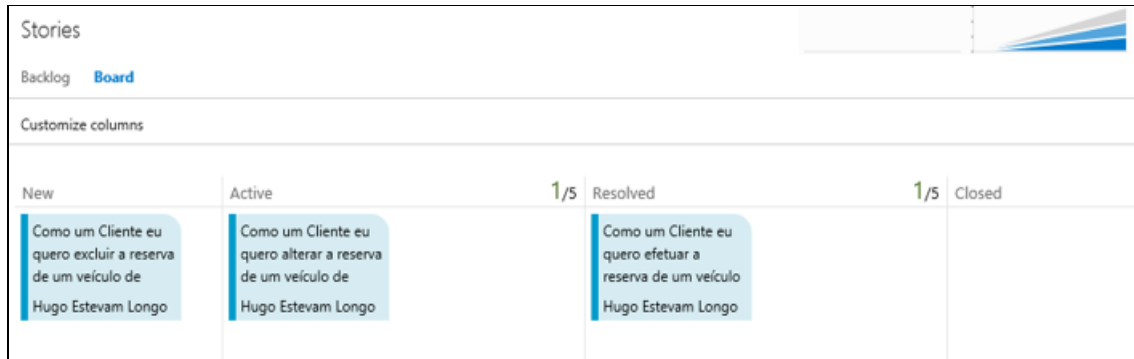


Figura 7. Quadro Kanban com o controle da lista de histórias de usuário

3.4 Teste de Aceitação do Estudo de Caso

Durante uma iteração, as histórias selecionadas pelo usuário na reunião de planejamento de iteração foram traduzidas em testes de aceitação com base nos respectivos critérios de aceitação. Uma história pode ter um teste de aceitação ou muitos, o que for preciso para garantir que a aprovação da funcionalidade.

O teste de aceitação foi realizado com o objetivo de verificar se o produto foi desenvolvido de acordo com as normas e critérios especificados e atende a todos os requisitos especificados pelo cliente. Neles foram aplicados a metodologia de teste de caixa preta, onde o usuário não está muito interessado na codificação do sistema, mas sim, na avaliação do funcionamento global, comparando-o com os requisitos especificados no início do projeto.

Com base nos requisitos especificados pelo usuário, a execução dos casos de teste ocorreu através da maneira iterativa ou sob a forma de um conjunto de parâmetros variáveis. O sistema teve que passar a operar num ambiente de computação que imitou o ambiente operacional real existente no contexto do usuário. Os resultados dos testes de aceitação foram denominados como: sucesso ou fracasso com base nas condições de funcionamento críticos e sucesso ou fracasso com base à avaliação final do usuário do sistema.

A Figura 8 apresenta a utilização dos testes de aceitação aplicados no estudo de caso.

User Story 17: Como um Cliente eu quero efetuar a reserva de um veículo de modo que possa realizar a locação posteriormente.

Tags Add...

Como um Cliente eu quero efetuar a reserva de um veículo de modo que possa realizar a locação posteriormente.

DETAILS STORYBOARDS IMPLEMENTATION **TEST CASES (3)** ALL LINKS (3) ATTACHMENTS HISTORY

Assigned To Hugo Estevam Longo
State Resolved
Reason Code complete and unit tests pass

ID	Work Item...	Title	Assigned To	State	Comment
20	Test Case	O carro não pode ser alugado por um Cliente com menos de 18 anos.	Design	Design	
21	Test Case	O carro não pode ser alugado por um Cliente com o nível de advertência maior o...	Design	Design	
22	Test Case	O carro não pode ser alugado por um Cliente sem licença para dirigir.	Design	Design	

Tested By (3)

PLANNING
Story Points
Risk
Stack Rank 500000

CLASSIFICATION
Area Rent a Car (Agile)
Iteration Rent a Car (Agile)\Reserva de Veículos

Figura 8. Critérios de aceitação cadastrados para a história de usuário

3.5 Comparação da História de Usuário com o Caso de Uso

Casos de Uso são mais representativos que Histórias de Usuário, pois podem variar entre dois parágrafos e dez páginas. Eles são bons para mostrar os caminhos alternativos de uma característica específica. As histórias também poderiam conseguir isso, desde que fossem escritos os caminhos excepcionais em diferentes histórias de usuário, mas esta abordagem é uma adaptação da técnica ao invés de um planejamento de uso, o que contrasta com os casos de uso.

Normalmente, as histórias não serão suficientes em uma organização na qual a documentação formal é obrigatória. Sua principal diferença em relação aos casos de uso ou cenários é que histórias de usuário têm o objetivo de capturar a perspectiva de que o usuário tem sobre o sistema. Algumas das diferenças entre os Casos de Uso e as Histórias de Usuário são apresentadas na Tabela 4.

Tabela 4. Diferenças entre casos de usos e histórias de usuário.

Caso de Uso / Cenários	Histórias de Usuário
São apresentados usando uma sintaxe restrita (semi-formal)	São apresentados usando a linguagem natural em forma de conversa
São especificações de interações de objetos	São descritivas e apresentam os desejos do cliente
Contêm "como"	Contêm "o quê" e "porquê"

Outro formato de levantamento de requisitos que é amplamente utilizado são os cenários. Algumas vezes, os termos de casos de uso e cenários são usados alternadamente. No entanto, o presente estudo usará a definição dada por Alessander e Maiden (2004) de que "Um caso de uso é sempre composto de vários cenários que descrevem formas alternativas para tentar atingir a meta".

Do ponto de vista de que os cenários fazem parte dos casos de uso, é possível avaliar que cada formato de exigência tem vantagens e desvantagens. Estudos como o de Alessander e Maiden (2004) questionam se os casos de usos são complementares às histórias de usuário ou se as tornam redundantes no desenvolvimento ágil.

O presente estudo tem como objetivo apresentar evidência para orientar equipes de desenvolvimento de software sobre a existência de qualquer benefício na utilização de casos de uso em relação às histórias de usuário, mas também, deseja apresentar as histórias como um novo método, além dos formatos de requisitos que estão sendo usados atualmente.

Na comparação entre casos de uso e requisitos ágeis, Alessander e Maiden (2004) afirmam que os dois métodos são complementares. Entretanto, outros autores apontam várias diferenças entre os dois métodos. Como principais diferenças destaca-se que a escrita dos requisitos no formato de casos de uso pode economizar tempo, se as pessoas lerem os documentos. No entanto, surge como ponto fraco a forte exigência de formatos para a escrita de casos de uso em relação aos requisitos ágeis.

As principais diferenças encontradas neste estudo, entre o levantamento de requisitos utilizando requisitos ágeis e casos de uso são apresentados na Tabela 5.

Tabela 5. Comparativos entre casos de uso e histórias de usuário

1	A história de usuário é uma breve descrição da funcionalidade através da visão do cliente; A história de usuário não é uma sequência de ações. Não é necessário modelar a interação entre o ator e o sistema, como sugerem os casos de uso.
2	A história de usuário é curta e consiste em uma ou duas frases escritas no idioma do usuário, focando na meta final da funcionalidade; Um caso de uso é mais pesado, mais rico em informações: objetivo, resumo, ator, pré-condição, evento de disparo, cenário de sucesso

	principal e fluxos alternativos e de erros.
3	Uma história de usuário é menor e, finalmente, pode ser vista como parte de um caso de uso específico: o cenário do fluxo principal ou uma extensão.
4	Histórias de usuário são usados para o planejamento, desempenham um papel importante no projeto durante a estimativa e planejamento. Os casos de uso não são utilizados para planejamento, apesar de poder usar a técnica "Pontos por caso de uso " para estimar o tamanho do projeto, não tem esse objetivo.
5	Histórias de usuário surgem mais rapidamente do que os casos de uso; Casos de uso necessitam de mais tempo para análise e escrita.
6	Histórias de usuário são mais legíveis do que os casos de uso; Casos de uso geralmente são maiores, muitas vezes, difíceis de ler, mesmo em um modelo estruturado.
7	As histórias de usuário são mais fáceis de manter do que um documento de 150 páginas com casos de uso.
8	As histórias de usuário baseiam-se em comunicação verbal e contam com a colaboração, discussão e proximidade para esclarecer detalhes; Caso de uso é um modelo textual associado com diagramas UML.
9	Histórias de usuário são, em teoria, escritas em cartões; Esses cartões não estão destinados a ser arquivados ao contrário dos casos de uso.
10	A história de usuário deve ser implementada e testada em uma iteração; Um caso de uso específico pode ser implementado em várias iterações.
11	Histórias de usuário podem ser facilmente escritas por um usuário ou cliente; A maioria dos casos de uso necessitam de mais tempo para serem escritos por usuários.
12	Histórias de usuário contém critérios de aceitação, escrito na parte de trás do cartão da história; Casos de testes são criados em um documento separado.

3.6 Otimizando a História de Usuário

No processo de análise de sistemas, ao levantar as informações referentes “ao que tem que ser feito”, são identificados os requisitos necessários para a completude do projeto. Estes requisitos identificados podem ter dentro de si, uma quantidade de outros requisitos implícitos. Pode haver também requisitos pequenos, que já permitam um entendimento e estimativa dos mesmos.

Estas discrepâncias entre os requisitos podem ser normalizadas com a técnica de INVEST proposta por Wake (2003). Mas normalizar não é a única responsabilidade do INVEST, ele auxilia a entender melhor os requisitos, a realizar estimativas mais acertadas, definir critérios de aceite, definir o valor agregado do requisito ao negócio e priorizar o que realmente deve ser entregue ao cliente. A INVEST é igual a:

- Independent - Independente
- Negotiable - Negociável
- Valuable - Avaliável
- Estimatable - Estimável
- Sized Appropriately - Dimensionada apropriadamente
- Testable – Testável

Deve-se buscar a independência entre as histórias de usuário, isto é, não ter correlação direta entre elas. Fica mais fácil realizar a estimativa focada a uma história do que realizar uma estimativa em um conjunto de histórias.

Uma história no seu processo de entendimento gera um conjunto de conversas, que são anotações realizadas na história para saber “o que tem que fazer” para deixar a história aderente às necessidades do cliente, isto faz parte do conceito de negociação.

Histórias devem gerar valor ao cliente, uma boa técnica de buscar este valor é o próprio cliente escrevendo as suas histórias. Nem sempre é possível o cliente escrevê-las, não importando no momento o que leva a esta impossibilidade.

No momento de realizar uma estimativa é importante que as histórias de usuário estejam independentes e negociadas. Se não existe mais dependência entre histórias a, estimativa vai ser mais eficiente e sem as funcionalidades desnecessárias para a entrega, o que melhora cada vez mais a previsão.

Não existe uma fórmula mágica para determinar o tamanho que as histórias de usuário devem ter, mas este tamanho se normaliza com o processo de levantamento de histórias e análise. Quanto menor a história, mais fácil de estimar.

Uma boa técnica de descobrir como a história de usuário deve se comportar, é realizar testes de validação. Conversas e testes de aceitação fecham um ciclo de entendimento da história. Qualquer implementação a mais, que não tenha sido identificada nas conversas ou testes não deve ser realizada na mesma iteração.

4 Trabalhos Relacionados

Uma série de estudos têm comparado Requisitos Tradicionais e Requisitos Ágeis. Em geral, identificam duas abordagens complementares. Paetsch, Eberlein e Maurer (2003) realizaram um estudo acerca deste assunto e concluíram que ambos têm objetivos em áreas-chave e que a principal diferença entre eles é a quantidade de documentação criada no projeto.

Eberlein e Leite (2002) apresentaram estudo que discutiu a aplicabilidade da engenharia de requisitos para processos ágeis. Argumentaram que quatro práticas (Interação com o Cliente, Análise de Requisitos Não-Funcionais e Gestão de Configuração e Mudança) devem ser adicionadas aos Requisitos Ágeis, a fim de garantir a qualidade do software produzido.

Meszaros (2004) escreveu um relato de experiência propondo quatro "*storyotypes*" (estereótipos da história), com base em casos de uso a serem usados como diretrizes para dividir grandes histórias de usuário. Como os casos de uso são amplamente utilizados e compreendidos como forma de levantamento de requisitos, Meszaros (2004) expôs sua preocupação de que os membros das equipes que tiveram experiência anterior com casos de uso, tivessem dificuldade em criar histórias de usuário. Esses desenvolvedores estão acostumados a trabalhar com modelos que podem ter muitos cenários e são mais propensos a criar uma grande história de usuário baseada apenas em um caso de uso.

Imaz e Benyon (1999) estudaram como histórias de usuário e casos de uso podem ser usados em conjunto para melhores interações durante o levantamento de requisitos. Concluíram que histórias de usuário são eficazes para a captura e interação, mas os casos de uso são necessários para a implementação quando a documentação formal é necessária. No entanto, não houve experimento controlado para comparar a utilização das histórias de usuário, como o descrito neste artigo.

5 Conclusão

Baseado na publicação de Kniberg (2008) entende-se que além da confirmação de que as histórias de usuário estão se tornando cada vez mais populares, muito provavelmente, continuarão a crescer até serem amplamente utilizados.

Histórias de usuário descrevem as necessidades do cliente. Quando uma história de usuário é escrita, o que o autor está descrevendo é puramente uma necessidade do usuário. É algo que o usuário precisa fazer no seu trabalho do dia-a-dia. Se nunca foi construído qualquer software para o cliente em questão, entende-se que essa necessidade ainda existe.

Os casos de uso descrevem o comportamento que será construído no software para atender tais necessidades. Um desenvolvedor que precisa construir um software deve ser capaz de ler um caso de uso e ter uma boa noção do que o software precisa fazer. Normalmente possui muitos detalhes, e descreve tudo o que o desenvolvedor precisa para construir, a fim de atender à necessidade do usuário. É por isso que precisa ter muito mais detalhes e possuir uma linguagem clara e inequívoca.

Apesar de existir um modelo mais tradicional proposto por Cohn (2004) que está sendo usado e ensinado, constata-se através das palavras de Kniberg que a evolução continua e, com ela, diversas mudanças em futuro próximo.

Os fabricantes de ferramentas CASE (*Computer-Aided Software Engineering*) tendem a produzir produtos que melhor correspondem à teoria, e que permitam criar histórias de usuário mais fáceis de escrever, com facilidade para manter e rastrear. As ferramentas já evoluíram, e evoluirão ainda mais, auxiliando a manter detalhes de UI (*User Interface*) fora da descrição dos cartões das histórias. Em breve, poder-se-á usar os diferentes níveis de formalidade para a escrita das histórias de usuário, através da geração automática baseada na descrição simples dos artefatos utilizados no desenvolvimento ágil.

Equipes distintas almejam quantidades diferentes de formalidade. Existem outros métodos para documentar as histórias de usuário que surgiram para responder à

diferentes necessidades. Em um extremo, uma organização pode exigir o uso formal que defina rigorosamente todos os comportamentos possíveis que o cliente deseja, somente para se aproximar do modelo de requisitos, enquanto no outro extremo, uma organização pode exigir histórias de usuário bastante simples. Qualquer dessas abordagens tende a sofrer mudanças. As organizações procurarão criar modelos alternativos que atendam os diversos tipos de necessidade e formalidade, para que os interessados tenham condições de escolher o modelo de acordo com o nível de detalhe consumado com suas necessidades.

Enquanto modelos são úteis, usuários têm desejo de preencher todas as caixas expostas em um cartão de história de uma vez. As pessoas continuarão a fazer mau uso deles, assim como já fizeram com os casos de uso e culparão a confusão resultante da forma em que as histórias de usuário foram concebidas. Algumas escolas ágeis já trabalham na mudança de comportamento, não deixando que o informal se aproxime do desenvolvimento irresponsável ou que volte a burocracia do modelo formal do processo unificado, embora se evidencia que essa iniciativa também tenha sido mal interpretada pelas pessoas que insistem em atacar ferramentas e processos ao invés do comportamento humano. Resta esperar que o modelo ágil, depois de tanta insistência, consiga mudar este comportamento.

Artigo recebido em junho de 2013 e aceito para publicação em abril de 2014

The Utilization of User Stories in Agile Requirements Survey

Abstract

From the popularity of the agile methods, there are discussions about what are user stories, why use user stories and how to describe the best way a system's behavior through the requirements engineering. The purpose of this article is to show the use of user stories are sufficient during the requirements gathering, approaching the different ways to describe, understand and use them today. To illustrate the stages of

development of the user story is presented a case study which aimed to apply techniques of this method compared the techniques of gathering requirements use case-driven.

Key Words: User Stories. Software Development

REFERÊNCIAS

Alexander, I. and Maiden, N., (2004) *Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle*, John Wiley.

Ambler, S., (2012) *Agile Requirements change Management*. Disponível em <http://agilemodeling.com/essays/changeManagement.htm>

Astels, D; Miller, G; Novak, M. (2002) *Extreme Programming Explained: Guia Prático*. Rio de Janeiro: Ed. Campus.

Beck, K. (2000) *Extreme Programming Explained,—Embrace Change*. Addison-Wesley.

Beck, K., Fowler, M., (2001) *Planning Extreme Programming*. Addison-Wesley. Cohn, M., *User Stories Applied: For Agile Software Development*, Addison-Wesley, 2004

Cockburn, A., (2000) *Writing Effective Use Cases*. Addison-Wesley.

Cockburn, A., (2002) *Agile Software Development*, Pearson Education.

Eberlein, A. and Sampaio do Prado Leite, J.C., *Agile Requirements Definition: A view from requirements engineering*, TCRe'02, 2002. Disponível em <http://www.enl.ucalgary.ca/tcre02/papers/Eberlein.pdf>

Fowler, M. and Highsmith, J., "The Agile Manifesto", *Software Development*, 2001, Vol. 9, pp. 28-32.

Grenning, J., (2011) *Test Driven Development for Embedded C*. Pragmatic Bookshelf. IEEE Computer Society, (1998) *IEEE Recommended Practice for Software Requirements Specifications*.

Imaz, M. and Benyon, D., How Stories Capture Interactions. INTERACT'99, 1999. Disponível em <http://www.dcs.gla.ac.uk/i99/programme.html>.

Jacobson, I. (1992) et al. Object-Oriented Software Engineering: A Use-Case Driven Approach, Addison-Wesley, Reading, MA.

Jeffries, R., “Essential XP: Card, Conversation, and Confirmation”, *XP Magazine*, 2001. Disponível em <http://xprogramming.com/articles/expcardconversationconfirmation/>

Meszaros, G., Using Storytypes to Split Bloated XP Stories, XP/Agile Universe Conference, 2004. Disponível em <http://www.informatik.uni-trier.de/~ley/db/conf/xpu/xpu2004.html>.

Microsoft, Team Foundation Service. Preview, 2013. Disponível em <http://tfs.visualstudio.com/>.

Paetsch, F., Eberlein, A. and Maurer, F., Requirements Engineering and Agile Software Development, WETICE 03 2003. Disponível em <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8713>.

Kniberg, H., *Scrum e XP Direto das Trincheiras*. InfoQ, 2008. Disponível em <http://www.infoq.com/br/minibooks/scrum-xp-from-the-trenches>.

Kruchten, P., (2003) The Rational Unified Process: An Introduction, Addison-Wesley.

Kulak, D. and Guiney, E., (2005) Use Cases - Requirements in Context Second Edition, Addison-Wesley.

Wake, B., *INVEST in Good Stories, and SMART Tasks*. XP123, 2003. Disponível em <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>.