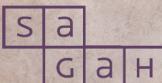


PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE

Clicéres Mack Dal Bianco



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS

Scrum e os requisitos

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Apresentar o processo de coleta visual de requisitos para o Scrum.
- Identificar os *stakeholders* e seus objetivos em um projeto Scrum.
- Descrever como coletar requisitos para *backlog* de produto.

Introdução

O sucesso de um projeto Scrum está atrelado às estratégias adotadas na coleta de requisitos. Nela, o foco está em identificar o que o cliente e os usuários realmente desejam e o que precisa ser implementado para satisfazer os usuários. Para tanto, os processos de documentação exigidos nos projetos tradicionais são resumidos, pois a prioridade é entender corretamente o que deve ser implementado. Além disso, uma das atividades mais críticas do projeto Scrum é identificar os *stakeholders* e seus objetivos. Por meio dos *stakeholders*, serão coletados requisitos para contemplar as necessidades dos usuários. Ainda referente às necessidades dos usuários, o *backlog* do produto desempenha um papel especial no Scrum. Todos os requisitos considerados necessários ou úteis para o produto estão listados no *backlog*. Você verá que ele pode ser comparado a um documento de requisitos incompleto e em constante mudança, contendo os requisitos necessários para o desenvolvimento.

Neste capítulo, você verá que, tanto para identificação dos *stakeholders* quanto para identificação dos requisitos do *backlog*, existem técnicas que podem ser empregadas nestas etapas, tais como Árvores e Florestas, SMART, mapeamento de histórias e histórias de usuários.

1 Coleta visual de requisitos para o Scrum

A coleta de requisitos é um processo utilizado para identificar quais são as necessidades do cliente. Posteriormente, define-se quais são as funcionalidades

e se estabelece prioridades no desenvolvimento de um projeto. Para melhor esclarecermos isso, vamos supor que João, um profissional de TI que atua como operador de suporte técnico, deseja se tornar um administrador de sistemas.

Nesse caso, sua primeira iniciativa é a obtenção de informações pertinentes ao cargo almejado, mais especificamente quanto aos requisitos necessários para se tornar um aspirante a administrador de sistemas. João reconhece que ainda precisa estudar e se qualificar. Então, João faz um levantamento das principais disciplinas e técnicas que deve dominar, direcionando seus estudos para os requisitos mais valorizados pelo mercado.

Nesse processo de coleta de requisitos todas as especificações do cliente são trazidas à mesa, e são estabelecidas as prioridades do projeto. Isto é, identifica-se os aspectos imprescindíveis do *software* a ser desenvolvido. E a importância disso se reflete nos benefícios de se conhecer mais profundamente as necessidades do cliente. O maior deles, sem dúvidas, é a compreensão do seu dia a dia, pois é isso o que permitirá à equipe desenvolver uma solução mais condizente com os problemas do cliente, otimizar o uso do tempo e reduzir as possíveis falhas.

A coleta de requisitos pode ser realizada por meio de entrevistas, questionários, *brainstorms*, prototipação, entre outras. No desenvolvimento tradicional o foco é a documentação. O desenvolvimento Scrum, por sua vez, objetiva atender às expectativas do cliente de maneira rápida documentado o mínimo. Pham e Pham (c2012a) apresentam uma estratégia de coleta visual de requisitos. Essa estratégia é apresentada a seguir.

Coleta visual

Em projetos ágeis você precisa ter uma boa maneira de identificar os requisitos. Uma forma de realizar a coleta de requisitos em projetos ágeis é por meio de histórias de usuários. Pham e Pham (c2012a) apresentam uma abordagem para auxiliar as equipes no processo de identificação das histórias de usuários para criar os *backlogs* de produtos. Dada a complexidade de outras técnicas, essa se apresenta como uma alternativa para facilitar esse processo. Pham e Pham (c2012a) sugerem que a coleta visual seja realizada em dois passos. No primeiro passo deve-se identificar os *stakeholders* e seus objetivos. No segundo passo deve-se coletar requisitos daqueles usuários que representam *stakeholders* e relacionar esses requisitos aos seus objetivos, para que sejam priorizados, usando para isso usando a analogia Árvores e a Floresta. A seguir você verá com detalhes esses dois passos.



Fique atento

Os *stakeholders* (partes interessadas, em português) são pessoas e organizações que podem ser afetadas por um projeto ou empresa, de forma direta ou indireta. Os *stakeholders* devem fazer parte do projeto de desenvolvimento de *software* e são importantes para o planejamento e a execução de um projeto. De acordo com Barbi (c2009–2010), considera-se *stakeholders* desde o patrocinador, os fornecedores, os membros da equipe de projeto, membros da diretoria da empresa até o público externo (usuários e vizinhos) que seja afetado pelo projeto.

Identificar os *stakeholders*

Os *stakeholders* são fundamentais em projetos Scrum, e identificá-los não é um processo simples. Para ter sucesso nesta tarefa é necessário fazer perguntas como: Quais são os objetivos ou metas de negócios? Como você mede a realização de metas? Por que você precisa deste projeto?

Uma maneira de identificá-los é utilizar a regra SMART, que é usada extensivamente para ajudar a definir objetivos. Essa técnica é um acrônimo que apresenta as seguintes características.

- Específico (*specific*): todos terão o mesmo entendimento do que são os objetivos.
- Mensurável (*measurable*): podemos determinar claramente se os objetivos foram alcançados.
- Alcançável (*achievable*): os *stakeholders* concordam sobre os objetivos.
- Realista (*realistic*): seremos capazes de realizar os objetivos do projeto com recursos que temos.
- Baseada em tempo (*time-based*): teremos tempo suficiente para realizar os objetivos

O Quadro 1 apresenta um exemplo de objetivos e métricas dos *stakeholders*.

Quadro 1. Gerenciar os *stakeholders*

Stakeholders	Objetivos	Métricas
Desenvolvimento do negócio	<ul style="list-style-type: none">■ Aumentar a parcela de mercado em 15% atingindo 200 mil clientes até o final do ano.	<ul style="list-style-type: none">■ Aumentar o número de usuários em 10 mil a cada mês.
Serviços ao cliente	<ul style="list-style-type: none">■ Reduzir as ligações dos clientes em 20% a cada trimestre.■ Reduzir o tempo de uma reclamação para 50% do tempo real.	<ul style="list-style-type: none">■ Números de chamadas.■ Tempo gasto ao telefone com o cliente com uma reclamação.

Fonte: Adaptado de Pham e Pham (c2012a).

Técnica Árvores e a Floresta

Para conhecer os *stakeholders* você se reunirá com os usuários que os representam, cada com seu papel e talvez com objetivos distintos. O dono do negócio (*product owner*), por exemplo, desejará escalabilidade e lucros, e o usuário (que é quem realmente usará) desejará telas amigáveis. Para entender as necessidades dos usuários e transformá-las em requisitos do produto de *software*, você pode utilizar a técnica Árvores e a Floresta, descrita por Pham e Pham (c2012a). A Figura 1 apresenta a estrutura desta técnica.



Figura 1. Técnica Árvores e a Floresta.

Fonte: Adaptada de Pham e Pham (c2012a).

Você deve iniciar pelo nível floresta, ou produto global. Verifique se o seu produto deveria ser composto — em outras palavras, quantas árvores devem existir na floresta, ou seja, quantos *stakeholders* (observe a Figura 2a). Em seguida, divida as árvores em seus galhos, a fim de agrupar as pessoas conforme funções ou seus objetivos com o produto, conforme ilustra a Figura 2b. Depois, relacione as folhas que são as histórias, para listar as prioridades e necessidades dos *stakeholders*.

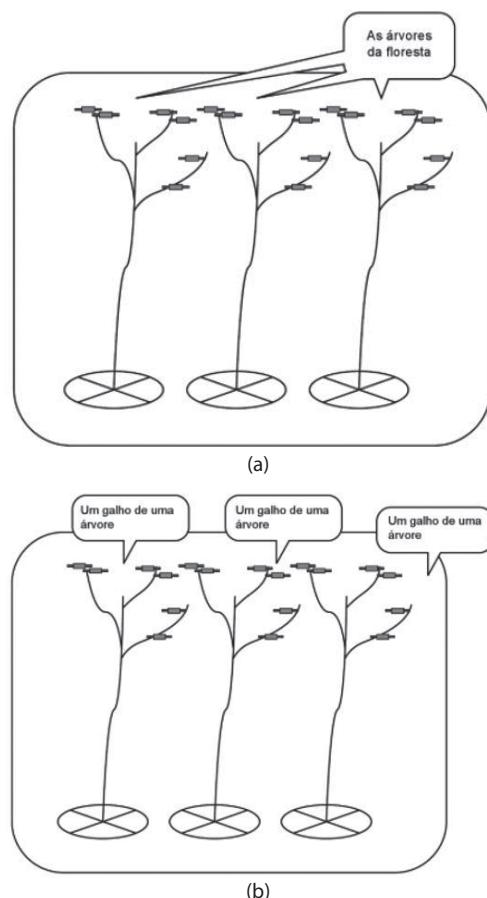


Figura 2. (a) Os stakeholders, representados de maneira análoga pelas árvores que compõem a floresta, e (b) as pessoas, representada por cada galho, que compõem um grupo.

Fonte: Adaptada de Pham e Pham (c2012a).

A técnica Árvores e a Floresta pode ser empregada para identificar os requisitos do *backlog* do produto da mesma forma como mostrado para os stakeholders. Você deverá buscar identificar quais são os requisitos chaves, que neste caso são as árvores, e para cada árvore busque localizar as categorias que estão relacionadas (galhos) e, posteriormente, liste as folhas, ou seja, os recursos que devem ser implementados para cada requisito. Um exemplo será apresentado na última seção deste capítulo.

Na próxima seção você verá detalhes sobre a identificação dos *stakeholders* e seus objetivos.

2 Stakeholders e seus objetivos

O conjunto *stakeholders* de um projeto engloba todas as pessoas que de alguma forma podem influir no sucesso do projeto. Cada projeto tem seu grupo de *stakeholders* próprio. A questão crítica é identificar todos os que podem influir (BARBI, c2010).

Dante disso, se você estiver criando um aplicativo ou serviço deve se preocupar em saber quem são os principais *stakeholders*. E então surgem as dúvidas: Como identificar quem são os *stakeholders* que estarão diretamente envolvidos? E se projetarmos um sistema que agrade os nossos *stakeholders* diretos e indiretos? E os patrocinadores do projeto? Se desenvolvemos um sistema que atenda às necessidades estaremos obtendo sucesso? E quais são os principais *stakeholders*?

Embora esteja claro que os usuários são realmente os principais interessados no projeto e que devemos, em grande parte, projetar considerando suas necessidades, deve-se ir além e analisar amplamente quem são os *stakeholders* do sistema proposto. Ao fazê-lo, é útil pensar em duas classes de *stakeholders*: *stakeholders* do sistema e *stakeholders* do projeto.

Conforme Leffingwell (2011), os *stakeholders* do sistema são os participantes interessados nele: uma parte interessada do sistema é qualquer pessoa que:

- usa diretamente o sistema;
- trabalha com os resultados de quem usa o sistema;
- será impactado pela implantação e operação de um sistema.

Essas partes interessadas incluem usuários e operadores, além de usuários que recebem relatórios e outras saídas do sistema; gerentes, compradores e administradores desses usuários; equipe de suporte e suporte técnico; desenvolvedores trabalhando em outros sistemas que integrar ou interagem com o sistema; profissionais de instalação e manutenção; e mais. Esses *stakeholders* do sistema serão os principais impulsionadores dos seus requisitos. Assim, para que o sistema seja bem-sucedido, é necessário considerá-los.

Os *stakeholders* do projeto têm interesse substancial e capacidade de investir no projeto que está desenvolvido. Incluem-se aí os patrocinadores do projeto, gerenciamento de projetos, gerenciamento de portfólio, executivos,

equipe de governança financeira e assim por diante. Neste caso, *stakeholder* do projeto é qualquer pessoa que:

- tem interesse no orçamento e no cronograma;
- tem interesse em compreender como o produto/sistema/solução será desenvolvido(a);
- estará envolvido no *marketing*, venda, instalação ou manutenção do sistema.

Nessa definição você pode ver que usuários diretos não são o único foco de um projeto. De fato, existe uma gama ainda maior de pessoas potencialmente afetadas por um novo sistema. Para ter sucesso, a equipe deve primeiro identificá-los, e então sintetizar seus objetivos em uma visão coesa.

A identificação e entendimento dos *stakeholders* pode proporcionar uma visão mais ampla dos processos em um negócio. A partir delas, é mais fácil para que os responsáveis pelo planejamento comprendam como a empresa pode melhorar, como processos podem ser otimizados e como garantir benefícios para todas as partes envolvidas.

Lyra, Gomes e Jacovine (2009) dividiram os interessados em internos e externos. Os *stakeholders* internos são aqueles que possuem alguma afiliação formal com a empresa. Ou seja, incluem: colaboradores, gestores, gerentes, proprietários e acionistas. Os *stakeholders* externos são aqueles que, apesar de diretamente afetados pela direção dos projetos da empresa em questão, não possuem afiliação com ela. Por exemplo: clientes, fornecedores, credores e investidores, Estado, ONGs, mídia e concorrentes.

Uma técnica para identificar os *stakeholders* é a árvores e florestas, que foi apresentada na seção anterior. Após serem identificados e classificados os *stakeholders*, também é importante entender quais os objetivos ou necessidades de cada um deles. Para entender os objetivos dos principais *stakeholders*, deve-se primeiro identificar aqueles que influenciam a organização.

Jemilo (2012) apresenta um *framework* para mapear os *stakeholders*, baseado nos seus objetivos ou interesses e no seu potencial para influenciar a cooperação ou não com o sistema. Analisando por esse viés, há quatro classes de *stakeholders*, segundo classificação de Savage *et al.* (1991 *apud* LYRA; GOMES; JACOVINE, 2009). A Figura 3a apresenta esse *framework*. No eixo y está a relação da influência ou poder. No eixo x está a relação de interesse ou objetivos com o projeto.

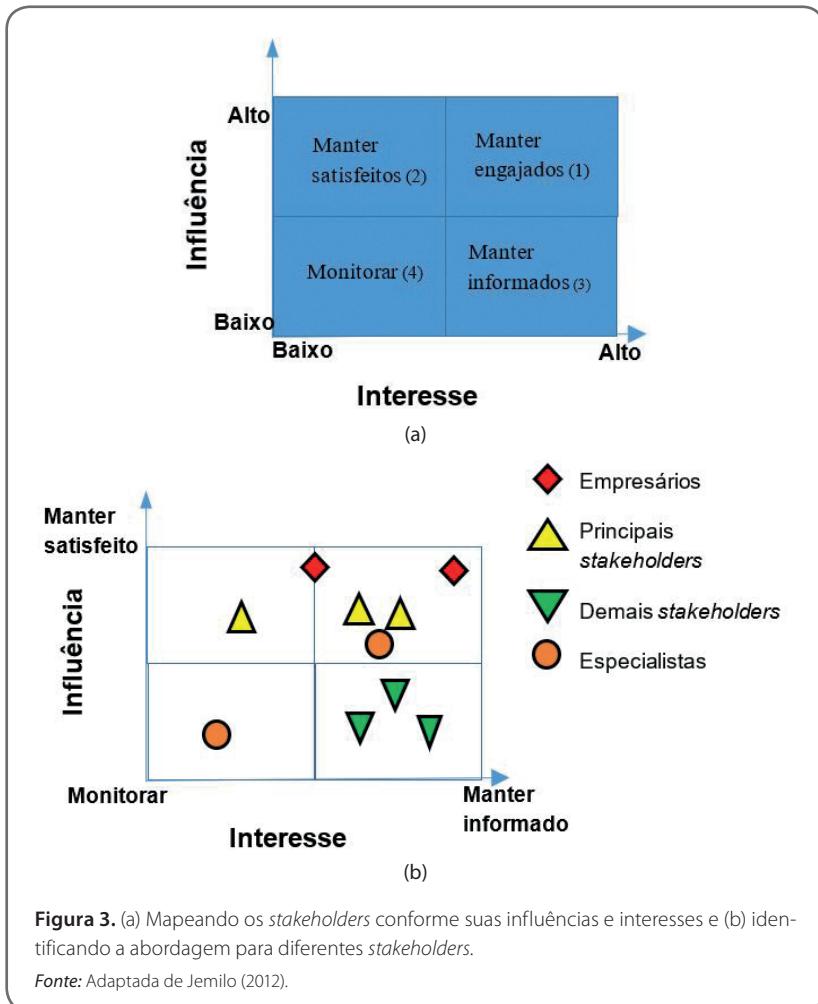


Figura 3. (a) Mapeando os stakeholders conforme suas influências e interesses e (b) identificando a abordagem para diferentes stakeholders.

Fonte: Adaptada de Jemilo (2012).

1. *Stakeholders* com alta influência e alto interesse — Podemos listar os donos do negócio e outras pessoas com autoridade para tomar decisões. São fáceis de identificar e darão sustentabilidade ao projeto, além de poderem cancelá-lo. Esses *stakeholders* são de fácil envolvimento; são todos aqueles que precisam ficar sabendo das decisões mais importantes e seus possíveis impactos. Nesses casos é importante mantê-los engajados.
2. *Stakeholders* com alta influência e baixo interesse — Neste grupo estão as pessoas com autoridade decisória significativa, mas falta de disponibilidade ou interesse para envolverem-se ativamente. Geralmente é difícil ter pontos de contato consistentes, sendo importante mantê-los satisfeitos.

3. *Stakeholders* com baixa influência e alto interesse — Podem ser impactados pelo projeto, mas têm pouca influência. Podem demandar mais do seu tempo do que você pode dar. Busque maneiras eficientes de se comunicar e mantenha-os informados sobre o projeto através de atualizações por *e-mail* e apresentações.
4. *Stakeholders* com baixa influência e baixa disponibilidade — Eles não estão (e não espere que estejam) significativamente envolvidos. Eles podem nem estar cientes do seu projeto e talvez não o querer. Saiba quem eles são; monitore-os, e talvez você consiga fazê-los mudar de quadrante.

A Figura 3 apresenta uma classificação observando a influência e o interesse dos diferentes *stakeholders*. De acordo com Jemilo (2012), os donos do negócio ou empresários e os principais *stakeholders* devem participar do Planejamento da Realize, dos *workshops* de *Adapt* e de seção de histórias de usuários (quando necessário) para opinarem, sugerirem, revisarem requisitos e, além disso, conhecerem o progresso do planejado *versus* o real.

Os principais *stakeholders* também devem estar envolvidos nas demonstrações das *Sprints*. Por sua vez, os demais *stakeholders* (normalmente são grupos menores e com menor influência) serão acionados conforme surgir a necessidade de mantê-los informados. Os especialistas no assunto são contratados conforme necessário para auxiliar nos dados de entrada.

Ainda de acordo com Jemilo (2012), é necessário priorizar os *stakeholders* conforme suas influências e interesses (objetivos). Além disso, deve-se avaliar o tempo que será despendido em cada *Sprint* e *Release* com os *stakeholders* certos. Segundo Pham e Pham (c2012a), a maior parte do seu tempo deve ser gasta trabalhando com os *stakeholders* da categoria (1), que possuem muita influência e interesse em seu projeto. Ao mesmo tempo você deve tentar fazer a categoria 2 se movimentar em direção à 1. E, em menor escala, fazer a categoria 4 se movimentar em direção à categoria 3.

Como não haverá terá tempo suficiente para conversar com todos os *stakeholders*. Sugere-se usar algum plano de gerenciamento de *stakeholders*, tentando atender seus objetivos maneiras diferentes, conforme a influência de cada um para o sucesso do seu projeto.

Vamos ver esse gerenciamento por meio de um exemplo. Carmem trabalha em uma empresa de TI e é responsável por gerenciar os *stakeholders*. Ela sabe quem são eles e quais são suas influências e seus interesses. Para tanto criou um plano de abordagem para os diferentes *stakeholders*, conforme você pode observar no Quadro 2.

Quadro 2. Gerenciar os *stakeholders*

Papéis	Responsáveis/ pessoas/grupos	Métodos de engajamento/ frequência
Dono do negócio/ empresários	Bruno Smith	<ul style="list-style-type: none"> ■ Reuniões individuais para discutir a visão, o roteiro e o recursos antes de cada reunião de planejamento. ■ Participação em <i>workshops</i> de requisitos (conforme necessário). ■ Participação no <i>workshop</i> de <i>Inspect & Adapt</i>. ■ Enviar <i>e-mail</i> quando o escopo do projeto está em risco.
Stakeholders principais	Carol Ollis	<ul style="list-style-type: none"> ■ Participação em <i>workshops</i> investigativos. ■ Apresentar a lista de <i>backlog</i> priorizada antes da reunião de liberação. ■ Participação no <i>workshop</i> de <i>Inspect & Adapt</i> (conforme necessário). ■ Participação na demonstração das <i>Sprints</i> do sistema. ■ Participação nas <i>Sprints</i> das equipes (opcional). ■ Comunicação por <i>e-mail</i> quando o escopo da <i>Sprint</i> ou do programa está em risco.
Demais <i>stakeholders</i>	Pedro Maulos	<ul style="list-style-type: none"> ■ Atualização por <i>e-mail</i> (conforme necessário). ■ Participação em <i>workshop</i> de requisitos (caso necessário).
Especialistas no assunto	Eloísa Natan	<ul style="list-style-type: none"> ■ Conduzir reuniões de <i>Sprints</i> (conforme necessário). ■ Participar de reuniões individuais ou em grupo (conforme demandas).

Fonte: Adaptado de Jemilo (2012).

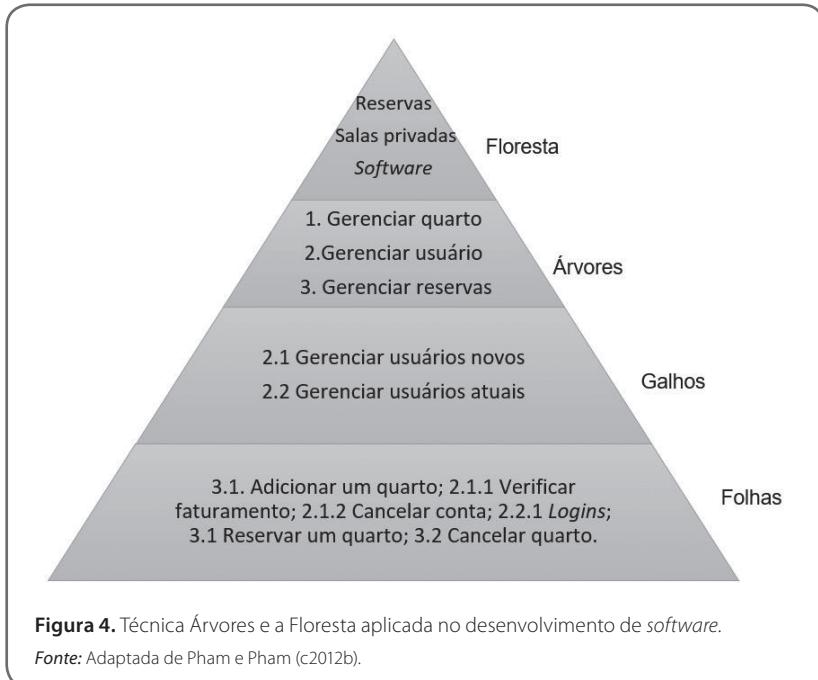
Essa técnica de mapeamento dos *stakeholders* pode orientar o tipo e a frequência de interações — conversas individuais, reuniões, convite para cerimônias, eventos e *workshops*.

A identificação dos *stakeholders*, os usuários responsáveis ou representativos, seus papéis e seus objetivos auxiliarão no entendimento das necessidades dos usuários e de como incluir essas necessidades por meio de requisitos no novo *software*. Na próxima seção será apresentado o *backlog* do produto, que contemplará todos os requisitos considerados necessários ou úteis ao produto.

3 Coleta de requisitos para o *backlog* de produto

O *backlog* do produto é uma lista de funcionalidades priorizada e desejada do produto. Essa lista fornece um entendimento centralizado e compartilhado do que implementar. De acordo com Rubin (c2012), o *backlog* é um artefato altamente visível, considerado o coração do Scrum, e que deve ser visível a todos os participantes do projeto.

A técnica Árvores e a Floresta é empregada como um recurso para auxiliar na identificação dos requisitos, sendo uma técnica visual e fácil de usar. A Figura 4 apresenta três árvores (ou áreas de interesses) para o novo produto – neste caso um *software* para reserva de quarto em um hotel. Mais especificadamente, as três árvores neste produto serão chamadas gerenciamento de usuários, gerenciamento de quartos e gerenciamento de reservas. Observe que a árvore gerenciamento de usuários tem dois galhos: “gerenciar novos usuários” e “referenciar usuários atuais”. Agora, analise as folhas dos respectivos galhos: “Verificar faturamento”, “Cancelar contas” e “Logins”. Finalmente, observe que visualmente é como o produto de *software* se parecerá com todas as folhas, todos os galhos e todas as árvores dentro da floresta.



Como dito anteriormente, o *backlog* do produto será composto por uma lista de requisitos que surgirão ao longo do desenvolvimento. Para identificar esses requisitos, são utilizadas as técnicas de mapeamento de histórias, histórias de usuários ou PBIs (itens do *backlog* do produto). De acordo com Rubin (c2012), as *features*, ou funcionalidades, normalmente são descritas por histórias de usuários. Inserir um recurso novo na tela de *login* para um *website*, ou alterar uma funcionalidade existente, tal como uma tela de login mais amigável, são exemplos de funcionalidades.

Contrariando as tradicionais coletas detalhadas de requisitos, o Scrum usa esse tempo para identificar os requisitos principais, que são discutidos e aprimorados progressivamente ao longo do projeto. Cada item da lista do *backlog* do produto representa um valor comercial desejável. Embora não exista um formato padrão para um PBI, o mapeamento de histórias foi proposto por Jeff Patton (2004), sendo uma ferramenta muito eficaz e útil para capturar requisitos, principalmente na fase inicial da coleta de requisitos.

De acordo com Parekh (2015), o mapeamento de histórias é uma atividade considerada envolvente. Possibilita um maior engajamento no processo de criação do *backlog* do produto. Uma parede pode servir de apoio para a

elaboração das histórias, em vez de escrever 100 páginas de uma macante documentação de requisitos.

O mapeamento de histórias é uma abordagem *top-down* de coleta de requisitos e é representado como uma árvore. Inicia com uma visão abrangente alcançada através de objetivos, que são definidos através da conclusão de atividades. E para concluir uma atividade, os usuários precisam definir tarefas. Essas tarefas podem ser transformadas em histórias de usuários para o desenvolvimento de *software*.

Estrutura do mapa de histórias: Objetivos – Atividades – Tarefas
– Histórias

Imagine um aplicativo de loja *on-line* e considere que o objetivo é localizar um produto. Podemos criar um ramo do mapa de história para entendê-lo melhor.

Localizar produto

Existem várias maneiras de o usuário atingir esse objetivo, tais como: navegar na árvore de categoria do produto, buscar texto livre no campo pesquisar ou em produtos em promoção. Para exemplificar, vamos demonstrar pela opção navegar na categoria de produto e criar o mapa de histórias.

Localizar produto

Navegar na
categoria Produtos

Agora, para encontrar o produto desejado, o usuário precisa executar determinadas tarefas; a relação dessas tarefas representa o mapa de histórias. Outro fato importante é que essas tarefas podem ser convertidas em histórias de usuários para o desenvolvimento de *software*. Veja a seguir a relação de tarefas (em amarelo).



Dessa maneira, pode-se detalhar cada ramo do mapa de histórias, começando pelas metas e construindo todo o mapa de histórias. De acordo com Parekh (2015), a construção de um mapa completo pode levar de 3 dias a 2 semanas, com base no tamanho e na complexidade do projeto. Em algum momento, precisamos de mais informações: por exemplo, é bom ter histórias, perguntas de acompanhamento, abordagens e alternativas. Isso é como enriquecer o mapa de história com mais informações. A seguir, algumas recomendações.

- Use cores diferentes para representar níveis diferentes no mapa da história, como laranja para objetivos, azul para recursos, verde para tarefas e amarelo para histórias.
- Use adesivos com pontos ou estrelas para representar notações especiais.
- Use pequenos adesivos para capturar notas, hipóteses, detalhes ou perguntas.
- História de usuários.

Outra técnica amplamente usada para criar a lista de *backlog* é a história de usuários. De acordo com Cohn (c2004), cada história descreverá um item ou uma necessidade para o usuário ou para o dono do negócio. Isso facilitará que o dono do produto priorize a lista do *backlog*, pois neste caso todos os itens estão apresentados em um formato comprehensível, facilitando a sua tomada de decisão — relembrando que no início da coleta não é importante identificar exaustivamente todos os requisitos, no entanto, quanto mais itens foram capturados no início mais eficiente será o desenvolvimento. Perceba que, definindo o papel dos *stakeholders* e focando nas histórias de usuário, o Scrum torna-se uma boa ferramenta de coleta de requisitos.

Apesar de não existir um formato específico de história de usuários, existem alguns termos considerados necessários. Veja uma estrutura clássica de história de usuário.



Exemplo

O formato típico é especificar o papel do usuário (a função do usuário), o que esse usuário deseja alcançar (a meta) e por que ele quer alcançá-la (o benefício).

Veja a estrutura de uma típica história de usuário para requisitos do *backlog*.

Como um <**PAPEL**> eu posso/gostaria/devolver <**FUNÇÃO**> para <**VALOR DE NEGÓCIO**>

Exemplos:

1. **Como um** cliente, eu **gostaria** de pagar usando meu cartão crédito para poder pagar minhas parcelas.

Pode-se incluir observações e, se necessário, detalhar as restrições: p. ex.: observações — aceitar Master, Visa e Amex; restrições — parcelar no máximo em 10 vezes; cliente não pode estar no SPC.

2. **Como um** vendedor, eu **gostaria** de consultar a produção semanal **para** ver se bateu a meta.

De acordo com Barbi (c2009–2010), ao descrever um requisito que é importante para um usuário ou para comprador de um sistema de *software*, deve-se considerar três aspectos:

1. uma breve descrição da história para servir como lembrete da funcionalidade;
2. conversações sobre as histórias para confirmar os detalhes escritos na descrição;
3. testes que podem ser usados para determinar quando uma história está completa.

As histórias de usuário, segundo Jeffries (2001), podem ser pensadas de uma maneira simples; mais especificadamente, a partir de três componentes, conhecidos como os três Cs: 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.

Épico

Uma grande história de usuário representa um conjunto de elementos que precisam ser analisados ou trabalhados. Estão em alto nível e devem ser decompostos para deixar clara a real necessidade. Pode ter a mesma estrutura ou outros formatos de história de usuário. Um requisito épico não pode ser realizado em uma *Sprint*, sendo necessário um trabalho de análise para decompor os épicos (JEMILO, 2012).



Exemplo

Exemplo de requisito ágil épico.

1. **Como um** vendedor, eu **gostaria** de poder visualizar minhas vendas **para** que possa analisar meu desempenho.

Perceba que visualizar minhas vendas é um requisito que precisa ser decomposto. Não está bem definido. Estamos, neste caso diante de um épico.

Uma visão abrangente do uso de histórias em projetos Scrum é apresentada por Leffingwell (c2011), conforme Figura 5. As funcionalidades ou recursos do produto (*features*) irão compor o *backlog* do produto. As funcionalidades são identificadas por meio de histórias (*story*). No momento do planejamento da *release* as funcionalidades podem ser decompostas em histórias. Essas funcionalidades são normalmente expressas com no máximo duas frases. O gerente do projeto juntamente com dono do produto irá definir a lista de *features*, ou funcionalidades, do produto. Cada funcionalidade fornece dados para a lista de *backlog* por meio das histórias de usuários (*story*).

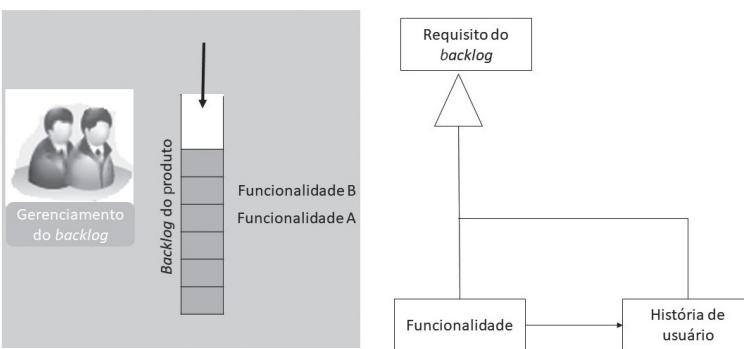


Figura 5. Funcionalidades são itens do backlog do produto.

Fonte: Leffingwell (c2011, p. 75).

Nem sempre, no entanto, deve-se utilizar histórias de usuários, especialmente em situações em que parece forçado, como a representação de certos defeitos.

Ao final deste capítulo você pode observar como é realizada a coleta visual de requisitos e que essa é considerada uma maneira simples de coleta ao ser comparada com outras técnicas existentes. A coleta visual pode ser empregada inicialmente para identificar os *stakeholders* e os requisitos do *backlog* do produto. Outros detalhes dos *stakeholders* foram apresentados para identificar seus objetivos e influências, possibilitando assim definir maneiras de inclui-los no projeto. Você também verificou como os requisitos do *backlog* podem ser coletados usando árvores e florestas, mapeamento de histórias e histórias de usuários.



Referências

BARBI, F. C. *Gestão de projeto*. c2010. Disponível em: <https://sites.google.com/a/gestaodeprojeto.info/www/analise-dos-stakeholders>. Acesso em: 26 jun. 2020.

COHN M. *User stories applied for agile software development*. Boston: Addison-Wesley, c2004.

JEFFRIES, R. *Essential XP*: card, conversation, confirmation. 2001. Disponível em: <https://ronjeffries.com/xprog/articles/expcardconversationconfirmation/>. Acesso em: 30 jun. 2020.

JEMILO, D. The stakeholders management framework: for teams, programs, and portfolios. [2012]. *Agile Alliance*, 2020. Disponível em: <https://agilealliance.org/wp-content/uploads/2016/01/Stakeholder-Management-by-Drew-Jemilo-Agile2012.pdf>. Acesso em: 26 jun. 2020.

LEFFINGWELL, D. *Agile software requirements*: lean requirements practices for teams, programs, and the enterprise. Upper Saddle River: Addison-Wesley, c2011.

LYRA, M. G.; GOMES, R. C.; JACOVINE, L. A. G. O papel dos stakeholders na sustentabilidade da empresa: contribuições para construção de um modelo e análise. *Revista da Administração Contemporânea*, v. 3, Edição Especial, p. 39–52, 2009. Disponível em: <http://www.spell.org.br/documentos/ver/8257/o-papel-dos-stakeholders-na-sustentabilidade-da-empresa--contribuicoes-para-construcao-de-um-modelo-de-analise>. Acesso em: 26 jun. 2020.

PAREKH, S. *Story mapping, visual way of building product backlog*. 2015. In: ThoughtWorks, c2020. Disponível em: <https://www.thoughtworks.com/insights/blog/story-mapping-visual-way-building-product-backlog>. Acesso em: 26 jun. 2020.

PATTON, J. *User story mapping*: building better products using agile software design. Sebastopol: O'Reilly Media, 2004.

PHAM, A.; PHAM, P. V. *Scrum em ação*: gerenciamento e desenvolvimento ágil de projetos de software. São Paulo: Novatec, c2012a.

PHAM, A.; PHAM, P. V. *Scrum in action*: agile software project management and development. Boston: Course Technology PTR, c2012b.

RUBIN, K. S. *Essential Scrum*: a practical guide to the most popular agile process. Upper Saddle River: Addison-Wesley, c2012.



Fique atento

Os *links* para sites da web fornecidos neste capítulo foram todos testados, e seu funcionamento foi comprovado no momento da publicação do material. No entanto, a rede é extremamente dinâmica; suas páginas estão constantemente mudando de local e conteúdo. Assim, os editores declaram não ter qualquer responsabilidade sobre qualidade, precisão ou integralidade das informações referidas em tais *links*.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS