

UNIVERSIDADE ESTADUAL DE
MATO GROSSO DO SUL

Computação, Licenciatura

William de Araujo Cadette

Aprendizagem de Scrum apoiada por dispositivos
móveis

UEMS
Outubro/2019

William de Araujo Cadette

Aprendizagem de Scrum apoiada por dispositivos móveis

Orientador: Prof. Me. Jorge Marques Prates

Monografia apresentada a Universidade Estadual de Mato Grosso do Sul – UEMS, para o Trabalho de Conclusão de Curso, como parte dos requisitos para obtenção do título de Licenciando em Computação.

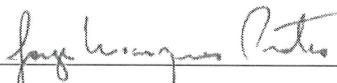
UEMS
Outubro/2019

William de Araujo Cadette

Aprendizagem de Scrum apoiada por dispositivos móveis

Monografia apresentada a Universidade Estadual de Mato Grosso do Sul – UEMS, para o Trabalho de Conclusão de Curso, como parte dos requisitos para obtenção do título de Licenciando em Computação.

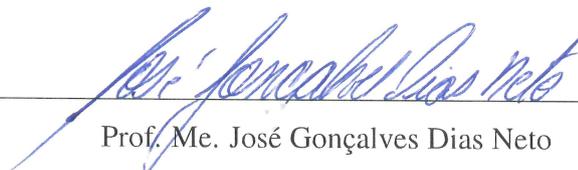
Membros da banca:



Prof. Me. Jorge Marques Prates (Orientador)
UEMS – Nova Andradina



Prof. Esp. André Castro Garcia
UEMS – Nova Andradina



Prof. Me. José Gonçalves Dias Neto
UEMS – Nova Andradina

Dedico este trabalho aos meus pais.

Agradecimentos

Agradeço a toda minha família por todo apoio e confiança.

Ao professor Jorge por todas orientações e ajuda.

A todos que me ajudaram direta e indiretamente para a conclusão deste trabalho.

E um agradecimento especial aos meus amigos, alunos que aceitaram me auxiliar com a validação e disponibilizaram seu tempo para usar meu aplicativo.

“Que a força esteja com você”

Resumo

Os métodos ágeis surgiram como uma resposta aos atrasos, orçamentos incorretos e projetos fracassados que os métodos tradicionais traziam. Estes métodos se mostram mais eficazes que seus antecessores em situações onde os requisitos não são completamente esclarecidos e possuem grandes chances de mudarem. O Scrum faz parte da família dos métodos ágeis, está ganhando muita popularidade, pois trata-se de um *framework* genérico que funciona com vários tipos de indústrias presentes atualmente. Tendo observado o grande potencial deste *framework*, a proposta deste trabalho foi a criação de um aplicativo móvel para auxiliar a quem quer estudar sobre o *framework* Scrum. A validação foi realizada com alguns alunos do curso de Computação, Licenciatura da UEMS – unidade de Nova Andradina, sendo que ao final foi aplicado um questionário para avaliação. Através dos resultados obtidos pelo questionário, pode-se ver o potencial da utilização do aplicativo para reforço dos conceitos de Scrum.

Palavras-chave: Aplicativo, Métodos Ágeis, Scrum.

Abstract

Agile methods emerged as a response to the delays, incorrect budgets, and failed projects that traditional methods brought. These methods are more effective than their predecessors in situations where requirements are not fully understood and are instable. Scrum is part of the family of agile methods, it's been gaining a lot of popularity because it's a generic framework that works with various types of industries present today. Remarking the great potential of this framework, the purpose of this work was to create a mobile application to help anyone who wants to study about the Scrum framework. The application was applied with some undergraduate students of the Computer, UEMS – Nova Andradina, at the end a questionnaire was applied for evaluation. Through the results obtained by the questionnaire, the potential of using the application to reinforce the concepts of Scrum can be observed.

Keywords: Application, Agile Methods, Scrum.

Lista de Figuras

2.1	Scrum <i>Framework</i> , extraído de Sabbagh (2014)	15
2.2	Quadro de tarefas Kanban, extraído de Hawks (2010)	22
2.3	Cartas de Planning Poker, extraído de Roger (2014)	23
3.1	Interface Principal do aplicativo	28
3.2	Interface de ajuda do aplicativo	29
3.3	Interface com todas as atividades do aplicativo	30
3.4	Interface das atividades abertas	30
3.5	Confirmação da alternativa selecionada	31
3.6	Gráfico das respostas da primeira questão do formulário	32
3.7	Gráfico das respostas da segunda questão do formulário	33
3.8	Gráfico das respostas da terceira questão do formulário	33
3.9	Gráfico das respostas da quarta questão do formulário	34
3.10	Gráfico das respostas da quinta questão do formulário	34
B.1	Exercício 26 <i>Representação do processo Scrum 1</i>	45
B.2	Exercício 29 <i>Representação do processo Scrum 2</i>	46

Lista de Tabelas

2.1	Comparação entre projetos feitos com métodos ágeis e o modelo cascata. Adaptado de StandishGroup (2015).	13
-----	--	----

Sumário

1	Introdução	10
1.1	Contexto	10
1.2	Motivação	11
1.3	Objetivos	11
1.4	Organização	11
2	Métodos Ágeis	12
2.1	Considerações Iniciais	12
2.2	O Manifesto Ágil	13
2.3	Scrum	14
2.3.1	Papéis do Scrum, o Time Scrum	16
2.3.2	Eventos Scrum	18
2.3.3	Artefatos Scrum	23
2.4	Trabalhos Relacionados	24
3	eScrum: Aplicação e Resultados	27
3.1	Considerações Iniciais	27
3.2	Aplicativo eScrum	28
3.3	Validação	32
3.4	Resultados	32
4	Conclusão	36
A	Formulário de avaliação do aplicativo eScrum	37
B	Banco de Questões	39
	Referências Bibliográficas	58

Introdução

1.1 Contexto

Muitos estudantes, por motivos diversos, não residem na cidade onde estudam, dispendo muito tempo com viagens para ir e voltar da universidade. Durante essas viagens, às vezes não é viável utilizar materiais de estudo como cadernos, livros ou *notebooks*. Simplesmente por não ser confortável ou por não conseguirem atingir o nível de concentração necessário para estudar nestes ambientes.

Visando fornecer algum objeto de estudo mais acessível nestes momentos, foi concebida a proposta deste trabalho: criar um aplicativo que pudesse ser usado para auxiliar o estudante com o conteúdo ao qual precisa estudar. Por ser compacto, não causaria tanto desconforto quanto os outros materiais, e com questões rápidas não exigiria muito da concentração.

O Scrum pertence à família dos métodos ágeis e possui um *framework* genérico, que promove a comunicação e o trabalho em equipe. Podendo assim ser usado tanto como objeto de estudos, como metodologia de ensino. Ao utilizar estes benefícios do Scrum junto da aprendizagem móvel, é possível fornecer aos alunos uma experiência de aprendizado única e eficiente.

Muitas vantagens são exploradas ao aderir à aprendizagem móvel, uma delas é que essa modalidade proporciona um ambiente de ensino versátil, prático e bastante convencional, visto que hoje em dia as pessoas passam bastante parte do tempo livre em seus dispositivos portáteis, principalmente *smartphones* (Barcelos et al., 2009).

Atribuir atividades complementares por meio de dispositivos móveis, é uma forma de gerenciar melhor esse tempo que as pessoas gastam em seus *smartphones*, explorando os conceitos do ensino à distância e das atividades extra classe.

1.2 Motivação

Um aplicativo pode facilmente alcançar e ajudar muitos estudantes em seus estudos, em diversas disciplinas, podendo ser modificado ao logo do tempo e conforme as suas necessidades. Nesta linha, ao desenvolver uma aplicação efetiva para um conteúdo específico, poderá esta funcionar para outros conteúdos, expandindo assim o número de alunos que poderão ser beneficiados.

Fornecer aos alunos uma ferramenta que possa os ajudar nos momentos mais complicados de realizar um estudo aprofundado, é uma forma de incentivá-los a se manterem em seus cursos. Afinal, se considerarmos que um dos fatores que pode gerar desistência é a falta de tempo para estudar, com o aplicativo eles teriam maior proveito de seu tempo de estudo.

Ensinar Scrum aos alunos é uma maneira de torná-los mais preparados para o mercado de trabalho. Sabendo que o Scrum funciona em diversas áreas e se mostra eficaz, quando bem executado, podendo os ajudar a se destacar dentre os outros, adaptando os conceitos aprendidos aos diversos problemas que surgirão no decorrer de suas carreiras, seja no ensino, na indústria, ou em qualquer outro meio de produção em que esse método possa ser utilizado.

1.3 Objetivos

Este trabalho tem como objetivo geral investigar o efeito a utilização de um aplicativo para auxiliar no reforço de conhecimento, e como objetivos específicos:

- Ensinar sobre o *framework* Scrum;
- Construir uma base de questões de múltipla escolha sobre o Scrum;
- Desenvolver um aplicativo;
- Realizar a validação do aplicativo;
- Analisar os resultados da validação aplicando um formulário aos participantes.

1.4 Organização

No Capítulo 2 é descrito uma contextualização sobre os métodos ágeis, com principal foco no Scrum e seus conceitos, que foi o método escolhido a ser abordado no aplicativo. Neste capítulo também podemos ver alguns trabalhos que utilizam o *framework* do Scrum na teoria e na prática.

No Capítulo 3 é apresentado sobre o aplicativo gerado e também suas funcionalidades. Ainda neste capítulo também são apresentados os resultados obtidos através da validação realizada com alguns alunos do curso de Computação, Licenciatura da UEMS – Nova Andradina. Por fim no Capítulo 4 são apresentadas as considerações finais deste trabalho.

Métodos Ágeis

2.1 Considerações Iniciais

O primeiro modelo de desenvolvimento que surgiu foi o Cascata, revolucionando o desenvolvimento de *software* na época, mas era intrinsecamente necessário que os requisitos fossem meticulosamente detalhados e documentados, já que uma vez iniciada a implementação mudanças não eram viáveis. Então modelos iterativos e incrementais surgiram baseados no cascata e sua proposta era de acomodar melhor as mudanças, contudo ainda era necessária a densa extração de requisitos e documentação detalhada antes de iniciar cada iteração (Cohen et al., 2004).

As abordagens tradicionais precisam antecipar uma grande quantidade de requisitos para poderem realizar um projeto de maneira eficaz, pois quando o projeto é iniciado mudanças nesses requisitos não são facilmente aceitas. O problema é que na maioria dos projetos as mudanças acontecem durante o andamento do projeto e precisam ser aceitas ou o produto final não sairá como o cliente espera (Abbas et al., 2008).

Logo, os métodos de desenvolvimento tradicionais eram insuficientes para a exigência do mercado, principalmente pelo fato de precisarem ser preditivas e muito bem planejadas antes de iniciar o projeto. Em alguns casos era possível realizar a extração completa dos requisitos, mas na maioria não, porque as mudanças no projeto acontecem e a equipe de desenvolvimento não consegue ter controle sobre este fato (Highsmith e Cockburn, 2001).

Mudanças eram necessárias e, embora os métodos tradicionais dificilmente saíssem da moda, era óbvio que nem sempre funcionavam como previsto em todas as situações. Profissionais reconheceram que novas práticas eram necessárias para melhor lidar com as mudanças nos requisitos. E essas novas práticas devem ser orientadas para as pessoas e flexíveis, oferecendo “regras geradoras” sobre “regras inclusivas” que se desintegram rapidamente em um ambiente dinâmico (Highsmith e Cockburn, 2001).

No início da década de 90 um novo conjunto de processos de desenvolvimento de *software* destacam-se dos já existentes e começam a ganhar força entre as empresas de desenvolvimento, na época esses processos ficaram conhecidos como “leves”, posteriormente conhecidos como métodos ágeis, e o objetivo deles era substituir os métodos existentes por serem muito ineficazes em sua aplicação resultando em produtos que quase nunca saiam como o esperado pelos clientes.

Em uma pesquisa realizada pela [StandishGroup \(2015\)](#), sobre mais de dez mil projetos de *software* dentro do período de 2011 a 2015, segmentados entre métodos ágeis e cascata, foram comparados mostrando que os métodos ágeis possuem uma maior taxa de sucesso e reduzem consideravelmente as chances de falha dos projetos, como mostrado na Tabela 2.1

Método	Sucesso	Desafio	Falha
Ágil	39%	52%	9%
Cascata	11%	60%	29%

Tabela 2.1: Comparação entre projetos feitos com métodos ágeis e o modelo cascata. Adaptado de [StandishGroup \(2015\)](#).

Não existe uma abordagem perfeita que atenda a todas as demandas necessárias na hora de produzir um *software*, cada uma tem suas características e enfrentam os problemas de forma diferente. É necessário então saber ser flexível na hora de iniciar um projeto e sempre buscar utilizar uma metodologia que mais se adequa ao serviço ([Glass, 2001](#)).

2.2 O Manifesto Ágil

Nos dias 11 e 13 de fevereiro de 2001 dezessete representantes dos métodos ágeis se reuniram na estação de esqui em Utah para uma reunião que tinha como objetivo investigar a relação entre seus processos de desenvolvimento. O resultado dessa reunião foi o manifesto ágil de desenvolvimento de *software* com um conjunto de valores e princípios.

O manifesto diz o seguinte [Beck et al. \(2001\)](#):

Estamos descobrindo maneiras melhores de desenvolver *software*, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

- **Indivíduos e interações** mais que processos e ferramentas;
- **Software em funcionamento** mais que documentação abrangente;
- **Colaboração com o cliente** mais que negociação de contratos;
- **Responder a mudanças** mais que seguir um plano.

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda. O manifesto também apresenta doze princípios a serem seguidos:

- Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de *software* de valor;
- Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas;
- Entregar *software* funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos;
- Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto;
- Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho;
- O método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara;
- *Software* funcional é a medida primária de progresso;
- Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes;
- Contínua atenção à excelência técnica e bom design, aumenta a agilidade;
- Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito;
- As melhores arquiteturas, requisitos e designs emergem de times auto organizáveis;
- Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

Métodos ágeis não são a solução para os problemas do desenvolvimento de *software* e também não podem substituir de vez os métodos tradicionais já que cada um tem suas vantagens, entretanto, devido ao seu foco nos participantes do projeto, se bem gerenciados, ajudam bastante para o fator de sucesso dos projetos.

2.3 Scrum

O Scrum é uma das metodologias ágeis mais usadas, é adaptativo, iterativo, flexível e eficiente. Ele evidencia a comunicação e garante um ambiente responsável, favorecendo um melhoramento contínuo do projeto e também da própria equipe envolvida. Seu *framework* pode ser empregado em todos os tipos de indústria e qualquer tipo de projeto ([Schwaber e Sutherland, 2017](#), [SCRUMstudy, 2016](#), [Wazlawick, 2013](#)).

Na Figura 2.1 é mostrado o *framework* do Scrum, onde, através da reunião de Sprint Planning, alguns itens são extraídos do Product Backlog para formar o Sprint Backlog e então iniciar o evento principal que é a Sprint. Diariamente são realizadas reuniões denominadas Daily Scrum e ao final da Sprint mais duas reuniões são realizadas, a Sprint Review e Sprint Retrospective, objetivando a apresentação, avaliação e melhora do trabalho realizado pela equipe de desenvolvimento, concluindo com um incremento utilizável do produto. Então o processo é realizado para todas as Sprints a serem realizadas até a conclusão do projeto.

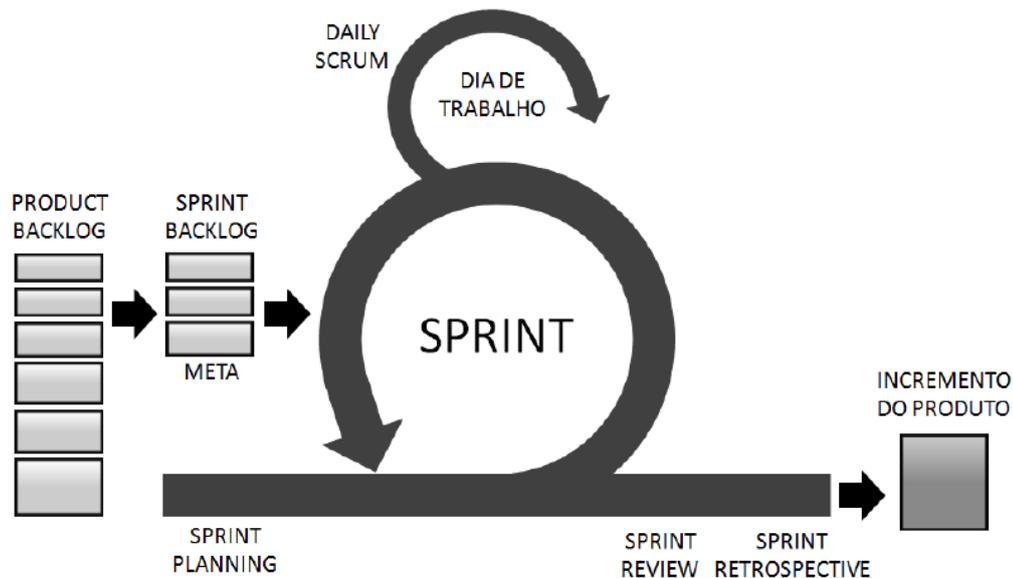


Figura 2.1: Scrum *Framework*, extraído de Sabbagh (2014)

De acordo com Sabbagh (2014), Scrum é um *framework* Ágil, simples e leve, utilizado para a gestão do desenvolvimento de produtos complexos imersos em ambientes complexos. Scrum é fundamentado no empirismo e utiliza uma abordagem iterativa e incremental para entregar valor com frequência e, assim, reduzir os riscos do projeto.

O Scrum consiste em uma equipe necessariamente pequena, com 5 a 11 envolvidos, para facilitar a interação entre os indivíduos, onde gerenciam a construção do projeto em iterações denominadas Sprints com uma documentação mínima feita especificamente para cada Sprint. Diversas reuniões são realizadas ao longo do projeto para melhorar ainda mais a comunicação, revisar os objetivos do projeto e reduzir os riscos ao máximo (Schwaber e Sutherland, 2017).

Segundo Schwaber e Sutherland (2017), o Scrum possui três pilares que apoiam no processo empírico:

- **Tranparência:** Os resultados devem mostrar pontos importantes do processo com os padrões especificados gerando o mesmo entendimento por todos que vejam;
- **Inspeção:** O time deve frequentemente inspecionar o trabalho que está sendo realizado para detectar possíveis imprevistos. Não deve ser feito com frequência que possa atrapalhar o objetivo do trabalho;

- **Adaptação:** Se o trabalho a ser desenvolvido desviar do objetivo a ponto do resultado se tornar inaceitável, o material já produzido precisa ser corrigido o quanto antes.

O Scrum determina quatro eventos para aumentar a eficiência da inspeção e adaptação: Sprint planning, Daily Scrum, Sprint Review e Sprint Retrospective.

O scrum possui também seus próprios valores, além dos valores do manifesto ágil, que segundo [ScrumAlliance \(2019\)](#), são:

- **Foco:** Os membros do Time Scrum devem se concentrar em uma tarefa de cada vez, trabalhando em equipe para entregar itens valiosos mais cedo;
- **Coragem:** Com o trabalho em equipe os envolvidos se sentem mais apoiados, deixando mais recursos disponíveis. Isso gera mais coragem para enfrentar maiores desafios;
- **Transparência:** Enquanto trabalhamos juntos, expressamos como estamos fazendo, o que está em nosso caminho e nossas preocupações para que possam ser abordados;
- **Comprometimento:** Quando temos mais controle sobre o que fazemos aumentamos o comprometimento com o sucesso;
- **Respeito:** À medida que trabalhamos juntos, compartilhando sucessos e fracassos, passamos a respeitar uns aos outros e a nos ajudar mutuamente a sermos dignos de respeito.

2.3.1 Papéis do Scrum, o Time Scrum

Cada membro tem sua função e importância para o bom funcionamento do *framework* Scrum, um time deve obrigatoriamente possuir um Scrum Master, um Product Owner e o time de desenvolvimento. Todos devem ser capazes de exercer seu papel e precisam estar em sintonia, tanto entre si quanto com o cliente.

2.3.1.1 Scrum Master

O Scrum Master tem a função de facilitador, ele quem é o responsável por ensinar as práticas do Scrum, remover restrições encontradas pelo time e garantir que o *framework* Scrum esteja coerente, fornecendo assim um ambiente favorável para que o projeto seja concluído corretamente.

Segundo [SCRUMstudy \(2016\)](#), o Scrum Master não é como um gerente de projeto, pois ele não pode ser considerado como um líder de projeto, ele está no mesmo nível que os outros membros do time e age como facilitador. Qualquer um do time que tenha as habilidades de facilitar o projeto pode ser denominado o Scrum Master.

O Scrum Master serve ao Product Owner de várias maneiras, incluindo ([Schwaber e Sutherland, 2017](#)):

- Garantindo que objetivos, escopo e domínio do produto sejam entendidos o melhor possível por todos do Time Scrum;

- Encontrando técnicas para o gerenciamento efetivo do Product Backlog;
- Ajudando o Time Scrum a entender as necessidades para manter os itens do Product Backlog claros e concisos;
- Compreendendo o planejamento do Produto em um ambiente empírico;
- Garantindo que o Product Owner saiba como organizar o Product Backlog para maximizar valor;
- Compreender e praticar a agilidade;
- Facilitando os eventos Scrum, conforme exigidos ou necessários.

O Scrum Master serve o Time de Desenvolvimento de várias maneiras, incluindo ([Schwaber e Sutherland, 2017](#)):

- Treinando o Time de Desenvolvimento em autogerenciamento e interdisciplinaridade;
- Ajudando o Time de Desenvolvimento na criação de produtos de alto valor;
- Removendo impedimentos para o progresso do Time de Desenvolvimento;
- Facilitando os eventos Scrum conforme exigidos ou necessários;
- Treinando o Time de Desenvolvimento em ambientes organizacionais nos quais o Scrum não é totalmente adotado e compreendido.

2.3.1.2 Product Owner

É o responsável por deixar claro para a equipe o objetivo do projeto em cada Sprint a ser realizada, para isso precisa manter o Product Backlog atualizado assegurando comunicação frequente com todos os envolvidos no projeto certificando de que este não fuja do esperado. É a única pessoa responsável pelo Product Backlog e algumas de suas tarefas são ([Schwaber e Sutherland, 2017](#)):

- Expressar claramente os itens do Product Backlog;
- Ordenar os itens do Product Backlog para alcançar melhor as metas e missões;
- Otimizar o valor do trabalho que o Time de Desenvolvimento realiza;
- Garantir que o Product Backlog seja visível, transparente, claro para todos, e mostrar o que o Time Scrum vai trabalhar a seguir;
- Garantir que o Time de Desenvolvimento entenda os itens do Product Backlog no nível necessário.

2.3.1.3 Time de Desenvolvimento

Um pequeno grupo multidisciplinar, responsável por realizar o desenvolvimento do produto, de acordo com as prioridades definidas pelo Product Owner gerando um incremento do produto utilizável a cada Sprint. É um time auto gerenciável, determinando como o produto será desenvolvido, planejando e acompanhando seu progresso. Para isso possui autonomia em suas decisões, tornando-se responsável pelos resultados atingidos (Sabbagh, 2014).

Algumas das características que o Time de Desenvolvimento possui são (Schwaber e Sutherland, 2017, Sabbagh, 2014):

- São auto organizados, ninguém diz ao time como transformar o Product Backlog em incrementos funcionais;
- São multifuncionais, possuindo as habilidades necessárias para criar o incremento do produto;
- Interagem com o Product Owner para esclarecer dúvidas ou solicitar decisões referentes ao produto e contribuem para o refinamento e aprimoramento do Product Backlog para a próxima Sprint;
- Recorrem ao Scrum Master ao surgirem impedimentos que dificultem seu trabalho sempre que possível;
- Cada membro pode possuir uma habilidade específica e pode ser especializado em uma determinada área, entretanto as responsabilidades do trabalho pertencem ao time como um todo;
- Obtém *feedback* do cliente e partes interessadas ao apresentar os resultados no final da Sprint;
- São orientados a excelência técnica, sempre se aperfeiçoando e procurando realizar seu trabalho com qualidade e consciência.

Apesar de cada membro do Time Scrum ter sua função específica no projeto, todos estão no mesmo nível hierárquico, logo todos têm total responsabilidade sobre o projeto geral e devem se empenhar em seguir as propriedades definidas pelo Scrum e serem competentes para que o produto seja realizado como o cliente espera e tenha boa qualidade.

2.3.2 Eventos Scrum

Alguns eventos prescritivos são estabelecidos para criar uma regularidade e minimizar a necessidade de reuniões não definidas no Scrum. Todos os eventos têm um tempo máximo definido e a partir do momento que a Sprint se inicia esse tempo deve ser cumprido sem alterações, mas sempre que o objetivo é alcançado o evento pode ser finalizado e o próximo se inicia (Schwaber e Sutherland, 2017).

2.3.2.1 Sprint

É o principal evento do Scrum, onde todos os requisitos do Sprint Backlog são transformados em um incremento funcional do produto. O ideal é que cada Sprint tenha duração de no máximo um mês, quando um prazo maior é estipulado a definição do que será construído pode mudar aumentando a complexidade e o risco de construção da funcionalidade a ser implementada (Schwaber e Sutherland, 2017).

Além do trabalho de desenvolvimento propriamente dito, ocorrem durante a Sprint a reunião de Sprint Planning, as reuniões de Daily Scrum, a reunião de Sprint Review, a reunião de Sprint Retrospective e quaisquer outras atividades ou reuniões realizadas com a participação do Time de Desenvolvimento. Todos esses eventos fazem parte do tempo estipulado para a Sprint (Sabbagh, 2014).

A Sprint poderá ser cancelada se o objetivo se tornar obsoleto. Isto pode ocorrer se a organização mudar sua direção ou se as condições do mercado ou das tecnologias mudarem. Geralmente a Sprint deve ser cancelada se ela não faz mais sentido às dadas circunstâncias. O cancelamento pode ser anunciado apenas pelo Product Owner e são incomuns, pois consome muitos recursos e todo trabalho precisa ser revisado e novamente planejado (Schwaber e Sutherland, 2017).

2.3.2.2 Sprint Planning

Um planejamento realizado antes de cada Sprint onde o Time Scrum define qual será o trabalho a ser realizado durante a Sprint. A duração do planejamento depende do tamanho definido para a Sprint, e deve durar no máximo oito horas para uma Sprint de um mês (Schwaber e Sutherland, 2017).

Nessa reunião o Time de Desenvolvimento e o Product Owner, facilitados pelo Scrum Master, selecionam os requisitos do Product Backlog a serem implementados na Sprint, nos quais julgam ser capazes de desenvolver na duração da Sprint originando o Sprint Backlog (Sabbagh, 2014).

É importante que os requisitos do Product Backlog estejam preparados para que a reunião seja eficiente e produtiva. Requisitos que não possuem detalhes suficientes, por exemplo, podem colocar toda a Sprint em risco (Sabbagh, 2014).

2.3.2.3 Daily Scrum

Uma reunião de 15 minutos realizada todos os dias durante o andamento da Sprint. Por ser breve o ideal é que seja feita em pé e frente a um quadro de anotações. Nela os membros do time relatam o andamento do projeto, respondendo a três perguntas (SCRUMstudy, 2016, Wazlawick, 2013, Audy, 2015):

1. O que eu fiz ontem que ajudou a atingir a meta da Sprint?
2. O que eu farei hoje para ajudar atingir a meta da Sprint?

3. Existe algum obstáculo me impedindo de atingir a meta da Sprint?

Reuniões Diárias melhoram a comunicação, eliminam outras reuniões, identificam e removem impedimentos para o desenvolvimento, destacam e promovem rápidas tomadas de decisão, e melhoram o nível de conhecimento do Time Scrum, por isso deve ser orientada pelo Scrum Master para que seja realizada todos os dias e no tempo correto, porém é de responsabilidade do Time de Desenvolvimento organizar e conduzir a reunião (Schwaber e Sutherland, 2017).

2.3.2.4 Sprint Review

Reunião com a duração máxima de quatro horas para uma Sprint de um mês, em que os resultados são apresentados pelo Time de Desenvolvimento ao Product Owner. Nesta reunião o Product Owner avalia se a meta foi ou não atingida e faz anotações que poderão se tornar em novos itens para o Product Backlog (Victorino e Schiessl, 2018).

É uma reunião informal com foco no incremento do produto gerado. Acontece no último dia da Sprint sendo obrigatórias as presenças do Time de Desenvolvimento, Product Owner e Scrum Master. Os demais convidados podem ser clientes, usuários, gerentes e outros aos quais as funcionalidades desenvolvidas na Sprint são relevantes e cujo *feedback* é importante (Sabbagh, 2014).

2.3.2.5 Sprint Retrospective

Reunião com a duração máxima de três horas para uma Sprint de um mês. Mediada pelo Scrum Master, ela se propõe a avaliar lições aprendidas ao longo da Sprint, o que foi bom na última Sprint e o que deve ser melhorado (Victorino e Schiessl, 2018).

Audy (2015), diz que para a reunião ser bem aproveitada deve ser construtiva e com foco no futuro, sem buscar culpados para os problemas ocorridos. O espaço de todos deve ser respeitado, ninguém precisa ser obrigado a falar. O tempo e a maturidade no método permitirão que a equipe se sinta mais à vontade para se abrir e falar as claras sobre suas dificuldades, acertos e erros.

O propósito da Sprint Retrospective é (Schwaber e Sutherland, 2017):

- Inspeccionar como a última Sprint foi em relação às pessoas, aos relacionamentos, aos processos e às ferramentas;
- Identificar e ordenar os principais itens que foram bem e as potenciais melhorias;
- Criar um plano para implementar melhorias no modo que o Time Scrum faz seu trabalho;

O Scrum permite a inserção de algumas técnicas e métodos que não estão listados em seu *framework*. E algumas dessas técnicas são utilizadas a tanto tempo junto ao Scrum e funcionam tão eficientemente que a maioria das equipes já adotam o Scrum junto dessas técnicas. Entre as mais comuns podemos citar:

2.3.2.6 Kanban

O Kanban foi idealizado e implementado com sucesso na indústria, mas sua gestão visual é utilizada em desenvolvimento de *software*, em áreas de negócio, corporativas, serviços e onde mais houver interação entre pessoas que tenham seu trabalho ou ações de alguma forma inter-relacionadas. Alguns objetivos e características mostram porque a maioria das áreas estão utilizando a gestão visual em seus processos de produção e serviços (Audy, 2015):

- Minimizar inventário e estoques de insumos, inacabados e acabados;
- Minimizar a movimentação dos materiais em processamento;
- Reduzir o tempo total de produção (lead time);
- Evitar gargalos ou carências no fluxo entre etapas do processo;
- Descentralização e auto-organização no controle de produção e estoque;
- Oferecer melhor tempo de resposta a mudanças na demanda;
- Execução em pequenos lotes, tanto quanto possível;
- Manter uma efetiva gestão visual em todas as etapas do processo;
- Garantir a cadência e o controle puxado do sistema, sem interrupções.

A Figura 2.2 demonstra o funcionamento de um quadro de tarefas básico com quatro colunas, sendo a primeira para identificar a funcionalidade a ser desenvolvida, em seguida a funcionalidade é dividida em tarefas menores e ficam dispostas na coluna de tarefas a fazer, a terceira e quarta coluna representam, respectivamente, as tarefas que estão sendo trabalhadas pelo time e as tarefas concluídas.

2.3.2.7 Planning Poker

O Planning Poker usa o conceito de complexidade. Para exercitar corretamente este conceito, existem baralhos semelhantes a série de Fibonacci (1, 2, 3, 5, 8, 13, 21), sempre somando os dois últimos números para chegar ao seguinte. A sequência representa a complexidade da funcionalidade a ser avaliada, quanto maior o valor, maior a complexidade (Audy, 2015).

É uma dinâmica coletiva, onde todos podem e devem se posicionar após bom entendimento do que é cada funcionalidade. Ao estimar, é necessário comparar com as estimativas anteriores, mantendo-se a coerência e refazendo quando necessário. As regras para o Planning Poker são (Audy, 2015):

- Cada integrante do Time de Desenvolvimento recebe um baralho de Planning Poker;
- É importante haver uma combinação do número a partir do qual devemos tentar quebrar em mais funcionalidades;

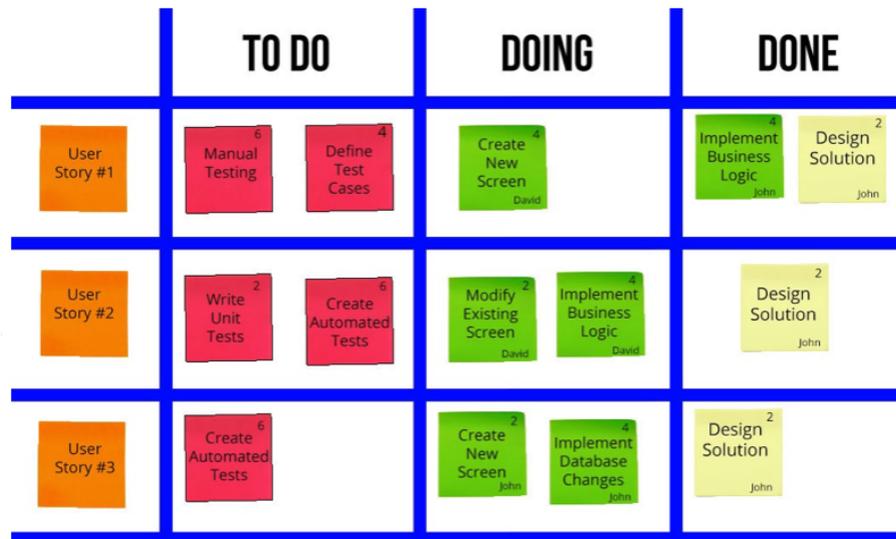


Figura 2.2: Quadro de tarefas Kanban, extraído de Hawks (2010)

- O time seleciona a menor funcionalidade a ser estimada como 1 para servir de referência no início;
- A partir desta funcionalidade mensurada como 1, vamos estimando as maiores, comparativamente;
- Uma funcionalidade é escolhida, o Product Owner pergunta se todos entenderam e pede para que escolham uma carta;
- Quando todos escolheram a sua carta, pede-se que mostrem para os demais;
- Se houver consenso, a funcionalidade está estimada e vamos adiante;
- Se não houver consenso todos devem justificar sua opção;
- Após os argumentos, roda-se mais uma vez;
- O limite hipotético é de 3 rodadas, quando então a estimativa segue a carta mais alta.

O Planning Poker é bastante usado para estimar a prioridade dos requisitos mais importantes que ficarão no início do Product Backlog.

Um baralho bastante convencional é apresentado na Figura 2.3, onde valores acima de 20 são considerados muito altos e a funcionalidade será muito custosa, portanto precisará ser reformulada e se preciso dividida em mais funcionalidades, algumas cartas especiais são apresentadas como o zero, significa que a funcionalidade é muito pequena e provavelmente irrelevante para ser considerada uma funcionalidade individual, e a interrogação, indica que o participante não tem o conhecimento necessário para estimar a complexidade da funcionalidade.

Os baralhos são bastante variados podendo apresentar valores diferentes e até algumas cartas especiais além destas, mas a dinâmica do jogo deve ser sempre a mesma independente do baralho usado ou da equipe que está a avaliar as funcionalidades.

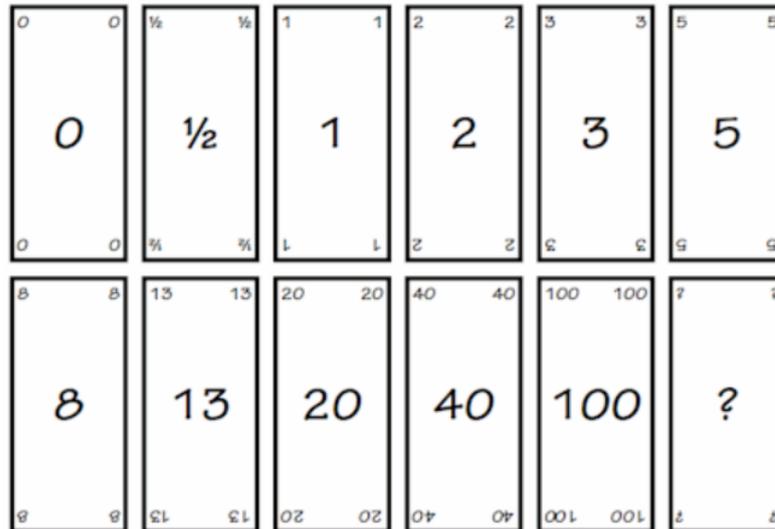


Figura 2.3: Cartas de Planning Poker, extraído de [Roger \(2014\)](#)

Cada evento é realizado para manter transparência e inspeção criteriosa permitindo que a adaptação seja realizada da forma mais natural e simples possível. Falhas em qualquer um destes eventos decorrerá em uma redução na transparência e consequentemente na redução da inspeção e adaptação reduzindo o valor e qualidade do produto ([Schwaber e Sutherland, 2017](#)).

2.3.3 Artefatos Scrum

De acordo com [Schwaber e Sutherland \(2017\)](#), os artefatos do Scrum representam o trabalho ou o valor para o fornecimento de transparência e oportunidades para inspeção e adaptação. Os artefatos definidos para o Scrum são especificamente projetados para maximizar a transparência das informações chave de modo que todos tenham o mesmo entendimento dos artefatos.

2.3.3.1 Product Backlog

É uma lista ordenada de tudo que é necessário para desenvolver o produto e de total responsabilidade do Product Owner, sendo este o único autorizado a realizar todas as mudanças necessárias que surgem durante o projeto. Os itens do Product Backlog devem possuir uma descrição, ordem, estimativa e valor ([Schwaber e Sutherland, 2017](#), [Sabbagh, 2014](#)).

[Schwaber e Sutherland \(2017\)](#), dizem que enquanto um produto é usado e ganha valor, e o mercado fornece feedback, o Product Backlog torna-se uma lista maior e mais completa. Requisitos nunca param de mudar, então o Product Backlog é um artefato vivo. Mudanças nos requisitos de negócio, condições de mercado ou tecnologia podem causar mudanças no Product Backlog.

Os primeiros itens descritos no Product Backlog devem ser mais claros e detalhados que os

itens mais abaixo. Estimativas mais precisas são feitas baseadas em maior clareza e detalhamento (Schwaber e Sutherland, 2017, Sabbagh, 2014).

2.3.3.2 Sprint Backlog

É uma lista com os requisitos mais relevantes do Product Backlog que serão desenvolvidos na determinada Sprint.

O Sprint Backlog é modificado pelo Time de Desenvolvimento ao longo da Sprint, com isso o Sprint Backlog vai ganhando detalhes e sendo incrementado conforme o Time de Desenvolvimento descobre a melhor forma de atingir o objetivo da Sprint (Schwaber e Sutherland, 2017).

Em qualquer ponto do tempo na Sprint, o total do trabalho remanescente dos itens do Sprint Backlog pode ser somado. O Time de Desenvolvimento monitora o total do trabalho restante pelo menos a cada Reunião Diária para projetar a probabilidade de alcançar o objetivo da Sprint. Ao acompanhar o trabalho restante ao longo de toda a Sprint, o Time de Desenvolvimento pode gerenciar o seu progresso (Schwaber e Sutherland, 2017).

2.3.3.3 Incremento

O incremento é a soma de todos os itens do Product Backlog completados durante a Sprint e o valor dos incrementos de todas as Sprints anteriores. Ao final da Sprint um novo incremento deve estar “Pronto”, o que significa que deve estar na condição de ser utilizado e atender a definição de “Pronto” do Time Scrum. Um incremento é uma parte principal inspecionável de trabalho pronto que suporta empirismo no final da Sprint. O incremento é um passo na direção de uma visão ou de um objetivo (Schwaber e Sutherland, 2017).

2.3.3.4 Definição de Pronto

A Definição de Pronto é um acordo formal entre Product Owner e Time de Desenvolvimento sobre o que é necessário para se considerar que um trabalho realizado na Sprint está “pronto”. São critérios definidos por ambos para garantir a transparência, por meio da compreensão compartilhada do que significa quando o Time de Desenvolvimento afirma que qualquer item ou o Incremento do Produto está “pronto” (Sabbagh, 2014).

Uma Definição de Pronto ideal estabelece que o resultado do trabalho de um Time de Desenvolvimento em uma Sprint seja entregável. Ou seja, idealmente nenhum desenvolvimento, teste, integração ou quaisquer tarefas adicionais devem ser necessárias para que o Incremento do produto produzido na Sprint possa ser entregue aos clientes do projeto. Entregar ou não ao final da Sprint, no entanto, é uma decisão de negócios e, mesmo que já seja possível, cabe ao Product Owner decidir quando o fazer (Sabbagh, 2014).

2.4 Trabalhos Relacionados

Pérez et al. (2018) realizaram um trabalho na disciplina de engenharia de *software* na Universidade de Castilla-La Mancha (UCLM), na qual os alunos foram divididos em três grupos,

então os conceitos dos métodos ágeis e Scrum foram explicados em comparação com os modelos tradicionais. Os alunos foram avaliados através de dois questionários, um com questões de alternativa sobre o Scrum e outro com questões para classificação, onde os alunos diziam se a opção seria referente ao Scrum ou ao desenvolvimento tradicional. Como auxílio, a plataforma moodle foi utilizada para disponibilizar os materiais aos alunos e também como meio para eles tirarem suas dúvidas. Em comparação dos resultados o segundo questionário obteve melhores pontuações entre os alunos.

Em sua proposta [Scharff et al. \(2012\)](#) apresentam um projeto global de desenvolvimento de software (GSD) para o desenvolvimento Android através do *framework* do Scrum, que precisou ser modificado para atuar no desenvolvimento distribuído dos participantes. Ao todo, 11 estudantes participaram formando duas equipes, algumas ferramentas como moodle, IBM RTC e Skype foram utilizadas para melhorar a comunicação, e manter um ambiente com o desenvolvimento atualizado e disponível para os integrantes das equipes. Em geral, os estudantes precisaram passar por treinamentos para conseguirem utilizar as ferramentas pois não conheciam algumas delas. Eles possuíam experiência em desenvolvimento, mas não em desenvolvimento distribuído, devido a este fato tiveram dificuldades, entretanto foi relatado que as ferramentas ajudaram bastante e a experiência dos estudantes foi muito boa e proveitosa.

[Devedzic e Milenkovic \(2010\)](#) levantam um estudo sobre a maneira que os métodos ágeis são trabalhados entre os diferentes níveis de cursos (graduação e pós-graduação). Notaram que os estudantes que aprendem sobre os métodos ágeis na prática conseguem ter um melhor aprendizado em relação aos que ficam apenas em teorias. Dizem também que as experiências de ensino e aprendizagem variam de acordo com o nível do curso, já que nos cursos de pós-graduação os estudantes geralmente possuem uma experiência prática em desenvolvimento de software.

[Schafer \(2017\)](#) propõe um projeto para construção do campus de sua universidade e um centro histórico da cidade em tamanho real através do Minecraft com o gerenciamento do Scrum, utilizando também conceitos do Scrum of Scrums. O projeto foi dividido em duas etapas, uma para construção do campus e outra para o centro histórico. Ao todo participaram oito equipes e os projetos foram divididos em oito partes, sendo uma para cada equipe. Como os estudantes precisavam realizar trabalhos extraclasse foi utilizada a plataforma Moodle com um repositório, um chat e fórum para cada equipe. Em geral os estudantes conseguiram cumprir os prazos e disseram que gostaram bastante da experiência e tiveram um conhecimento prático do trabalho com Scrum, tendo ainda um bom conhecimento sobre gerenciamento de projetos.

Em seu artigo [Rico e Sayani \(2009\)](#) relatam uma alteração no currículo do curso de engenharia de software de sua universidade. Nele explicam que alguns estudantes precisaram produzir um website utilizando os métodos ágeis para a conclusão do curso. Três equipes foram criadas, duas delas escolheram utilizar o Scrum, a outra escolheu o método *Release Planning*. Ambas as equipes tinham experiência com programação, porém tiveram uma resistência inicial por não conhecerem os métodos ágeis e tampouco terem utilizado, já que estavam habituados aos

métodos tradicionais. Em geral conseguiram superar os desafios e obtiveram um desempenho melhor que com os métodos tradicionais.

Wagh (2012) fez o uso do Scrum para trabalhar a prática e ensinar os conceitos de forma indireta e sem a necessidade de alterar o currículo do curso. Para tal o *framework* do Scrum precisou ser alterado, deixando os prazos de entregas mais folgados para não prejudicar os estudantes com as outras disciplinas. Várias ferramentas e métodos extras foram utilizados para aprimorar o desempenho dos alunos e facilitar a aplicação do projeto. Por fim, os alunos precisaram preencher um formulário para obter um *feedback* do projeto, grande maioria respondeu que obteve uma melhor experiência com gerenciamento de projetos e trabalho em equipe.

Em seu curso de engenharia de software, oferecido na Universidade Federal de São Carlos – SP (UFSCar), é necessário o desenvolvimento de um projeto para finalizar o curso, este projeto era realizado com metodologias tradicionais e muitos problemas eram enfrentados. Em reação às essas dificuldades, Zorzo et al. (2013) propuseram o uso de métodos ágeis, especificamente o Scrum, com objetivo de iniciar as atividades de desenvolvimento mais cedo e reduzir os projetos incompletos no fim do prazo. O conteúdo do curso foi reorganizado para seguir o progresso do projeto, iniciando com conteúdos mais simples e avançando gradativamente com constantes entregas para seguir as práticas do Scrum. Em geral, alguns problemas foram encontrados e os estudantes ficaram um pouco resilientes no início, mas no decorrer conseguiram se adaptar e aproveitar bem a experiência. Os resultados finais foram satisfatórios visto que todos os grupos conseguiram entregar o projeto completo no final do curso, e através da prática de alguns dos princípios ágeis obtiveram um melhor aprendizado de seu funcionamento.

Fernanda et al. (2018) desenvolveram uma adaptação dos valores e princípios do manifesto ágil, assim como do *framework* Scrum, a fim de conseguir uma melhor experiência prática em sala de aula com essas metodologias. A aplicação do modelo proposto contou com a participação de 250 estudantes, 6 professores e durou 3 semestres. Por fim notou-se resultados positivos no processo de ensino–aprendizagem, tais como: organização ideal dos alunos e do trabalho do professor, aquisição de competências e aprendizagem dos alunos, motivação e aumento do interesse na participação de atividades acadêmicas, maior desempenho acadêmico em disciplinas, entre outros.

Com a preocupação de melhorar o aprendizado dos estudantes nas aulas e ao mesmo tempo prepará-los para o mercado de trabalho, Madeja e Biñas (2018) realizam uma pesquisa para saber qual o método ágil de desenvolvimento e sistema operacional são mais disseminados no mercado de trabalho. De acordo com os dados levantados, conseguem constatar que o Scrum e o Android são mais adotados em relação às outras metodologias e sistemas. Visando deixar os estudantes mais familiarizados e preparados para o mercado de trabalho, foi criado um curso de Desenvolvimento de Aplicações para Dispositivos Inteligentes com foco na plataforma Android estruturando as aulas do curso com os conceitos do Scrum. Os resultados do projeto foram bastante satisfatórios, facilitando o aprendizado dos estudantes no desenvolvimento Android e no gerenciamento de projetos ágeis.

eScrum: Aplicação e Resultados

3.1 Considerações Iniciais

O aplicativo foi desenvolvido especificamente para ajudar a quem pretende reforçar os conhecimentos em Scrum. Todas as atividades disponibilizadas nele foram retiradas de provas de vários concursos públicos que acontecem pelo país. As provas originais podem ser encontradas no site¹.

Vale realçar que as questões foram pesquisadas em diversos canais, como livros, sites, artigos, todas as provas do POSCOMP e ENADE até o ano atual, outras provas de conhecimentos gerais na área de computação e as provas de concursos públicos. Entretanto o único canal que havia questões sobre o Scrum eram as provas de concursos públicos.

Todas as questões inseridas no aplicativo estão dispostas no Apêndice B deste documento. Foram pesquisadas todas as provas oficiais e seus respectivos gabaritos, para garantir que nenhuma questão e alternativa fosse inserida incorretamente.

O aplicativo foi desenvolvido na IDE do Ionic 5², que trabalha com HTML+CSS e Javascript, facilitando o desenvolvimento. A plataforma desenvolve aplicações híbridas, e é muito vantajosa na hora de programar e ajustar a interface, porém como consequência é muito mais complexo ao configurar o *hardware* dos *smartphones* (*bluetooth*, sensores, câmera, leitores, entre outros), em relação às IDEs que desenvolvem aplicações nativas.

A validação do aplicativo foi realizada com alunos do 3º e 4º ano do curso de Computação da UEMS – Nova Andradina. Foram escolhidas estas turmas dado que já haviam estudado o Scrum e o aplicativo não funcionaria efetivamente com alunos que não possuíam conhecimento algum sobre a disciplina, pois não disponibiliza os conceitos, mas apenas exercícios sobre o

¹<https://www.gabaritou.com.br/>

²<https://ionicframework.com/>

Scrum.

3.2 Aplicativo eScrum

O aplicativo eScrum foi desenvolvido para ser uma ferramenta de auxílio a quem estiver estudando Scrum, reforçando seus conceitos, e com alta disponibilidade. Podendo assim rever o conteúdo quando não estiver com seu material disponível, afinal é mais fácil andar com um *smartphone* que livros e cadernos.

Durante seu desenvolvimento, algumas dificuldades foram encontradas, devido a versão da IDE trabalhada ser recém lançada. Os materiais disponíveis na internet eram de versões antigas e não eram totalmente compatíveis com a utilizada, ou não eram tão claros, o que ocasionou diversos contratempos e atrasos.

Em suma o aplicativo possui uma interface inicial, uma interface de ajuda, outra com todos os exercícios a serem selecionados, também uma para cada exercício específico e um *feedback* para suas alternativas, indicando se a alternativa escolhida foi correta ou incorreta.

Na interface principal, ilustrada na Figura 3.1, podemos ver o logotipo do aplicativo e os botões “Iniciar” e “Como Funciona” que direcionam, respectivamente, à interface dos exercícios, ilustrada na Figura 3.3 e à interface de ajuda, ilustrada na Figura 3.2.

15:29 0.04 K/s 28%



INICIAR

COMO FUNCIONA

Figura 3.1: Interface Principal do aplicativo

Na Interface de ajuda temos uma breve descrição do aplicativo e em seguida uma explicação

de como o mesmo funciona, fornecendo assim, uma maneira do usuário esclarecer suas dúvidas e não ficar perdido ou receoso ao navegar pelo aplicativo e suas funcionalidades.



Figura 3.2: Interface de ajuda do aplicativo

Ao clicar no botão iniciar da Interface principal, o usuário é direcionado à interface dos exercícios, onde estão concentrados todos os exercícios dispostos em uma matriz, fornecendo ao usuário visão da maior quantidade de exercícios possíveis, onde ele pode escolher qual exercício deseja fazer com o mínimo de interação possível dentro da interface.

Ao selecionar um exercício, a interface do exercício será aberta [3.4](#), mostrando seu conteúdo. Dentre os exercícios podemos encontrar atividades de verdadeiro ou falso, certo e errado e completar as frases, todas com respostas de múltipla escolha, para tornar mais rápido a quem estiver usando o aplicativo.

E por fim temos o *feedback* da alternativa escolhida, para isso simplesmente é mostrado uma interface de aviso, como na [Figura 3.5](#), indicando se a alternativa escolhida é a correta ou não.

Devido aos contratempos e atrasos enfrentados durante o desenvolvimento, algumas funcionalidades ficaram de fora nesta versão do aplicativo, mesmo assim ele consegue servir como auxiliar, reforçando o conhecimento de quem quer estudar a disciplina a qual ele aborda.



Figura 3.3: Interface com todas as atividades do aplicativo

15:46 0.13 K/s 29%

← Exercício 01

Os métodos ágeis são fundamentados no desenvolvimento e entrega funcional tendo em vista atender aos requisitos dos clientes. Eles agregam um conjunto de princípios provenientes do manifesto ágil, tais como:

- envolvimento do cliente;
- entrega incremental;
- pessoas não processos;
- aceitação das mudanças;
- manutenção da simplicidade.

I- O Scrum adota a entrega incremental por meio de Sprints.

II- O Scrum adota a simplicidade por meio do uso da programação em pares.

III- O Scrum adota o envolvimento do cliente com a priorização e a negociação dos requisitos na concepção de Sprints.

É correto o que se afirmar em

II, apenas.

III, apenas.

I e II, apenas.

I e III, apenas.

I, II e III.

15:30 0.17 K/s 28%

← Exercício 26

A figura abaixo representa uma visão geral de uma metodologia ágil.

Os itens de I a IV representam as reuniões comumente realizadas nesta metodologia, que são, respectivamente,

Master Meeting - Product Owner Meeting - Time Meeting - Daily Meeting.

de Retrospectiva - de Planejamento - Diária - de Revisão.

de Planejamento - Diária - de Retrospectiva - de Revisão.

de Planejamento - de Retrospectiva - de Revisão - Diária.

Product Owner Meeting - Master Meeting - Daily Meeting - Time Meeting.

15:47 0.18 K/s 29%

← Exercício 45

Na reunião de planejamento do Sprint Backlog, se o Product Owner afirmar que todos os requisitos do produto foram identificados, é correto concluir que o Backlog do Produto está completo, visto que este é uma lista ordenada de todos os requisitos necessários para o desenvolvimento do produto.

Certo.

Errado.

Figura 3.4: Interface das atividades abertas

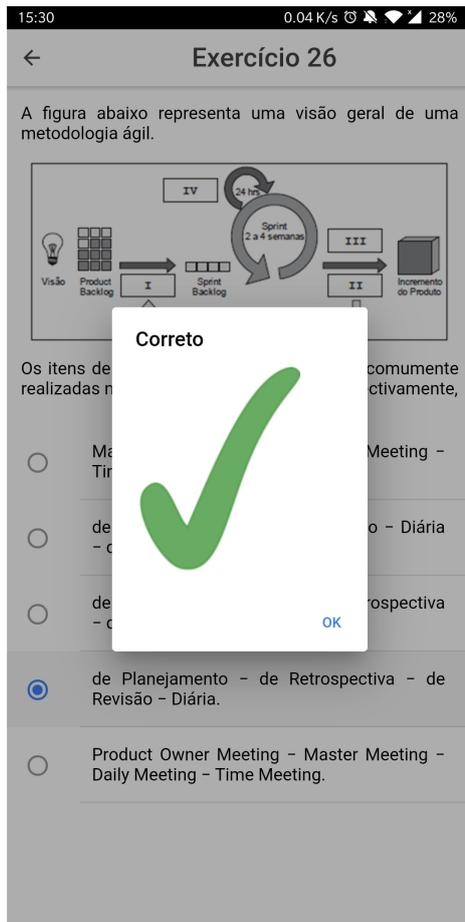


Figura 3.5: Confirmação da alternativa selecionada

3.3 Validação

Os testes do aplicativo foram realizados com os alunos do 3º e 4º ano do curso de Computação, Licenciatura da UEMS – Nova Andradina. Ao todo 4 alunos estavam aptos e se dispuseram a participar. Todos receberam um link³ para realizarem o *download*, instalar e utilizar o aplicativo durante 5 dias, tempo máximo disponível para a realização da aplicação e conclusão do trabalho.

Para realizar a instalação era necessário que eles tivessem acesso a um *smartphone* com sistema Android. Não houve a necessidade de realizar treinamentos, os alunos apenas foram instruídos a utilizar o aplicativo ao menos uma vez por dia. Entretanto os participantes era livres para escolher a maneira que fosse melhor para cada um, tentando não afetar negativamente os testes e nem forçar qualquer tipo de resultado.

Durante a validação não surgiu nenhum tipo de problema nem reclamações dos participantes. Alguns deles buscavam uma melhor compreensão da alternativa correta, pois em alguns casos haviam dúvidas nas solução do exercício. Porém nada relacionado à falhas, invalidades e dúvidas no funcionamento do aplicativo.

Após o prazo estipulado foi aplicado um formulário com 6 questões, apresentado no Apêndice A deste documento, onde as cinco primeiras questões são referentes a eficácia do aplicativo e a última sobre melhorias no mesmo. As respostas serão apresentadas e discutidas na próxima seção.

3.4 Resultados

A seguir serão apresentados os resultados das respostas dos alunos para cada uma das 6 questões dispostas no formulário disponível no Apêndice A.

1 – Em uma escala de 0 a 5, qual era seu nível de conhecimento sobre o Scrum antes de utilizar o aplicativo?

Esta questão foi inserida com intuito de saber qual era o conhecimento inicial dos participantes, de acordo com suas próprias perspectivas. Através das respostas, podemos notar na Figura 3.6 que 25% dos alunos possuíam um conhecimento muito bom em Scrum, 50% possuíam um bom conhecimento, e 25% tinham conhecimento intermediário sobre o conteúdo.

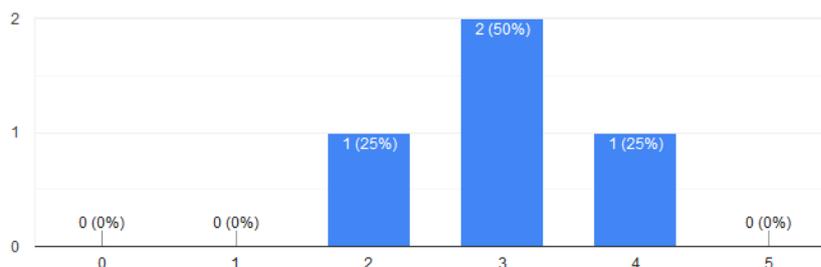


Figura 3.6: Gráfico das respostas da primeira questão do formulário

³<https://drive.google.com/open?id=148Btl1sophks-SkdDkPKet0KunN-WQaW>

2 – Quantas vezes ao dia você abriu o aplicativo?

Esta pergunta tinha o propósito em saber qual a frequência de uso que os alunos participantes mantiveram durante os 5 dias que o utilizaram. De acordo com os resultados apresentados na Figura 3.7 podemos notar que houve um resultado muito bom, pois todos usavam o aplicativo mais de uma vez ao dia, indicando que o objetivo da frequência de uso foi atingido.

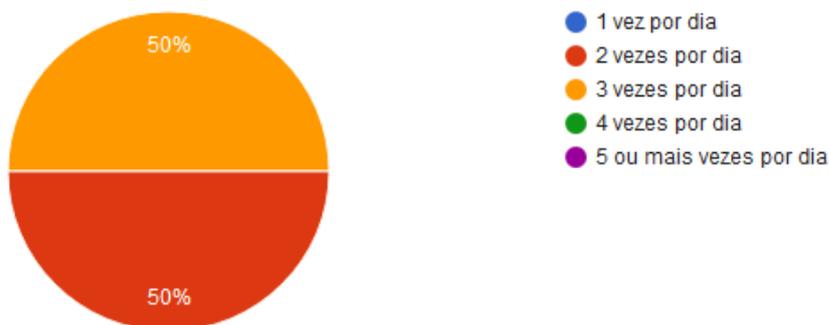


Figura 3.7: Gráfico das respostas da segunda questão do formulário

3 – Em média, quanto tempo você gastou no aplicativo por dia?

Esta questão foi inserida para descobrir se o principal propósito do aplicativo foi atingido, que seria fornecer atividades rápidas para auxiliar no estudo. Como mostrado na Figura 3.8, o tempo máximo de uso de todos foi de 10 minutos ou menos, mostrando que de fato o aplicativo cumpre seu propósito, principalmente ao analisar os resultados da questão 5, onde mostra uma melhora no conhecimento de todos.

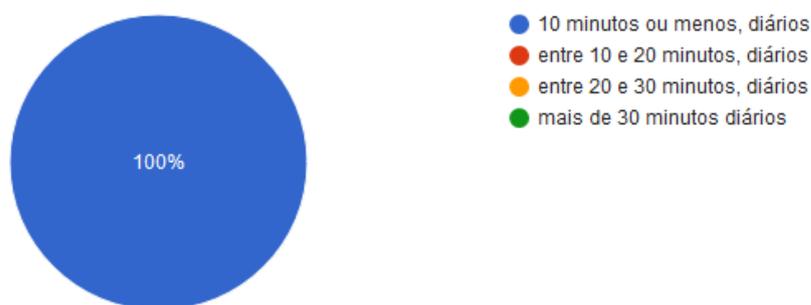


Figura 3.8: Gráfico das respostas da terceira questão do formulário

4 – Em quais momentos você usava o aplicativo?

A intenção aqui era saber se outro objetivo do aplicativo foi atingido, que seria para suprir a necessidade de um objeto de estudo compacto e que pudesse estar presente quando não dava para carregar grandes quantidades de materiais consigo. Como apresentado na Figura 3.9 75% dos participantes utilizaram o aplicativo no trabalho ou nos momentos vagos, quando não haviam grandes quantidades de materiais disponíveis. Enquanto apenas 25% utilizou em casa, onde temos este material disponível.

Logo podemos concluir que o aplicativo funcionou para a maioria dos participantes, que não tiveram acesso ou tempo para se concentrar em estudos aprofundados e conseguiram utilizar o aplicativo para suprir a falta de tais materiais.

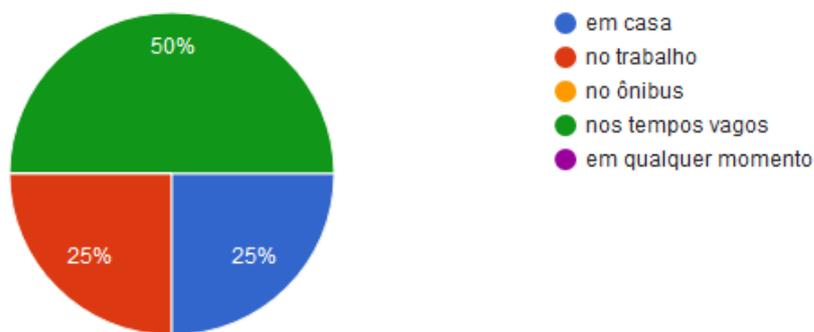


Figura 3.9: Gráfico das respostas da quarta questão do formulário

5 – Em uma escala de 0 a 5 diga quanto o aplicativo te auxiliou no reforço de conceitos do Scrum.

Analisando as respostas individuais, mostradas na Figura 3.10, pode-se notar que todos os alunos que participaram conseguiram ter seu conhecimento sobre o Scrum reforçado. Ponto positivo porque o aplicativo conseguiu atingir e suprir as necessidades nos estudos de cada um.

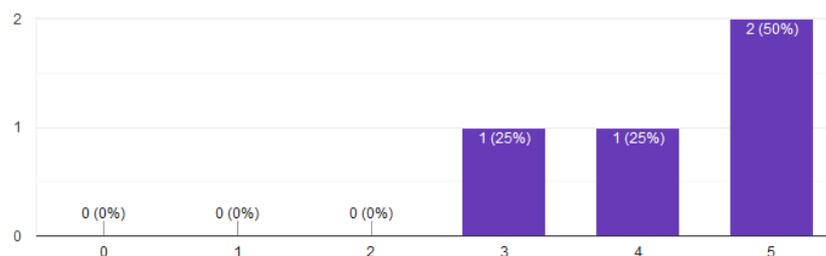


Figura 3.10: Gráfico das respostas da quinta questão do formulário

6 – Baseado na sua experiência usando o aplicativo, como ele poderia melhorar?

Esta foi uma questão aberta para que todos pudessem informar melhorias e soluções para problemas nos quais enfrentaram durante o uso do aplicativo, as respostas obtidas foram as seguintes:

- “Poderia ter uma pontuação para que eu saiba quanto que tenho que melhorar mais. E também poderia marcar quais questões eu já fiz, porque como tem muita questão quando eu voltava no app já não lembrava em qual questão eu tinha parado.”
- “Seria interessante conforme fossemos respondendo as questões tivesse sinalizado as questões já respondidas.”

O ponto mais mencionado, foi a falta de algum tipo de sinalização diferenciando as alternativas resolvidas e não resolvidas. Realmente, este é um ponto a ser melhorado, pois a interface dos exercícios disponibilizava todos os exercícios na tela, não indicando quais exercícios já haviam sido resolvidos.

- “Com dicas e sugestões em questões. E explicação do motivo de ser aquela resposta correta.”

Este *feedback* sugere a criação de dicas para a solução das questões, o que ajuda o usuário a entender não somente o motivo de ter errado alguma questão, como também dá a ele maior detalhes do assunto. Isto proporciona maior conhecimento e ainda ajuda a esclarecer eventuais dúvidas.

- “Seria interessante realizar as perguntas por fases, a cada pergunta respondida corretamente uma nova pergunta seria desbloqueada.”

Adicionar fases ao aplicativo o torna muito mais interessante e informativo, no sentido de que o usuário consegue se localizar facilmente dentro da interface sabendo seu progresso. Ainda há o aspecto lúdico em que o usuário é informado de seu crescimento em nível, dentro das fases do aplicativo.

Conclusão

O *Framework* Scrum foi o tema escolhido para ser abordado no aplicativo. Pois este possui um grande potencial, principalmente por ser um *framework* maleável, que não se limita a apenas o desenvolvimento de software, como também permite a inserção de métodos de outros *frameworks*.

O desenvolvimento da aplicação sofreu alguns agravantes, como falta de tempo, quantidade muito pequena de participantes para a validação e a pouca necessidade. Porém, baseado na amostra dos resultados obtidos com a aplicação, pode-se dizer que o aplicativo atingiu as expectativas, e conseguiu de fato, ajudar os alunos a reforçar o conhecimento do conteúdo.

Para incorporar os pontos sugeridos, fica como proposta para trabalhos futuros a implementação de um ranque geral de todos os usuários ou alguma outra funcionalidade que pudesse desenvolver alguma motivação entre os usuários. Também será desenvolvida uma interface para que o usuário conseguisse ver seu progresso e saber onde está com maior dificuldade, além de algumas dicas e esclarecimentos nas alternativas das atividades. Para a validação, espera-se realizar a validação com uma quantidade maior de alunos, assim pode-se obter melhor *feedback* sobre o aplicativo.

Seria interessante também analisar mais amplamente as respostas discursivas onde os alunos propuseram melhorias ao aplicativo apresentado. Esta análise poderia ser feita juntamente com os participantes da aplicação deste projeto, a fim de entender melhor suas necessidades e as propostas apresentadas. Não existe um aplicativo que funcionará igualmente para todos que o usam, devido a muitos fatores, entretanto todos podem ser melhorados para aderir às necessidades de seu público.

Formulário de avaliação do aplicativo eScrum

1) Em uma escala de 0 a 5, qual era seu nível de conhecimento sobre o Scrum antes de utilizar o aplicativo?

Nenhum () 0 () 1 () 2 () 3 () 4 () 5 Muito

2) Quantas vezes ao dia você abriu o aplicativo?

- () 1 vez por dia
- () 2 vezes por dia
- () 3 vezes por dia
- () 4 vezes por dia
- () 5 ou mais vezes por dia

3) Em média, quanto tempo você gastou no aplicativo, por dia?

- () 10 minutos ou menos, diários
- () entre 10 e 20 minutos, diários
- () entre 20 e 30 minutos, diários
- () mais de 30 minutos diários

4) Em quais momentos você usava o aplicativo?

- () em casa
- () no trabalho
- () no ônibus
- () nos tempos vagos

em qualquer momento

outros: _____

5) Em uma escala de 0 a 5, diga quanto o aplicativo te auxiliou no reforço de conceitos do Scrum.

Nada 0 1 2 3 4 5 Muito

6) Baseado na sua experiência usando o aplicativo, como ele poderia melhorar?

Banco de Questões

1) Os métodos ágeis são fundamentados no desenvolvimento e entrega funcional tendo em vista atender aos requisitos dos clientes. Eles agregam um conjunto de princípios provenientes do manifesto ágil, tais como:

- envolvimento do cliente;
- entrega incremental;
- pessoas, não processos;
- aceitação das mudanças;
- manutenção da simplicidade.

O Scrum é um exemplo de método ágil de gerenciamento de projetos. Avalie as afirmações a seguir sobre a relação do Scrum com os princípios do manifesto ágil.

I– O Scrum adota a entrega incremental por meio de Sprints.

II– O Scrum adota a simplicidade por meio do uso da programação em pares.

III– O Scrum adota o envolvimento do cliente com a priorização e a negociação dos requisitos na concepção de Sprints.

É correto o que se afirmar em

- (a) II, apenas.
- (b) III, apenas.
- (c) I e II, apenas.
- (d) I e III, apenas.
- (e) I, II e III.

2) O que melhor descreve a Sprint Review?

(a) É uma reunião para que o Product Owner faça uma revisão das atividades do Time de Desenvolvimento durante o Sprint.

-
- (b) É uma reunião para que o Time de Desenvolvimento faça uma revisão dos seus processos durante o Sprint.
 - (c) É uma reunião para dar aos clientes e demais partes interessadas uma oportunidade de inspecionar o que foi feito durante o Sprint e dar feedback.
 - (d) É uma reunião utilizada para construir o espírito de equipe.

3) Em que o planejamento ágil é diferente do tradicional?

- (a) O planejamento ágil é similar ao tradicional.
- (b) O planejamento ágil é feito apenas uma vez.
- (c) O planejamento é considerado desperdício, então é feito o mínimo possível.
- (d) O plano ágil não pode ser modificado.
- (e) O planejamento ágil utiliza apenas o nível de detalhes disponíveis em cada momento.

4) O princípio ágil “Simplicidade– a arte de maximizar a quantidade de trabalho não realizado– é essencial” pode ser melhor traduzido em quais termos?

- (a) “Qualidade é o que importa”.
- (b) “Evite o desperdício”.
- (c) “Trabalhe o mínimo possível”.
- (d) “Entregue apenas o que estiver pronto”.
- (e) “Maximize o seu trabalho”.

5) Com que frequência a reunião de retrospectiva deve ser realizada e por quê?

- (a) Sempre que o Time de Desenvolvimento achar necessário, pois o time é auto-organizado.
- (b) Se o Time de Desenvolvimento está entregando, a reunião de retrospectiva não é necessária.
- (c) Ao final de cada Sprint, pois é através dela que o Time de Desenvolvimento inspeciona seus processos para então adaptá-los, de forma a melhorar continuamente.
- (d) Ao final do projeto, porque as lições aprendidas podem ser usadas nos próximos projetos.

6) Durante uma reunião de Sprint Planning, a carga de trabalho necessária para cumprir os itens desejados pelo Product Owner parece maior do que a capacidade do Time de Desenvolvimento. Que ação faz mais sentido?

- (a) Time de Desenvolvimento e Product Owner colaboram e possivelmente removem ou modificam itens.
- (b) Cancela-se o Sprint.
- (c) O Product Owner exige que o Time de Desenvolvimento trabalhe horas extras somente por esse Sprint, desde que isso não aconteça novamente.
- (d) O Time de Desenvolvimento inicia o Sprint e pega emprestados desenvolvedores de outros times.
- (e) Aumenta-se o tamanho da Sprint.

-
- 7) Para que serve o Daily Scrum (reunião diária)?
- (a) Para que o Time de Desenvolvimento preste contas para os clientes do projeto.
 - (b) Para realizar discussões e ajustes técnicos no trabalho do Time de Desenvolvimento.
 - (c) Para promover visibilidade sobre o trabalho e comunicação entre os membros do Time de Desenvolvimento.
 - (d) Para que o Time de Desenvolvimento informe seus impedimentos ao Scrum Master.
 - (e) Para que o Time de Desenvolvimento preste contas para o Scrum Master.

- 8) Quando termina uma Sprint?
- (a) Quando o tempo fixo determinado para a Sprint acaba.
 - (b) Quando o Product Owner determina que a Sprint acabou.
 - (c) Quando todas as tarefas do Sprint Backlog estão completas.
 - (d) Quando todos os itens do Product Backlog atingem a Definição de Pronto.

- 9) Quando uma Sprint deve ser interrompida?
- (a) Quando o Time de Desenvolvimento sente que o trabalho planejado está difícil demais.
 - (b) Quando a meta estabelecida para o Sprint não faz mais sentido.
 - (c) Quando o cliente solicita uma mudança urgente.
 - (d) Quando o Time de Desenvolvimento sente que o trabalho planejado não será terminado.

10) No Scrum, o Product Owner, o Scrum Master e demais interessados no produto definem o Product Backlog, estabelecendo os itens a serem desenvolvidos, ordenados a partir dos mais importantes ou relevantes, e respeitando critérios de ordenação que incluem fatores como valor, custo, conhecimento ou risco.

- (a) Certo.
- (b) Errado.

- 11) Em uma equipe que trabalha orientada pelo Scrum,
- (a) a descoberta de um “erro” ou “defeito” recebe máxima prioridade e justificativa a extensão da Sprint por até 15 dias.
 - (b) a identificação de novos requisitos pode ser feita durante reunião (Daily Scrum), na qual estão presentes clientes do futuro produto.
 - (c) o Scrum Master é responsável por orientar a equipe de desenvolvimento e definir como transformar itens do backlog no incremento proposto para a sprint em questão.
 - (d) a responsabilidade pela produção do incremento cabe a toda a equipe de desenvolvimento, a despeito de um determinado membro da equipe possuir habilidades específicas.

12) Na metodologia Scrum, quais são os itens registrados dentro de uma “Retrospectiva”?

-
- (a) Pontos positivos, negativos e melhorias para a próxima iteração.
 - (b) Itens entregues e itens a serem desenvolvidos.
 - (c) Itens não entregues a serem desenvolvidos na próxima iteração.
 - (d) Estimativas para o desenvolvimento de funcionalidades escolhidas pelo cliente.

13) O Scrum está sendo implantado dentro da sua empresa, portanto existe a necessidade de se criar o Time Scrum que é formado pelo:

- (a) Product Backlog, Time de Planejamento e Scrum Master
- (b) Product Owner, Time de Desenvolvimento e Scrum Sprint
- (c) Product Backlog, Time de Planejamento e Scrum Sprint
- (d) Product Owner, Time de Desenvolvimento e Scrum Master
- (e) Product Backlog, Time de Desenvolvimento e Scrum Sprint

14) Assinale a opção correta a respeito do emprego do product backlog na abordagem ágil para o levantamento de requisitos e para o desenvolvimento de sistemas, de forma coerente com a interpretação do Scrum.

- (a) Um product backlog é uma lista de tarefas que poderão ser desenvolvidas em, aproximadamente, quatro semanas.
- (b) Após a descoberta das users histories, os times de desenvolvimento são responsáveis por definir as histórias de maior prioridade do product Backlog.
- (c) O product backlog deve ser completo ao início do projeto em relação às histórias de usuário, para evitar adaptações no planejamento.
- (d) Quando o sistema entra em produção, o product backlog é encerrado.
- (e) O product backlog documenta a descrição, a ordem, a estimativa e o valor dos seus itens componentes, reunindo as funções e os requisitos do produto, bem como as melhorias que devem ser feitas nesse produto nas futuras versões.

15) Uma das quatro oportunidades formais que o Scrum prescreve para a inspeção e a adaptação das tarefas para atingir os objetivos é a reunião de

- (a) Definição do time
- (b) Definição de Backlog
- (c) Definição de documentos
- (d) Definição de qualidade
- (e) Revisão da Sprint

16) Na elaboração das estórias que irão compor o Backlog de uma Sprint do Scrum, devem-se levantar os objetivos e ser

- (a) mais técnico e propor solução.
- (b) menos técnico e desenhar um workflow.

-
- (c) mais técnico e detalhar o desenvolvimento.
 - (d) menos técnico e se ater ao negócio.
 - (e) menos técnico e focar nos componentes já existentes.

17) Uma das vantagens do Scrum é que, com ele, aplica-se um mecanismo de redução sistemática de risco, por meio de seu ciclo frequente de inspeção e adaptação.

- (a) Certo.
- (b) Errado.

18) Daily Scrum, também conhecida como Daily Standup, é uma reunião diária rápida para se atualizar o Scrum Master sobre o estado do projeto.

- (a) Certo.
- (b) Errado.

19) Três fundamentos da teoria de controle empírico de processos são aplicáveis ao modelo Scrum: transparência, inspeção e adaptação. Considerando esse assunto, julgue o próximo item.

A transparência ocorre devido ao uso de técnicas de encapsulamento presentes nas linguagens orientadas a objetos, que possibilitam abstrair a complexidade do código que está sendo desenvolvido com o uso do Scrum.

- (a) Certo.
- (b) Errado.

20) No Scrum, os projetos são particionados em ciclos de tempo denominados Sprints, nos quais um conjunto de atividades deve ser executado.

- (a) Certo.
- (b) Errado.

21) Os objetivos de uma Daily Scrum incluem a disseminação do conhecimento sobre o que foi executado no dia anterior, a identificação de impedimentos e a priorização do trabalho a ser realizado.

- (a) Certo.
- (b) Errado.

22) No Contexto da metodologia Scrum, ocorre uma Sprint quando o Scrum Master acelera o ciclo de desenvolvimento de um produto ou serviço.

- (a) Certo.
- (b) Errado.

23) Conforme a metodologia SCRUM, Sprint Planning Meeting é uma reunião de planejamento em que o Scrum Master prioriza os itens do Product Backlog e a equipe seleciona as atividades a serem implementadas no período.

- (a) Certo.
- (b) Errado.

24) Com relação ao Scrum, considere:

I– O Product Owner, ou dono do produto, é responsável por garantir que o Scrum seja entendido e aplicado. Faz isso para garantir que o Time Scrum adere à teoria, práticas e regras do Scrum. É um servo-líder para o Time Scrum.

II– O Scrum Master é o responsável por maximizar o valor do produto e do trabalho do Time de Desenvolvimento. Como isso é feito pode variar amplamente nas organizações, Times Scrum e indivíduos.

III– O coração do Scrum é a Sprint, um time-boxed de um mês ou menos, durante o qual um “Pronto”, versão incremental potencialmente utilizável do produto, é criado.

Está correto o que consta APENAS em

- (a) I e III.
- (b) I e II.
- (c) III.
- (d) II e III.
- (e) II.

25) Analise as afirmativas a seguir sobre Scrum e assinale a alternativa correta

I– As Sprints, ciclos de desenvolvimento, em geral têm duração de 4 ou 5 dias.

II– O Product Backlog é essencialmente gerenciado e mantido pelo Product Owner.

III– As equipes de desenvolvimento devem ser consideravelmente pequenas para que os resultados sejam ótimos; no geral, a quantidade máxima de desenvolvedores para cada equipe gira em torno de 2 ou 3 pessoas.

- (a) Somente a alternativa I é verdadeira.
- (b) Somente a afirmativa II é verdadeira.
- (c) As alternativas I, II e III são verdadeiras.
- (d) Somente as alternativas i e II são verdadeiras.

26) A figura abaixo representa uma visão geral de uma metodologia ágil. Os itens de I a IV representam as reuniões comumente realizadas nesta metodologia, que são, respectivamente,

- (a) Master Meeting – Product Owner Meeting – Time Meeting – Daily Meeting.
- (b) de Retrospectiva – de Planejamento – Diária – de Revisão.
- (c) de Planejamento – Diária – de Retrospectiva – de Revisão.
- (d) de Planejamento – de Retrospectiva – de Revisão – Diária.

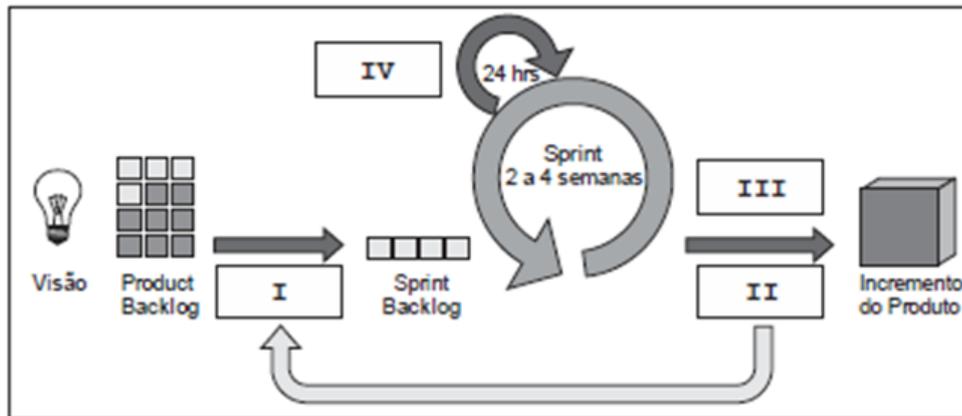


Figura B.1: Exercício 26 *Representação do processo Scrum 1*

(e) Product Owner Meeting – Master Meeting – Daily Meeting – Time Meeting.

27) O Scrum é um *framework* ágil para suportar o desenvolvimento e manutenção de produtos complexos. Nesta abordagem,

(a) a Equipe de Desenvolvimento é composta pelo Product Owner, o Time Scrum e o Scrum Master. A Equipe é autoorganizável e escolhe qual a melhor forma para completar seu trabalho.

(b) o Scrum Master é o responsável por maximizar o valor do produto e do trabalho do Time Scrum. É a única pessoa responsável por gerenciar o Backlog do Produto.

(c) o tamanho ideal do Time Scrum, responsável pela codificação dos testes, é de 3 componentes. Se houver mais de 5 integrantes é exigida muita coordenação e não vai funcionar.

(d) o trabalho a ser realizado na Sprint é planejado na reunião de planejamento da Sprint, que é um time-box de 8 horas para uma Sprint de 1 mês de duração. Para Sprints menores, este evento deve ser proporcionalmente menor.

(e) a Revisão da Sprint é uma oportunidade para o Time Scrum inspecionar a si próprio e criar um plano para melhorias a serem aplicadas na próxima Sprint. Esta é uma reunião time-boxed de 5 horas para uma Sprint de 2 meses.

28) O Scrum é um modelo ágil para a gestão de projeto de software. No Scrum,

(a) o Scrum Team é a equipe de desenvolvimento com 6 a 10 pessoas, necessariamente dividida em papéis como analista, designer e programador.

(b) o Scrum Master é um gerente e um líder como nos modelos prescritivos, já que as equipes não são auto-organizadas.

(c) o Product Backlog precisa ser completo desde o início do projeto, contemplando todas as funcionalidades.

(d) as funcionalidades a serem implementadas em cada projeto (requisitos ou histórias de usuário) são mantidas em uma lista chamada de Product Backlog.

(e) o Product Owner define quais são os requisitos mais importantes a serem tratados em cada

sprint, porém, não é o responsável pelo ROI (Return Of Investment), nem por avaliar as necessidades dos clientes.

29) Observe a figura que apresenta elementos básicos do *framework* Scrum. As caixas I, II,

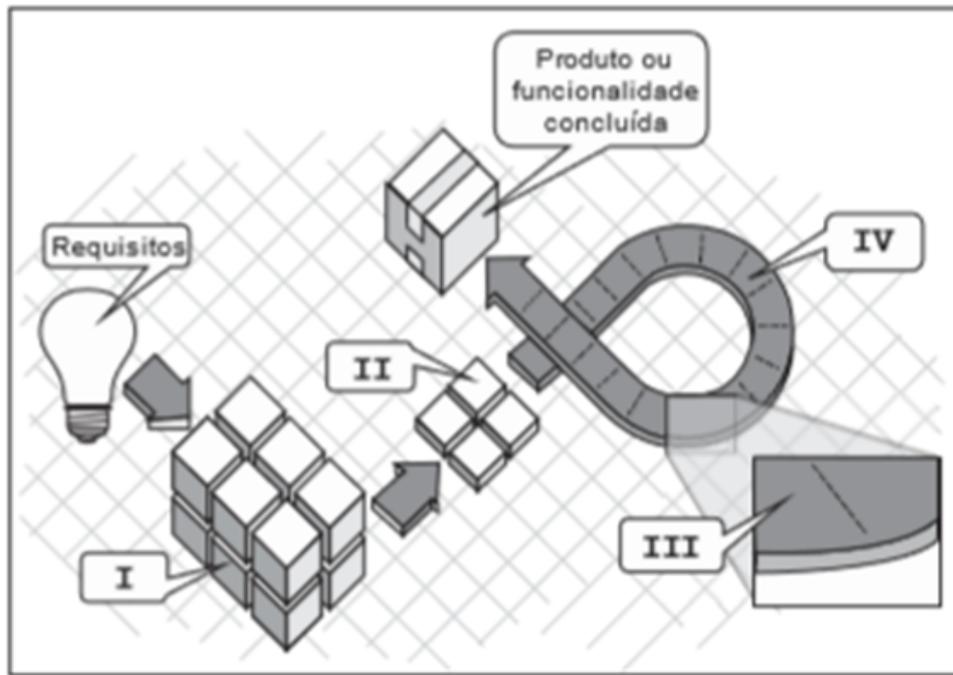


Figura B.2: Exercício 29 *Representação do processo Scrum 2*

III e IV são, correta e respectivamente, preenchidas por:

- (a) requisitos do produto – sprint planning meeting – time-box – daily scrum
- (b) backlog do produto – backlog da sprint – daily scrum – sprint
- (c) box do produto – sprint review – sprint box – time-box
- (d) sprint planning meeting – sprint review – sprint meeting – sprint
- (e) backlog da sprint – backlog do produto – daily meeting – sprint box

30) Um dos pontos da metodologia Scrum é o Daily Scrum, que consiste em uma reunião diária com aproximadamente 15 minutos de duração onde são tratados assuntos relacionados ao projeto. Nessa reunião são feitas 3 perguntas a cada membro do time de desenvolvimento, constando o que foi feito desde a última reunião, o que será feito até a próxima reunião e qual

- (a) modelo de testes está sendo utilizado pela tarefa atual.
- (b) o tempo restante para finalização da tarefa.
- (c) a relação da tarefa atual com o outro membro da equipe.
- (d) a tarefa que está sendo executada no momento.
- (e) obstáculo impede o desenvolvedor de prosseguir com a tarefa.

31) Em um projeto gerido com a metodologia Scrum, um produto estará, ao final de cada

Sprint, completamente testado, estando 100% completos todos os requisitos do Product Backlog.

- (a) Certo.
- (b) Errado.

32) A metodologia Scrum, ágil para gerência de projetos, baseia-se em ciclos de 30 dias, denominados Sprints, em que se trabalha para alcançar objetivos bem definidos.

- (a) Certo.
- (b) Errado.

33) A metodologia Scrum prega que a equipe complete e entregue partes do produto final constantemente ao final de cada iteração. Essa iteração deve ser curta e possuir tempo de execução definido previamente.

- (a) Certo.
- (b) Errado.

34) No Scrum, o Product Owner (PO) é responsável por definir a visão do produto e remover os impedimentos, enquanto o Scrum Master (SM) é responsável por elaborar e manter o Product Backlog, bem como por ajudar o PO a executar suas atividades diárias.

- (a) Certo.
- (b) Errado.

35) Scrum é uma metodologia ágil para gerência de projetos que

- (a) tem foco na gerência do portfólio de projetos da empresa, buscando otimizar o trabalho feito de forma global.
- (b) garante a qualidade por meio de uma política formal de testes, realizados por um escritório que atende a vários projetos.
- (c) libera os programadores para se dedicarem integralmente ao desenvolvimento do software, sem necessidade de comunicação com os clientes.
- (d) utiliza reuniões com objetivos específicos para o planejamento e o acompanhamento do projeto.
- (e) utiliza os modernos métodos de comunicação da Web 3.0.

36) O ciclo de vida da metodologia Scrum se divide nas fases de pré-planejamento, desenvolvimento e pós-planejamento. O documento denominado Product Backlog é gerado na fase de desenvolvimento.

- (a) Certo.
- (b) Errado.

37) No âmbito do desenvolvimento ágil de sistemas de informação, é INCORRETO afirmar que, no Scrum,

- (a) as atividades são definidas com uma duração fixa.
- (b) o foco é nas tarefas e não nos objetivos e resultados.
- (c) o desenvolvimento é iterativo e incremental.
- (d) cada iteração foca nas necessidades mais prioritárias.
- (e) cada iteração é finalizada com funcionalidades completas.

38) Sobre os princípios do método de desenvolvimento Scrum, que são consistentes com o manifesto ágil, julgue as seguintes afirmativas e assinale a alternativa correta.

I– Testes e documentação constantes são realizados à medida que o produto é construído.

II– O processo produz frequentes incrementos de software que podem ser inspecionados, ajustados, testados, documentados e expandidos.

III– O trabalho de desenvolvimento e o pessoal que o realiza é dividido em partições claras, de baixo acoplamento, ou em pacotes.

- (a) Apenas as afirmativas I e II são corretas.
- (b) Apenas as afirmativas I e III são corretas.
- (c) Apenas as afirmativas II e III são corretas.
- (d) Todas as afirmativas são corretas.
- (e) Nenhuma das afirmativas é correta.

39) A metodologia Scrum é uma forma de trabalho rígida empregada em ambientes organizacionais departamentais e conservadores.

- (a) Certo.
- (b) Errado.

40) Uma das características da metodologia ágil Scrum é

- (a) focar nas práticas de engenharia.
- (b) focar na documentação formal do software.
- (c) ser um método iterativo e incremental.
- (d) exigir o planejamento do projeto, de acordo com as práticas do PMBOK.
- (e) não exigir interação com o cliente.

41) Sprint é a denominação dada ao padrão de processo de realização de tarefas das atividades metodológicas do Scrum, cujos princípios são consistentes com o manifesto ágil, pois incorporam as atividades estruturais de requisitos, análise, projeto, evolução e entrega.

- (a) Certo.
- (b) Errado.

42) O Scrum é um processo ágil que prioriza a entrega de maior valor de negócio no menor tempo.

- (a) Certo.
- (b) Errado.

43) Scrum é um modelo utilizado no desenvolvimento ágil de software. No Scrum um dos conceitos mais importantes é o Sprint, que consiste em um ciclo de desenvolvimento que, em geral, vai de duas semanas a um mês.

No início de cada Sprint é feito um I, no qual a equipe prioriza os elementos do II a serem implementados e transfere esses elementos para o III, ou seja, a lista de funcionalidades a serem implementadas no ciclo que se inicia.

A equipe se compromete a desenvolver as funcionalidades e o IV se compromete a não trazer novas funcionalidades durante o mesmo Sprint.

As lacunas I, II, III, IV são preenchidas, corretamente e respectivamente, por

- (a) Sprint Burdown, Product Backlog, Sprint Backlog, Scrum Team.
- (b) Sprint Planning Meeting, Product Backlog, Sprint Backlog, Product Owner.
- (c) Scrum Planning, Sprint Backlog, Product Backlog, Product Owner.
- (d) Sprint Planning Meeting, Product Backlog, Sprint Backlog, Scrum Master.
- (e) Scrum Daily Meeting, Product Backlog, Sprint Backlog, Scrum Master.

44) Na metodologia Scrum, NÃO faz parte de uma revisão do sprint (sprint review) o seguinte procedimento:

- (a) Todo o time colabora no que deve ser feito em seguida, de modo que esta revisão contribua para reuniões de planejamento subsequentes.
- (b) O proprietário do produto identifica o que está pronto e o que ainda está por fazer.
- (c) O time de desenvolvimento discute quais fatores positivos e negativos ocorreram durante a sprint e como os problemas foram resolvidos.
- (d) O time de desenvolvimento apresenta o trabalho que foi desenvolvido e responde questões sobre o incremento.
- (e) Todo o time cria um plano para implementar melhorias no modo como o time efetua seu trabalho.

45) Na reunião de planejamento do Sprint Backlog, se o Product Owner afirmar que todos os requisitos do produto foram identificados, é correto concluir que o Backlog do Produto está completo, visto que este é uma lista ordenada de todos os requisitos necessários para o desenvolvimento do produto.

- (a) Certo.
- (b) Errado.

46) Qual é a forma primária pela qual o Scrum Master colabora para que um Time de Desenvolvimento seja efetivo?

- (a) Assegurando que as reuniões iniciem e terminem no horário adequado.
- (b) Mantendo funcionalidades de maior valor no alto do Product Backlog.
- (c) Impedindo o contato entre clientes e Time de Desenvolvimento.
- (d) Agindo como um facilitador e removendo impedimentos.

47) Pelo que o Scrum Master é responsável?

- (a) Por ensinar Scrum e buscar que seja adotado e utilizado corretamente.
- (b) Por definir as tarefas que o Time de Desenvolvimento deverá fazer na Sprint.
- (c) Pelas reuniões do Time de Desenvolvimento e pelos objetivos que o Time de Desenvolvimento traça para si.
- (d) Por definir métricas e gerenciar o desempenho do Time de Desenvolvimento.

48) O recurso que, devido ao fato de ser considerado guardião do processo, garante que as técnicas de Scrum sejam aplicadas em um projeto denomina-se

- (a) líder de equipe.
- (b) coordenador do projeto.
- (c) líder técnico
- (d) Product Owner.
- (e) Scrum Master.

49) Acerca da metodologia ágil Scrum, assinale a opção correta.

- (a) O scrum master desempenha o papel de gerente de projetos no ciclo de vida do projeto, além de possuir as atribuições de orientador do time especialista.
- (b) Os sprints possuem um tempo fixo de duração de uma semana, o que determina que o sprint review ocorra diariamente e que a retrospectiva seja realizada ao final de cada sprint.
- (c) Um time scrum deve englobar uma equipe voltada para os esforços especialistas, devendo o gerenciamento da equipe ser centralizado no scrum master.
- (d) A atribuição do backlog do produto é compartilhada entre o dono do produto (product owner), o scrum master e a equipe especialista.
- (e) O scrum master deve ser um líder cooperativo para a equipe, devendo ser um orientador e ponto focal entre o dono do produto (product owner) e a equipe especialista.

50) O time de desenvolvimento do *framework* Scrum visa garantir que este seja entendido e aplicado, de forma a assegurar aderência à teoria, às práticas e às regras desse *framework*.

- (a) Certo.
- (b) Errado.

51) O Scrum Master deve assumir a gerência de um projeto ágil com base no SCRUM, de modo a definir as prioridades para que a equipe entregue, primeiramente, os produtos de software que agreguem maior valor ao negócio do cliente.

- (a) Certo.
- (b) Errado.

52) Em quais atividades o Product Owner comumente se envolve durante o tempo de desenvolvimento em um Sprint?

- (a) Prover feedback sobre o trabalho do Time de Desenvolvimento quando solicitado, responder a perguntas do Time de Desenvolvimento sobre itens no Sprint atual e priorizar as tarefas do Time de Desenvolvimento.
- (b) Prover feedback sobre o trabalho do Time de Desenvolvimento quando solicitado, responder a perguntas do Time de Desenvolvimento sobre itens no Sprint atual e preparar itens para o próximo Sprint junto ao Time de Desenvolvimento.
- (c) O Product Owner não se envolve em atividades durante o Sprint.
- (d) Trabalhar junto aos clientes, acompanhar a reunião diária e monitorar as tarefas do Time de Desenvolvimento.
- (e) responder a perguntas do Time de Desenvolvimento sobre itens no Sprint atual, trabalhar junto aos stakeholders e acompanhar a reunião diária.

53) Que frase melhor descreve a principal responsabilidade do Product Owner?

- (a) Definir e monitorar o cronograma do projeto.
- (b) Dizer ao Time de Desenvolvimento o que ele deve fazer.
- (c) Gerenciar o projeto garantindo que o trabalho esteja de acordo com os compromissos estabelecidos com os clientes do projeto.
- (d) Otimizar o retorno sobre o investimento do trabalho realizado pelo Time de Desenvolvimento.

54) Deve-se colocar em produção cada incremento produzido na Sprint...

- (a) Sempre, sem exceção.
- (b) Sempre que o Product Owner decidir que faz sentido.
- (c) Sempre que a equipe garantir que o incremento é entregável.
- (d) Sempre que o Scrum Master decidir que faz sentido.
- (e) Sempre que o incremento estiver livre de erros.

55) Quem tem a última palavra sobre as prioridades dos itens no Product Backlog?

- (a) O Scrum Master.
- (b) O Gerente de Projetos.
- (c) O Time de Desenvolvimento.

-
- (d) Os clientes do projeto.
 - (e) O Product Owner.

56) Na metodologia Scrum, o responsável por definir uma lista de critérios de aceitação do produto é o

- (a) Chefe do desenvolvimento.
- (b) Product Owner.
- (c) Scrum Master.
- (d) Time de desenvolvimento.
- (e) Gerente de orçamento.

57) O cancelamento do time-box da Sprint, antes do seu término, só poderá ser realizado formalmente pelo

- (a) Product Owner.
- (b) Administrador de infraestrutura.
- (c) Scrum Master.
- (d) Designer de interface.
- (e) Analista de usabilidade.

58) Um projeto Scrum inicia-se com o Product Owner, que coleta informações dos stakeholders a fim de que seja elaborada uma lista de requisitos e de um Backlog de produto.

- (a) Certo.
- (b) Errado.

59) No Scrum, um projeto se inicia com uma visão simples do produto que será desenvolvido. A visão pode ser vaga a princípio e ir tornando-se clara aos poucos. O I então, transforma essa visão em uma lista de requisitos funcionais e não-funcionais para que, quando forem desenvolvidos, reflitam essa visão. Essa lista, chamada de II, é priorizada pelo III de forma que os itens que gerem maior valor ao produto tenham maior prioridade.

Completa, correta e respectivamente, as lacunas I, II e III:

- (a) Daily Scrum – Scrum Team – Sprint
- (b) Daily Scrum – Product Backlog – Sprint Planning Meeting
- (c) Product Owner – Sprint – Product Backlog
- (d) Scrum Team – Sprint Planning Meeting – Product Owner
- (e) Product Owner – Product Backlog – Product Owner

60) SCRUM é um *framework* baseado no modelo ágil. No SCRUM,
(a) o Scrum Team é a equipe de desenvolvimento, necessariamente dividida em papéis como analista, designer e programador. Em geral o Scrum Team tem de 10 a 20 pessoas.

-
- (b) as funcionalidades a serem implementadas em cada projeto (requisitos ou histórias de usuários) são mantidas em uma lista chamada de Scrum Board.
- (c) o Scrum Master é um gerente no sentido dos modelos prescritivos. É um líder, um facilitador e um solucionador de conflitos. É ele quem decide quais requisitos são mais importantes.
- (d) um dos conceitos mais importantes é o Sprint, que consiste em um ciclo de desenvolvimento que, em geral, tem duração de 4 a 7 dias.
- (e) o Product Owner tem, entre outras atribuições, a de indicar quais são os requisitos mais importantes a serem tratados em cada Sprint. É responsável por conhecer e avaliar as necessidades dos clientes.

61) Se for averiguado, em uma organização, que o Scrum Master gerencia o Backlog do Produto, é correto afirmar que houve falha na execução de papéis, visto que cabe unicamente ao Product Owner gerenciar o Backlog do Produto.

- (a) Certo.
- (b) Errado.

62) No Scrum, que papel é responsável pela visão do produto e pelo retorno do investimento?

- (a) Scrum Master.
- (b) Product Owner.
- (c) Sprint Planner.
- (d) Gerente do Projeto.
- (e) Analista de Sistemas Sênior.

63) O time de desenvolvimento escolheu, juntamente com o Product Owner, itens do Product Backlog para desenvolver no Sprint e se comprometeu a atingir uma meta negociada com o Product Owner. Quem deve determinar como o Time de Desenvolvimento deve realizar seu trabalho?

- (a) O Product Owner
- (b) O Scrum Master
- (c) Os gerentes funcionais a quem o Time de Desenvolvimento presta contas
- (d) O Time de Desenvolvimento

64) Idealmente, qual o papel da equipe de qualidade em um projeto ágil?

- (a) Não existe equipe de qualidade em projetos ágeis, a qualidade faz parte do trabalho do Time de Desenvolvimento.
- (b) Testar o produto desenvolvido pelo Time de Desenvolvimento apontar os erros cometidos pelo Time de Desenvolvimento e garantir a qualidade do produto gerado pelo Time de Desenvolvimento.

-
- (c) Garantir a qualidade do produto gerado pelo Time de Desenvolvimento.
 - (d) Apontar os erros cometidos pelo Time de Desenvolvimento.
 - (e) Testar o produto desenvolvido pelo Time de Desenvolvimento.

65) Em um projeto ágil, quem deve estimar o trabalho a ser realizado?

- (a) O Time de Desenvolvimento, o Scrum Master e o Product Owner.
- (b) O Time de Desenvolvimento e o Scrum Master.
- (c) O Product Owner.
- (d) O Gerente de Projetos.
- (e) O Time de desenvolvimento.

66) O que melhor traduz o que se espera de um Time de Desenvolvimento em um projeto ágil?

- (a) Qualidade e precisão nas suas estimativas.
- (b) Qualidade no produto gerado, propriedade sobre suas decisões e responsabilidade.
- (c) Fazer o que seu gerente comandar.
- (d) Trabalhar o tempo que for necessário para terminar o que foi planejado.
- (e) Cumprir exatamente aquilo que foi planejado.

67) Uma das razões por que se mede a velocidade do Time de Desenvolvimento nas Sprints é...

- (a) Não se deve medir a velocidade do Time de Desenvolvimento em nenhum Sprint.
- (b) Para que o Product Owner possa cobrar do Time de Desenvolvimento que ele mantenha sempre a mesma velocidade.
- (c) Para que o Product Owner possa cobrar do Time de Desenvolvimento que ele tenha uma velocidade cada vez maior.
- (d) Para ajudar o Time de Desenvolvimento a decidir o quanto do Product Backlog ele irá selecionar para desenvolver na Sprint.

68) Um novo desenvolvedor entrou em um Time de Desenvolvimento que já exista. Ele está tendo conflitos todo o tempo com os membros e está tornando o ambiente hostil. Se necessário, quem deve identificar o problema e iniciar o processo para resolvê-lo?

- (a) O Scrum Master é responsável, pois ele deve remover impedimentos.
- (b) O Time de Desenvolvimento é responsável, pois ele é um time auto organizado, facilitado pelo Scrum Master.
- (c) O RH da empresa é o responsável, pois deve cuidar das contratações e demissões.
- (d) O Product Owner é responsável, pois ele controla o retorno sobre o investimento do trabalho.

69) O escopo, a importância e a estimativa de um Sprint do Scrum são definidos pelo Pro-

duct Owner.

(a) Certo.

(b) Errado.

70) De acordo com a metodologia Scrum, a constituição ideal da equipe de desenvolvimento para que o trabalho se mantenha ágil deve ser de menos de três pessoas.

(a) Certo.

(b) Errado.

Referências Bibliográficas

ABBAS, N.; GRAVELL, A. M.; WILLS, G. B. Historical roots of agile methods: Where did “agile thinking” come from? In: *International conference on agile processes and extreme programming in software engineering*, Springer, 2008, p. 94–103.

AUDY, J. *Scrum 360: Um guia completo e prático de agilidade*. Editora Casa do Código, 2015.

BARCELOS, R.; TAROUÇO, L.; BERCHT, M. O uso de mobile learning no ensino de algoritmos. *RENOTE*, v. 7, n. 3, p. 327–337, 2009.

BECK, K.; BEEDLE, M.; BENNEKUM, A. V.; COCKBURN, A.; CUNNINGHAM, W.; FOWLER, M.; GRENNING, J.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARRICK, B.; MARTIN, R. C.; MELLOR, S.; SCHWABER, K.; SUTHERLAND, J.; THOMAS, D. Manifesto for agile software development. Disponível em: <http://agilemanifesto.org/>. Acesso em: 12 março 2019, 2001.

COHEN, D.; LINDVALL, M.; COSTA, P. An introduction to agile methods. *Advances in computers*, v. 62, n. 03, p. 1–66, 2004.

DEVEDZIC, V.; MILENKOVIC, S. R. Teaching agile software development: A case study. *IEEE transactions on Education*, v. 54, n. 2, p. 273–278, 2010.

FERNANDA, S.; MANUEL, S.; GERMANIA, R.; SAMANTA, C.; DANILO, J.; PATRICIO, A. Agile methodologies applied in teaching-learning process in engineering: A case of study. In: *2018 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2018, p. 1201–1207.

GLASS, R. L. Agile versus traditional: Make love, not war! *Cutter IT Journal*, v. 14, n. 12, p. 12–18, 2001.

HAWKS, D. 4 advantages of physical task boards. Disponível em: <https://agilevelocity.com/scrum/4-advantages-of-physical-task-boards/>. Acesso em: 21 março 2019, 2010.

HIGHSMITH, J.; COCKBURN, A. Agile software development: the business of innovation. *Computer*, v. 34, n. 9, p. 120–127, 2001.

MADEJA, M.; BIÑAS, M. Implementation of scrum methodology in programming courses. In: *2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, IEEE, 2018, p. 333–340.

PÉREZ, C. R.; CABALLERO, I.; RODRÍGUEZ, M. Improving the experience of teaching scrum. In: *2018 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2018, p. 1598–1605.

RICO, D. F.; SAYANI, H. H. Use of agile methods in software engineering education. In: *2009 Agile Conference*, IEEE, 2009, p. 174–179.

ROGER Planning poker e ideal day: Técnicas de abordagem de estimativa Ágil. Disponível em: <https://www.devmedia.com.br/planning-poker-e-ideal-day-tecnicas-de-abordagem-de-estimativa-agil/31220>. Acesso em: 21 março 2019, 2014.

SABBAGH, R. *Scrum: Gestão ágil para projetos de sucesso*. Editora Casa do Código, 2014.

SCHAFER, U. Training scrum with gamification: Lessons learned after two teaching periods. In: *2017 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2017, p. 754–761.

SCHARFF, C.; HENG, S.; KULKARNI, V. On the difficulties for students to adhere to scrum on global software development projects: preliminary results. In: *Proceedings of the Second International Workshop on Collaborative Teaching of Globally Distributed Software Development*, IEEE Press, 2012, p. 25–29.

SCHWABER, K.; SUTHERLAND, J. Scrum guide: the definitive guide to scrum: The rules of the game. *SCRUM.org, Nov-2017*, 2017.

SCRUMALLIANCE Agile atlas. Disponível em: <https://www.scrumalliance.org/learn-about-scrum/agile-atlas>. Acesso em: 17 abril 2019, 2019.

SCRUMSTUDY Um guia para o conhecimento em scrum (guia sbok). Disponível em: <https://www.scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide-2016-Portuguese.pdf>. Acesso em: 15 abril 2019, 2016.

STANDISHGROUP The chaos report: 2015. 2015.

VICTORINO, M. C.; SCHIESSL, M. O papel do profissional da informação na automação de serviços de informação utilizando o método ágil scrum. In: *XIX Encontro Nacional de Pesquisa em Ciência da Informação (XIX ENANCIB)*, 2018.

WAGH, R. Using scrum for software engineering class projects. In: *2012 Agile India*, IEEE, 2012, p. 68–71.

WAZLAWICK, R. *Engenharia de software: conceitos e práticas*, v. 1. Elsevier Brasil, 1–73 p., 2013.

ZORZO, S. D.; PONTE, L.; LUCRÉDIO, D. Using scrum to teach software engineering: A case study. In: *2013 IEEE Frontiers in Education Conference (FIE)*, IEEE, 2013, p. 455–461.