Scrum Simplificado: Definindo Artefatos.

José Fernandes Rezende Neto

[1]Bacharelado em Engenharia de Computação–Centro Universitário de Anápolis (UniEVANGÉLICA)–Anápolis–GO

j.fr.neto@hotmail.com

Resumo. *O presente trabalho objetiva apresentar artefatos de backlog do framework SCRUM, uma Metodologia Ágil de trabalho, de ciclo de vida Incremental. Aqui será definido as competências de cada artefato de backlog, seus relacionamentos entre si, assim como também restrições e técnicas de identificação. Para tanto fora realizada uma pesquisa de tipo exploratória realizada através de levantamento bibliográfico em documentações de ferramentas, livros e artigos sobre o assunto desse tema.*

Palavras-chave: *Scrum. Desenvolvimento Ágil. Engenharia de Requisitos.*

Abstract. *The present work targets to intruduce artifacts of the Scrum framework, an Agile Metodology of work, of incremental lifecicle. Here will be defined the jurisdiction of each backlog artifact, its relationships between each other, as its restrictions and identification techniques. Therefore was performed a research of exploratory kind done through bibliographic survey in documentation of tools, books and articles about the subject theme*

Palavras-chave: *Scrum. Agile Development. Requirements Engineering.*

1. Introdução

Scrum é uma Metodologia Ágil de desenvolvimento de software criada por Jeff Sutherland e Ken Schwaber em aprimoramento as metodologias de processo sequenciais que mantinham um processo rígido demais para um desenvolvimento dinâmico. Um framework/ferramenta Ágil descreve ferramentas e métodos para o gerenciamento de projetos para o desenvolvimento de software, no Scrum tudo que é realizado visa produzir um artefato.

> 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. [Schwaber, Ken e Sutherland, Jeff. Guia do Scrum. https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf. Consultado em 30 de Julho de 2020. Pg. 13]

Os artefatos do Scrum são produzidos durante o ciclo de vida do processo sendo eles: Product Backlog, Sprint Backlog e Incremento. Neste artigo será definido os limites de cada um desses Artefatos e suas competências, exemplificando através de analogias e como se encaixa cada um no ciclo de vida Scrum.

2. Ciclo de Vida

> Qualquer processo lógico usado por um analista de sistema para desenvolver ou modificar um sistema de informação. Ciclo de vida dos sistemas inclui requisitos, design, desenvolvimento, integração, teste, validação, treinamento, propriedade do usuário, operações, análise e manutenção. [1][Systems Development Life Cycle from. FOLDOC. Consultado em 31 de julho de 2020].

Na prática um ciclo de vida são etapas/processos para se produzir um resultado, que no caso é um sistema de informação. É necessário também ter em mente que um ciclo de vida define **como** as coisas serão feitas, ou pelo menos, um processo de como **deveriam** ser feitas para atingir um objetivo e que se chama ciclo de vida porque este processo atuará durante a vida do sistema, isto é, não somente ao final do seu desenvolvimento, mas enquanto haver suporte para o sistema.

No Scrum se usa um ciclo de vida iterativo, o qual visa entregar um MVP (em inglês: Minimum Viable Product, Em português: Produto Mínimo Viável).

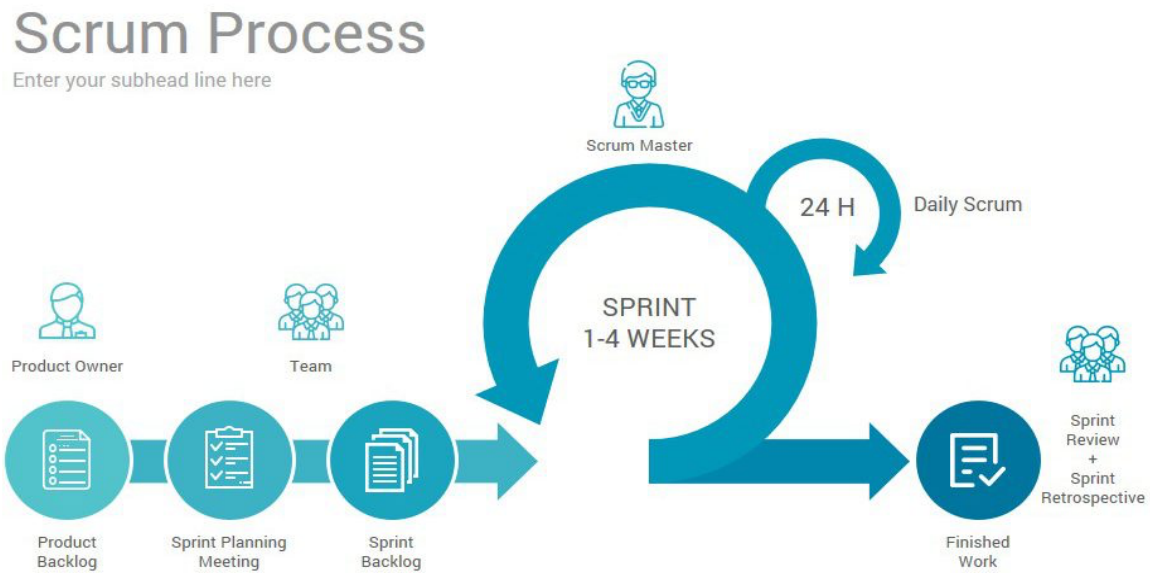
> Muito popular entre startups e durante o desenvolvimento de novos produtos, a ideia por trás do MVP é que o negócio entenda qual é o menor conjunto possível de funcionalidades necessárias para que o produto seja lançado em produção e possa começar a receber feedbacks dos usuários enquanto produz valor para a organização. [Gomes, André. Scrum: o Framework Mínimo Viável. Publicado em: novembro de 2018. Consultado em 30 de julho de 2020].

Para o desenvolvimento do MVP há uma série de processos no ciclo de vida do Scrum dos quais se visa em produzir artefatos.

---

[1] No Original: Any logical process used by a systems analyst to develop or redesign an information system. SDLC includes requirements, design, development, integration, testing, validation, training, user ownership, operations, analysis and maintenance.

Figura 1 - Ciclo de Vida Scrum



Fonte: Blog Education.

3. O Product Backlog

O ***Product Backlog*** (Pendências do Produto) é onde se encontra o conjunto de tarefas/funcionalidades/produtos que serão definidas e priorizadas pelo ***Product Owner*** [2](Proprietário do Produto) e desenvolvidas pela ***Equipe de Desenvolvimento***[3].

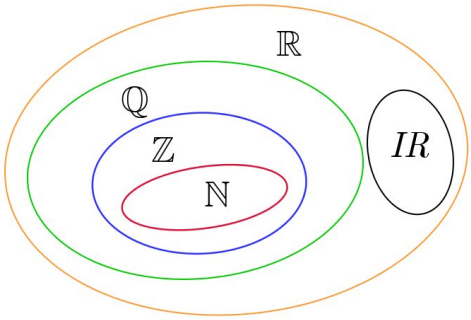
Um ***Product Backlog*** não somente é a lista de tudo que há de ser desenvolvido em um produto, mas também se trata de uma lista ordenada por prioridade, da qual o ***Product Owner*** deve definir o que precisa ser entregue com prioridade para satisfação do cliente (Schwaber, 2013. Sutherland, 2013).

As tarefas do ***Product Backlog*** devem ser organizadas e divididas em partes menores para uma compreensão melhor, cada qual, agrupada dentro da tarefa à qual foi derivada. O objetivo é de que de um objetivo/funcionalidade de fácil compreensão para o cliente se extraia uma tarefa pequena e especifica com linguagem de baixo nível, de fácil compreensão para o desenvolvedor. Uma analogia a isso são os conjuntos numéricos onde por exemplo: números naturais se encontram dentro dos números inteiros, assim como os números inteiros se encontram dentro do conjunto dos números racionais.

[2] Uma pessoa a qual ficará responsável por definir, gerenciar e Maximizar o valor do produto e do trabalho da equipe [Schwaber, Ken e Sutherland. Guia do Scrum. https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf. Consultado em 30 de Julho de 2020. Pg. 5], isto é, defender as intensões do cliente e ao mesmo tempo decidir o que precisa ser feito primeiro para se entregar uma solução útil.

[3] Um Time de pessoas que trabalham para entregar uma "parte" do produto que seja utilizável [Schwaber, Ken e Sutherland. Guia do Scrum. https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf. Consultado em 30 de Julho de 2020. Pg. 6]

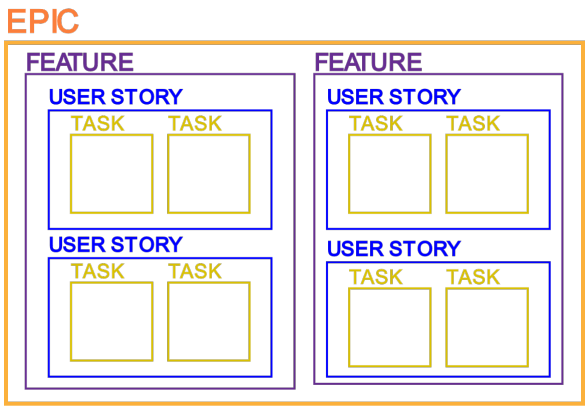
Figura 2 - Conjuntos Numéricos



Fonte: O Autor

Nesse sentido um item de Backlog pode ser categorizado como: ***Epic, Feature, User Story e Task*** **.** Essas categorias funcionam como conjuntos numéricos no sentido que, comportam dentro deles uns aos outros nessa ordem, do maior para o menor.

Figura 3 - Categorias de Items de Backlog



Fonte: O Autor

A **figura 3** descreve como **idealmente** a hierarquia dos items de backlog devem ser organizados dentro de um Artefato de **Product Backlog.**

Estando claro como é a hierarquia desses artefatos resta saber definir dentro do que há de ser produzido para um software o que se encaixa e não encaixa em determinada categoria de **Item de Backlog**, seja nas atividades, nos requisitos, ou de tarefas menores que derivam de requisitos e a diferença dessas categorias entre si.

4. Definindo o que é um Epic

> "Épicos são recipientes para requisitos maiores. Epics são progressivamente refinados em um grupo de pequenas Histórias de

> Usuário no tempo devido[4].” [Rubin, Kenneth. Essential Scrum: A Practical Guide to the Most Popular Agile Process, 2012].

Um ***Epic*** (Em português: Épico) é o artefato de mais alto-nível na hierarquia desses artefatos, isto é, o que mais claramente reflete as intensões do cliente e menos objetivo sobre **o quê** necessariamente deve ser feito e mais sobre **o quê** se deseja obter no final do processo de desenvolvimento. Neste sentido é algo que deve refletir um objetivo geral do cliente com foco em Negócios, porém deve ser genérica e grande o suficiente para englobar outras ações do usuário, por exemplo:

- Um site de Vendas Online
- Um aplicativo de Vendas por Celular
- Uso de computação em Nuvem
- Melhora na performance

Em comparação com Programação Orientada a Objetos, um ***Epic*** é algo que se assemelha a fazer uma biblioteca, Namespace[5] ou projeto dentro de uma solução, no sentido que, para se concluí-los há de haver um grande esforço e deve fazer mais facilmente sentido por alguém que não faz parte do desenvolvimento. Em resumo, um ***Epic*** é um projeto que não chamamos de projeto (Macdonald, Kent. 2018?). Seguindo os objetivos do exemplo acima podemos ter:

- Plataforma Web Mercado Livre
- Aplicação mobile Mercado Livre
- Uso do Amazon Web Services
- Mudança na Arquitetura do sistema

Nesse sentido um ***Epic*** não diz nada ao desenvolvedor em nenhum sentido como será feito somente uma ideia geral do que será feito, de modo que, os envolvidos do projeto consigam entender e enxergar o valor comercial do projeto.

5. Definindo uma Feature/Theme

> “Um Tema fornece uma forma conveniente de indicar que aquele grupo de Histórias de Usuário têm algo em comum, como sendo parte de uma mesma parte funcional”[6] [Rubin, Kenneth. Essential Scrum: A Practical Guide to the Most Popular Agile Process, 2012].

---

[4] Texto Original: Epics are useful as placeholders for large requirements. Epics are progressively refined into a set of smaller user stories at the appropriate time

[5] Um espaço de nomes ("namespace" em inglês) é um delimitador abstrato (container) que fornece um contexto para os itens que ele armazena (nomes, termos técnicos, conceitos...), o que permite uma desambiguação para itens que possuem o mesmo nome mas que residem em espaços de nomes diferentes. Como um contexto distinto é fornecido para cada container, o significado de um nome pode variar de acordo com o espaço de nomes o qual ele pertence. (Wikipedia. Espaço de nomes. Disponível em: https://pt.wikipedia.org/wiki/Espa%C3%A7o_de_nomes. Acesso em: 30 de Julho de 2020.

[6] Texto original: A theme provides a convenient way to indicate that a set of stories have something in common, such as being in the same functional area.

Idealmente uma **Feature/Theme** (em português: Recurso/Tema) deve funcionar como um filtro para determinada funcionalidade ou conjunto de funcionalidades que compartilham características entre si, uma funcionalidade que executa uma ação CRUD (create, read, update, delete/ criar , ler, atualizar e deletar) poderá ser contida dentro de uma **Feature**, por exemplo:

- Gerenciamento de Cliente (**Feature**)
    - Cadastro de Cliente
    - Atualização de Cliente
    - Visualização de Cliente
    - Deletar Cliente

Desse modo é possível observar também que Features têm foco em mostrar o valor comercial/objetivo a ser alcançado através das funcionalidades a serem implementadas ou ações a serem executadas e por assim ser focada em valor comercial deve novamente fazer um sentido para os não-desenvolvedores.

Uma analogia à ***POO***: uma feature pode ser comparada como uma classe que engloba um conjunto de métodos/funções.

6. Definindo um User Story

> [...] uma descrição de uma funcionalidade do produto usada para propósito de escopo e planejamento. Histórias de Usuário irão se decompor em um nível que serão entregues em uma única iteração e fornecer valor[7].[Mcdonald, Kent. Beyond Requirements: Analysis with an Agile Mindset. Publicado em: 21 setembro 2015]

A partir deste nível deve ser mais claro, do ponto de vista do desenvolvedor, o "**o quê?"** será objetivamente desenvolvido/produzido.

Uma ***User Story*** (História de Usuário, em português) é nomeada e descrita usando linguagem de cunho mais técnico. Elas descrevem uma funcionalidade/objeto específico, sua estimativa de produção deve ser maior que um dia e menor que uma semana e seu produto deve agregar valor ao projeto.

Uma ***User Story*** é arquiteturalmente, exemplos de um User Story é:

- Cadastro de cliente
- Atualização do Cliente
- Deletar Cliente
- Visualização do Cliente
- Visualização do produto

[7] Texto original: [...] a description of a product feature used for planning and scoping purposes. User stories will be decomposed to a level that can be delivered in a single iteration and provide value

7. Definindo uma Task

Uma ***Task*** (em português: tarefa) é o menor ***Item Backlog*** possível de um ***Product Backlog***. Uma task tem foco em arquitetura e tende a descrever uma tarefa em linguagem de baixo nível.

Tasks compõe uma ***User Story***, isto é, descrevem atividades a serem completadas para que seja entregue um ***User Story*** completo. O objetivo de uma ***Task*** é encapsular uma tarefa que possa ser completada em menor tempo possível, idealmente, ***Tasks*** são atividades que podem ser completadas em um período de tempo de 1 a 2 dias.

A descrição de uma task deve ser simples e objetiva e deve especificar uma tarefa para a equipe de desenvolvimento cumprir. Uma ***Task*** descreve objetivamente **como** o ***User Story*** a qual esta ***Task*** compõe deve ser implementado, como por exemplo:

- Layout da tela de cadastro do usuário
- Estilizar botões de interface
- Aprimorar fontes

8. Como escrever Backlog Items?

Descrito cada classe de ***Backlog Item*** (em português: Item Pendente) se faz necessário como dizer como cada um deve ser elaborado. Tendo em mente que o Scrum preza pelo tempo de desenvolvimento do ***Produto*** e não pelo tempo de produção de documentação, não há necessidade de detalhamento de ***Backlog Items*** sem necessidade, nem ao menos de ***Backlog Items*** que não estão em desenvolvimento lembrando que o ***Product Backlog*** funciona como uma lista, onde os itens de maior valor para o produto, são produzidos primeiro.

Backlog Items em produção por outro lado, idealmente, devem ser bem detalhados a fim de alcançar alguns benefícios como:

- Compreensão do escopo pela equipe
- Estimativas de esforço
- Testes de desenvolvimento
- Certificar que o produto atinja os critérios de aceitação

8.1. Nome

Apesar de não haver uma regra concreta para nomeação de ***Backlog Items***, há boas práticas para nomeação destes, por exemplo: O Nome de um ***Backlog Item*** deve refletir o que ele é e não como ele é, é um substantivo e não uma ação. Um bom exemplo é um requisito chamado ***Cadastrar Clientes*** é melhor identificável por ***Cadastro de Clientes.***

8.2. Descrição

A descrição deve suprir o time com detalhes suficientes para a compreensão do escopo do ***Backlog Item.*** O foco deve descrever o objetivo do ponto de vista do usuário com aquele ***Backlog Item*** e não dizer como deve ser desenvolvido o produto. Descrever como se dito “que tipo de missa, mas não como a missa deve ser ministrada”.

8.3. Critério de Aceitação

Critério de aceitação deve descrever como deve se comportar para que esse ***Backlog Item*** seja considerado "pronto", definir ***Critérios de Aceitação*** pede que há uma negociação entre o cliente e o ***Time***[8]. Através dessa informação é fornecido informação base para criação de testes.

8.4. Valor de Negócio [9]

É um valor que define o quão importante é o ***Backlog Item*** em relação aos outros itens, onde quanto maior o valor, mais importante é o item. Recomendável que use esse valor para definir a ordem de prioridade dos ***Backlog Items.***

8.5. Esforço

É um valor que define a quantidade de trabalho para completar o ***Backlog Item***. Maioria das metodologias ágeis recomendam o uso de uma sequência binária (1, 2, 4, 8...) ou de Fibonacci (1, 2, 3, 5, 8...), porém se pode usar qualquer valor numérico preferível pelo time. Idealmente o esforço deve representar uma equação: Complexidade / Capacidade do Time = Esforço.

9. Sprint

> "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." [Schwaber, Ken e Sutherland, Jeff. Guia do Scrum. https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf. Consultado em 30 de Julho de 2020. Pg. 8]

Uma sprint (em português: Arrancada) é um processo de duração fixa que pode variar entre 2 a 4 semanas, do qual se objetiva entregar um incremento potencialmente utilizável do projeto.

9.1. Reuniões

Sprints usam de reuniões diárias para que os integrantes estabeleçam 3 coisas: O que foi feito? O que será feito? E se há algum impedimento. Devem ser reuniões curtas e objetivas.

9.2. Planejamento (Sprint Panning)

O planejamento de uma ***Sprint*** deve ser feito em conjunto com a equipe e nele será definido o que será desenvolvido durante aquele período de tempo.

9.3. Sprint Backlog

A Sprint Backlog (em português: Pendências da Arrancada) seria uma lista dos artefatos a serem produzidos durante uma Sprint com base no conhecimento tecnológico e dos requisitos do projeto.

---

[8] Time Scrum: Grupo de pessoas compostas de Product Owner (Proprietário do Produto), Scrum Master (Mestre Scrum), Dev Team (Time de desenvolvimento).

[9] Nome Original: Business Value

10. O Incremento

O Incremento é o produto de todo o processo de uma ***Sprint*** consequentemente por uma iteração do ***Ciclo de vida*** do ***Scrum***. É Importante estabelecer que esse produto deve ser utilizável independente se ele estará disponível para uso geral dos usuários do produto.

11. Considerações Finais

O Scrum é um potente framework de projeto se usado adequadamente. Cada vez mais o mercado suporta ferramentas para organização de projetos Scrum, embora a metodologia Ágil não seja perfeita, ainda é muito interessante quando aplicada em cenários mais caóticos de projeto pela sua maleabilidade e fácil compreensão, vale também notar que a divisão de grandes tarefas/requisitos em tarefas menores torna a compreensão do que deve ser feito e como mais orgânica e de fácil execução o que não só é benéfico individualmente, mas mantêm o trabalho em equipe mais organizado e o produto desse trabalho, mais alinhado com as necessidades do cliente.

12. Referências

Albaladejo, Xavier. Agile Processes in Software Engineering and Extreme Programming. 12° Edição. 2011. Disponível em: https://books.google.com.br/books?id=hO2dYE9TFgYC&pg=PA307&dq=epic+feature+user+story+define&hl=pt-BR&sa=X&ved=2ahUKEwjf_Org8vfqAhXgJ7kGHW98AYQQ6AEwAHoECAAQAg#v=onepage&q&f=false. Acesso em 30 de julho de 2020.

Charles, Wilson. Chconley. Danielson, Steve. Jakobs, Mike. KathrinEE. Azure Boards Documentation: Define features and epics. Microsoft Docs. Disponível em: https://docs.microsoft.com/en-us/azure/devops/boards/backlogs/define-features-epics?view=azure-devops&tabs=agile-process. Acesso em 30/07/20.

Charles, Wilson. Chconley. Danielson, Steve. Jakobs, Mike. KathrinEE. Azure Boards Documentation: Define features and epics. Microsoft Docs. Disponível em: https://docs.microsoft.com/en-us/azure/devops/boards/backlogs/backlogs-overview?view=azure-devops. Acesso em 30/07/20.

Guay, Constantin. Scrum tips: Differences between epics, stories, themes and features. Publicado em: 26 de janeiro de 2018. Disponível em: https://const.fr/blog/agile/scrum-differences-epics-stories-themes-features/. Acesso em 30/07/20.

MCDONALD, KENT. Themes vs Epics vs Features vs User Stories, 2018. Disponível em: https://www.kbp.media/themes-epics-features-user-stories/. Acesso em 30/07/20.

Rubin, Keneth. Essential Scrum: A Practical Guide to the Most Popular Agile Process. Editora: Addison-Wesley Professional. Publicado em: 16 de julho de 2012.

REHKOPF, Max. Epics, Stories, Themes, and Initiatives. Disponível em: https://www.atlassian.com/agile/project-management/epics-stories-themes. Acesso em 30 de julho de 2020

Ventura, Plínio. Scrum tips: Differences between epics, stories, themes and features Epic, Feature e User Story (Epico, Funcionalidade e História de Usuário). Publicado em: 24 de maio de 2019. Disponível em: https://www.ateomomento.com.br/epic-feature-e-user-story/-features/. Acesso em 30 de julho de 2020.