

ScrumForce

O ScrumForce é uma metodologia de desenvolvimento de software baseada no Scrum, que funciona como container para práticas ágeis de Engenharia de Software e gerenciamento de projetos.

Definição do Scrum

Scrum é um *framework* leve que ajuda as pessoas, times e organizações a gerar valor por meio de soluções adaptativas para problemas complexos, de forma produtiva e criativa.

Scrum é:

- Leve
- Simples de entender
- Difícil de dominar

Scrum não é um processo, técnica ou um método prescritivo. Em vez disso, cria um espaço estruturado dentro do qual é possível empregar vários processos ou técnicas. O Scrum foi projetado para que se possa continuamente melhorar o produto, o time e o ambiente de trabalho.

Scrum é baseado no empirismo e *lean thinking*. O empirismo afirma que o conhecimento vem da experiência e da tomada de decisões com base no que é observado. O *lean thinking* reduz o desperdício e se concentra no essencial.

O Scrum emprega uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e o controle de riscos. O Scrum envolve grupos de pessoas que, coletivamente, possuem todas as habilidades e conhecimentos necessários para fazer o trabalho e compartilhar ou adquirir essas habilidades conforme necessário.

Scrum combina quatro eventos formais para inspeção e adaptação, contidos dentro de um evento, a Sprint. Esses eventos funcionam porque implementam os pilares empíricos do Scrum: transparência, inspeção e adaptação.

Transparência

Aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados. A transparência requer que estes aspectos tenham uma definição padrão comum para que os observadores compartilhem um mesmo entendimento comum do que está sendo visto.

Inspeção

Os praticantes do Scrum devem, frequentemente, inspecionar os artefatos e o progresso em direção ao objetivo da *Sprint* para detectar variações indesejadas. Esta inspeção não deve ser

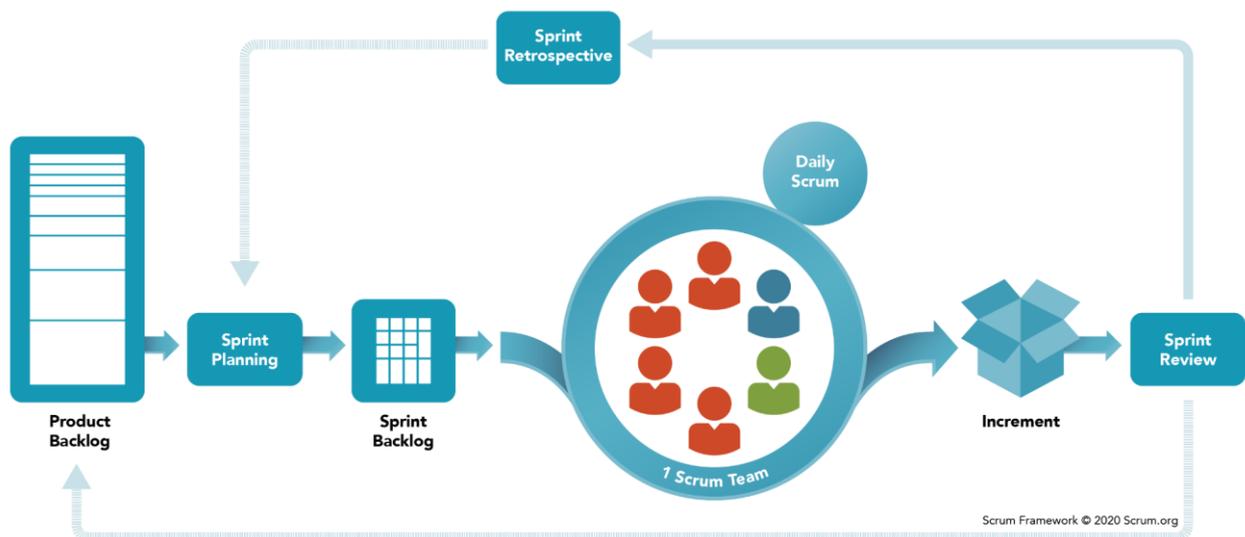
tão frequente que atrapalhe o objetivo dos trabalhos. As inspeções são mais benéficas quando realizadas de forma diligente por inspetores especializados no trabalho a se verificar.

Adaptação

Se um inspetor determina que um ou mais aspectos de um processo desviou para fora dos limites aceitáveis, e que o resultado do produto será inaceitável, o processo ou o material sendo produzido deve ser ajustado. O ajuste deve ser realizado o mais breve possível para minimizar mais desvios.

O Scrum prescreve quatro Eventos formais para inspeção e adaptação, como descrito na seção Eventos do Scrum deste documento.

- Planejamento da Sprint
- Reunião diária
- Revisão da Sprint
- Retrospectiva da Sprint



Os Valores do Scrum

Os valores do Scrum são:

1. Coragem: Os membros do Time Scrum devem ter coragem para fazer a coisa certa e trabalhar em problemas difíceis.
2. Foco: Todos devem focar no trabalho da Sprint e nos objetivos do Time Scrum.

3. Comprometimento: Comprometer-se pessoalmente a alcançar os objetivos do Time Scrum.
4. Respeito: Os membros do Time Scrum devem respeitar uns aos outros para serem capazes e independentes.
5. Abertura: O Time Scrum e as partes interessadas devem concordar em estar abertos a respeito de todo o trabalho e os desafios da execução desse trabalho.

Quando esses valores são incorporados e vividos pelo Time Scrum, os pilares do Scrum de **transparência, inspeção e adaptação** tornam-se vivos e constroem a confiança para todos, além de contribuir para o crescimento e amadurecimento do time. Os membros do Time Scrum aprendem e exploram estes valores à medida que trabalham com os eventos, papéis e artefatos do Scrum. O Sucesso no uso do Scrum depende das pessoas se tornarem mais proficientes na vivência destes cinco valores.

Os Papéis do Scrum

O Time Scrum

- O Time Scrum é composto pelo **Product Owner, o Scrum Master e os Desenvolvedores**.
- Times Scrum são auto-organizáveis e multifuncionais. Times auto-organizáveis escolhem qual a melhor forma para completarem seu trabalho, em vez de serem dirigidos por outros de fora do Time. Times multifuncionais possuem todas as competências necessárias para completar o trabalho sem depender de outros que não fazem parte da equipe. Esse modelo de time do Scrum é projetado para otimizar a flexibilidade, criatividade e produtividade.
- Times Scrum entregam produtos de forma iterativa e incremental, maximizando as oportunidades de *feedback*. Entregas incrementais de produto “**pronto**” garantem que uma versão **potencialmente funcional** do produto do trabalho esteja sempre disponível.

O Product Owner

O Product Owner, ou dono do produto, é o responsável por maximizar o valor do produto e do trabalho do Time de Desenvolvimento. Como isso é feito pode variar amplamente entre as organizações, Times Scrum e indivíduos.

O Product Owner é a única pessoa responsável por gerenciar o Backlog do Produto. O gerenciamento do Backlog do Produto inclui:

- Expressar claramente os itens do Backlog do Produto;
- Ordenar os itens do Backlog do Produto para alcançar melhor as metas e missões;
- Otimizar o valor do trabalho que o Time de Desenvolvimento realiza;

- Garantir que o Backlog do Produto seja visível, transparente, claro para todos, e mostrar no que o Time Scrum vai trabalhar a seguir; e,
- Garantir que o Time de Desenvolvimento entenda os itens do Backlog do Produto no nível necessário.

O Product Owner pode fazer o trabalho acima, ou delegar para o Time de Desenvolvimento fazê-lo, mas o Product Owner continua sendo o responsável por sua aprovação.

O Product Owner é uma pessoa e não um comitê. O Product Owner pode representar o desejo de um comitê no Backlog do Produto, mas aqueles que quiserem uma alteração nas prioridades dos itens de Backlog devem convencer o Product Owner.

Para que o Product Owner tenha sucesso, toda a organização deve respeitar as suas decisões. As decisões do Product Owner são visíveis no conteúdo e na priorização do Backlog do Produto. Ninguém deve forçar o Time de Desenvolvimento a trabalhar em um conjunto de requisitos diferente do que foi definido pelo Product Owner.

O Time de Desenvolvimento

O Time de Desenvolvimento consiste de profissionais que realizam o trabalho de entregar um incremento potencialmente liberável para o uso do produto “**pronto**” ao final de cada Sprint. Um incremento “**pronto**” é requerido na Revisão da Sprint.

Somente integrantes do Time de Desenvolvimento criam incrementos.

Os times de desenvolvimento são estruturados e autorizados pela organização para organizar e gerenciar seu próprio trabalho. A sinergia resultante faz com que a eficiência e a eficácia do Time de Desenvolvimento seja aprimorada como um todo a cada Sprint. Os Times de Desenvolvimento têm as seguintes características:

- Eles são auto organizados. Ninguém (nem mesmo o Scrum Master) diz ao Time de Desenvolvimento como transformar o Backlog do Produto em incrementos de funcionalidades potencialmente liberáveis para o uso;
- Times de Desenvolvimento são multifuncionais, possuindo todas as habilidades necessárias, enquanto equipe, para criar o incremento do Produto.
- O Scrum não reconhece títulos para os integrantes do Time de Desenvolvimento que não seja o **Desenvolvedor**, independentemente do trabalho que está sendo realizado pela pessoa; não há exceções para esta regra.
- O Scrum não reconhece sub-times no Time de Desenvolvimento, independente dos domínios específicos de conhecimento que precisam ser abordados, tais como teste, arquitetura, operação ou análise de negócios.
- Individualmente os integrantes do Time de Desenvolvimento podem ter habilidades especializadas e área de especialização, mas a responsabilidade pertence ao Time de Desenvolvimento como um todo.

Tamanho do Time de Desenvolvimento

O tamanho ideal do Time de Desenvolvimento é pequeno o suficiente para se manter ágil e grande o suficiente para completar uma parcela significativa do trabalho dentro dos limites da Sprint.

Menos de três integrantes no Time de Desenvolvimento diminui a interação e resulta em um menor ganho de produtividade. Times de desenvolvimento menores podem encontrar restrições de habilidades durante a Sprint, gerando um Time de Desenvolvimento incapaz de entregar um incremento potencialmente utilizável.

Havendo mais de nove integrantes é exigida muita coordenação. Times de Desenvolvimento grandes geram muita complexidade para um processo empírico gerenciar. Uma vez que o processo empírico se baseia na experiência e na tomada de decisões a partir do que é conhecido, uma comunicação eficiente é de extrema importância.

Em equipes muito grandes, é difícil saber o que cada um está fazendo. E a equipe fica mais lenta quando tenta descobrir. Em geral, a equipe se divide social e funcionalmente em subequipes que começam a trabalhar com objetivos cruzados. A multifuncionalidade se perde. As reuniões que deveriam levar minutos agora duram horas.

Os papéis de Product Owner e de Scrum Master não são incluídos nesta contagem, a menos que eles executem também o trabalho do Backlog da Sprint.

O Scrum Master

O Scrum Master é responsável por promover e disseminar o Scrum para que ele seja entendido e aplicado de maneira adequada, tal como foi projetado em sua essência. O Scrum Master faz isso ajudando todos a entenderem a teoria, as práticas, as regras e os valores do Scrum.

O Scrum Master é um líder-servidor para o Time Scrum. O Scrum Master ajuda aqueles que estão fora do Time Scrum a entender quais as suas interações com o Time Scrum são úteis e quais não são. O Scrum Master ajuda todos a mudarem estas interações para maximizar o valor criado pelo Time Scrum.

O Scrum Master trabalhando para o Product Owner

O Scrum Master serve o Product Owner de várias maneiras, incluindo:

- Encontrando técnicas para o gerenciamento efetivo do Backlog do Produto;
- Claramente comunicar a visão, objetivo e itens do Backlog do Produto para o Time de Desenvolvimento;
- Ensinar a Time Scrum a criar itens de Backlog do Produto de forma clara e concisa;
- Compreender a longo-prazo o planejamento do Produto no ambiente empírico;
- Compreender e praticar a agilidade; e,

- Facilitar os eventos Scrum conforme exigidos ou necessários.

O Scrum Master trabalhando para o Time de Desenvolvimento

O Scrum Master serve o Time de Desenvolvimento de várias maneiras, incluindo:

- Treinar o Time de Desenvolvimento em autogerenciamento e interdisciplinaridade;
- Ensinar e liderar o Time de Desenvolvimento na criação de produtos de alto valor;
- Remover impedimentos para o progresso do Time de Desenvolvimento;
- Facilitar os eventos Scrum conforme exigidos ou necessários; e,
- Treinar o Time de Desenvolvimento em ambientes organizacionais nos quais o Scrum não é totalmente adotado e compreendido.

O Scrum Master trabalhando para a Organização

O Scrum Master serve a Organização de várias maneiras, incluindo:

- Liderando e treinando a organização na adoção do Scrum;
- Planejando implementações Scrum dentro da organização;
- Ajudando os funcionários e demais partes interessadas a compreender e aprovar o Scrum e o desenvolvimento empírico de produtos;
- Provocando mudanças que aumentam a produtividade do Time Scrum; e,
- Trabalhando com outros Scrum Masters para aumentar a eficácia da aplicação do Scrum nas organizações.

Os Artefatos do Scrum

Scrum invoca transparência. Decisões para otimizar o valor e o controle de riscos são feitas com base na percepção do estado dos artefatos. Na medida em que a transparência é plena, essas decisões têm uma base sólida. Na medida em que os artefatos não são completamente transparentes, estas decisões podem ser falhas, valores podem diminuir e riscos podem aumentar.

O Scrum Master deve trabalhar com o Product Owner, o Time de Desenvolvimento, e outras partes envolvidas para entender se os artefatos estão plenamente transparentes. Há práticas para lidar com transparência incompleta e o Scrum Master deve ajudar todos a aplicar a prática mais apropriada na falta de uma transparência plena. O Scrum Master pode detectar transparência incompleta pela inspeção dos artefatos, percebendo padrões, ouvindo atentamente o que está sendo dito, e detectando diferenças entre o esperado e os resultados reais.

O papel do Scrum Master é trabalhar com o Time Scrum e com a organização para aumentar a transparência dos artefatos. Este trabalho geralmente envolve aprendizagem,

convencimento e mudança. Transparência não ocorre de um dia para o outro, mas é um caminho a ser percorrido.

Backlog do Produto

O Backlog do Produto é uma lista ordenada de tudo que é conhecido ser necessário no produto. É a única origem dos requisitos para qualquer mudança a ser feita no produto. O Product Owner é responsável pelo Backlog do Produto, incluindo seu conteúdo, disponibilidade e ordenação.

Um Backlog do Produto nunca está completo. Os primeiros desenvolvimentos estabelecem os requisitos inicialmente conhecidos e melhor entendidos. O Backlog do Produto evolui tanto quanto o produto e o ambiente no qual ele será utilizado. Ele é dinâmico e muda constantemente para identificar o que o produto necessita para ser mais apropriado, competitivo e útil. Se um produto existe, seu Backlog do Produto também existe.

O Backlog do Produto lista todas as características, funções, requisitos, melhorias e correções que formam as mudanças que devem ser feitas no produto nas futuras versões. Os itens do Backlog do Produto possuem os atributos de descrição, ordem, estimativa e valor. Os itens do Backlog geralmente incluem descrições de testes que comprovam sua completude quando "Prontos".

Enquanto um produto é usado e ganha valor, e o mercado fornece feedback, o Backlog do Produto torna-se uma lista maior e mais completa. Requisitos nunca param de mudar, então o Backlog do Produto é um artefato vivo. Mudanças nos requisitos de negócio, condições de mercado ou tecnologia podem causar mudanças no Backlog do Produto.

Múltiplos Times Scrum frequentemente trabalham juntos no mesmo produto. Um Backlog do Produto é usado para descrever o trabalho previsto para o produto. Um atributo do Backlog do Produto que agrupa itens pode ser então aplicado.

O refinamento do Backlog do Produto é a ação de adicionar detalhes, estimativas e ordem aos itens no Backlog do Produto. Este é um processo contínuo no qual o Product Owner e o Time de Desenvolvimento colaboram nos detalhes dos itens do Backlog do Produto. Durante o refinamento do Backlog do Produto, os itens são inspecionados e revisados. O Time de Scrum decide como e quando o refinamento está "Pronto". Este refinamento usualmente não consome mais de 10% da capacidade do Time de Desenvolvimento. Contudo, os itens do Backlog do Produto podem ser atualizados a qualquer momento pelo Product Owner ou a critério do Product Owner.

Os itens do Backlog do Produto de ordem mais alta (topo da lista) devem ser mais claros e mais detalhados que os itens de ordem mais baixa. Estimativas mais precisas são feitas baseadas em maior clareza e maior detalhamento; quanto menor a ordem na lista, menos detalhes. Os itens do Backlog do Produto que irão ocupar o Time de Desenvolvimento na próxima Sprint são mais refinados, de modo que todos os itens possam ser "Prontos" dentro do time-boxed da Sprint. Os itens do Backlog do Produto que podem ser "Prontos" pelo Time de Desenvolvimento dentro de uma Sprint são considerados "Preparados" para seleção no

Planejamento da Sprint. Itens do Backlog do Produto geralmente adquirem este grau de transparência através das atividades de refinamento descritas acima.

O Time de Desenvolvimento é responsável por todas as estimativas. O Product Owner deve influenciar o Time de Desenvolvimento, ajudando no entendimento e nas decisões conflituosas de troca, mas as pessoas que irão realizar o trabalho fazem a estimativa final.

Monitorando o Progresso a Caminho dos Objetivos

Em qualquer ponto do tempo, o total do trabalho restante para alcançar o objetivo pode ser somado. O Product Owner acompanha o total do trabalho restante pelo menos a cada Revisão da Sprint. O Product Owner compara este valor com o trabalho restante nas Revisões das Sprints anteriores, para avaliar o progresso na direção de completar o trabalho previsto pelo tempo desejado para alcançar o objetivo. Esta informação deve ser transparente para todas as partes interessadas.

Várias práticas para prever tendências foram usadas para prever o progresso, tais como gráficos burndown¹ ou fluxos cumulativos. Estas têm se provado úteis. Contudo, não substituem a importância do empirismo. Em ambientes complexos, o que acontecerá é desconhecido. Somente o que já ocorreu pode ser usado para uma tomada de decisão a respeito do que virá.

Backlog da Sprint

O Backlog da Sprint é um conjunto de itens do Backlog do Produto selecionados para a Sprint, juntamente com o plano para entregar o incremento do produto e atingir o objetivo da Sprint. O Backlog da Sprint é a previsão do Time de Desenvolvimento sobre qual funcionalidade estará no próximo incremento e sobre o trabalho necessário para entregar essa funcionalidade em um incremento "Pronto".

O Backlog da Sprint torna visível todo o trabalho que o Time de Desenvolvimento identifica como necessário para atingir o objetivo da Sprint. Para garantir melhoria contínua, é incluído no mínimo um item de prioridade alta sobre melhoria do processo identificado na última Reunião de Retrospectiva.

O Backlog da Sprint é um plano com detalhes suficientes para que as mudanças no progresso sejam entendidas durante a Reunião Diária. O Time de Desenvolvimento modifica o Backlog da Sprint ao longo de toda a Sprint, e o Backlog da Sprint vai surgindo durante a Sprint. Este surgimento ocorre quando o Time de Desenvolvimento trabalha segundo o plano e aprende mais sobre o trabalho necessário para atingir o objetivo da Sprint.

Sempre que um novo trabalho é necessário, o Time de Desenvolvimento adiciona-o ao Backlog da Sprint. Conforme o trabalho é realizado ou completado, a estimativa do trabalho restante é atualizada. Quando elementos do plano são considerados desnecessários, eles são removidos. Somente o Time de Desenvolvimento pode alterar o Backlog da Sprint durante

¹ Gráfico que representa o desenvolvimento do Backlog da Sprint no decorrer do tempo: no eixo Y, a quantidade de trabalho restante a ser realizada em pontos; no eixo X, o tempo em dias.

a Sprint. O Backlog da Sprint é altamente visível, uma imagem em tempo real do trabalho que o Time de Desenvolvimento planeja completar durante a Sprint, e que pertence exclusivamente ao Time de Desenvolvimento.

Monitorando o Progresso da Sprint

Em qualquer ponto do tempo na Sprint, o total do trabalho remanescente dos itens do Backlog da Sprint 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.

Incremento

O incremento é a soma de todos os itens do Backlog do Produto 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. O incremento deve estar na condição de ser utilizado independente do Product Owner decidir por liberá-lo ou não.

Transparência dos artefatos

Scrum invoca transparência. Decisões para otimizar o valor e o controle de riscos são feitas com base na percepção existente do estado dos artefatos. Na medida em que a transparência é plena, essas decisões têm uma base sólida. Na medida em que os artefatos não são completamente transparentes, essas decisões podem ser falhas, valores podem diminuir e riscos podem aumentar.

O Scrum Master deve trabalhar com o Product Owner, Time de Desenvolvimento, e outras partes envolvidas para entender se os artefatos estão plenamente transparentes. Há práticas para lidar com transparência incompleta; o Scrum Master deve ajudar todos a aplicar a prática mais apropriada na falta de uma transparência plena. O Scrum Master pode detectar transparência incompleta pela inspeção dos artefatos, percebendo padrões, ouvindo atentamente o que está sendo dito, e detectando diferenças entre o esperado e os resultados reais.

O trabalho do Scrum Master é trabalhar com o Time Scrum e com a organização para aumentar a transparência dos artefatos. Este trabalho geralmente envolve aprendizagem, convencimento e mudança. Transparência não ocorre de um dia para o outro, mas é um caminho.

Definição de "Pronto"

Quando um item do Backlog do Produto ou um Incremento é descrito como "Pronto", todos devem entender o que o "Pronto" significa. Embora, isso possa variar por Time Scrum, os integrantes devem ter um entendimento compartilhado do que significa o trabalho estar

completo, assegurando a transparência. Esta é a “Definição de Pronto” para o Time Scrum e é usada para descrever quando o trabalho está realmente completo no Incremento do produto.

A mesma definição orienta o Time de Desenvolvimento no conhecimento de quantos itens do Backlog do Produto podem ser selecionados durante o Planejamento da Sprint. O propósito de cada Sprint é entregar incrementos de funcionalidades potencialmente liberáveis para o uso que aderem à definição atual de “Pronto” do Time Scrum.

O Time de Desenvolvimento entrega um Incremento de funcionalidade do produto a cada Sprint. Por que este incremento é utilizável, o Product Owner pode escolher liberá-lo para o uso imediatamente. Se a definição de “Pronto” para um incremento é parte das convenções, padrões ou diretrizes de desenvolvimento da organização, todos os Times Scrum devem segui-la como um mínimo.

Se “Pronto” para um incremento não é uma convenção de desenvolvimento da organização, o Time de Desenvolvimento do Time Scrum deve definir uma definição de “pronto” apropriada para o produto. Se há múltiplos Times Scrum trabalhando no produto, os Times de Desenvolvimento de todos os Times Scrum devem definir mutuamente uma definição de “Pronto”.

Cada Incremento é adicionado a todos os Incrementos anteriores e completamente testado, garantindo que todos os Incrementos funcionam juntos.

Como um Time Scrum maduro, é esperado que a sua definição de “Pronto” seja expandida para incluir critérios mais rigorosos de alta qualidade. Note que novas definições podem descobrir trabalho a ser feito em Incrementos “Prontos” anteriormente.

Todo produto deve ter uma definição de “Pronto” que é um padrão para qualquer trabalho feito sobre ele. A definição de “Pronto” evita retrabalho.

Eventos do Scrum

Os eventos prescritos são usados no Scrum para criar uma regularidade e minimizar a necessidade de reuniões não definidas no Scrum. Todos os eventos são *time-boxed*, ou seja, tem início, uma duração máxima e fim. Uma vez que a Sprint começa, sua duração é fixada e não pode ser reduzida ou aumentada. Os eventos restantes podem terminar sempre que o propósito do evento é alcançado, garantindo que uma quantidade adequada de tempo seja gasta não permitindo desperdícios no processo.

Além da *Sprint*, que é um container para outros eventos, cada evento no Scrum é uma oportunidade de inspecionar e adaptar alguma coisa. Estes eventos são especificamente projetados para permitir uma transparência e inspeção criteriosa. Falhas na inclusão de qualquer um destes eventos resultará na redução da transparência e na perda de oportunidades para inspecionar e adaptar.

A Sprint

O evento principal do Scrum é a *Sprint*, um período de um mês ou menos, durante o qual é criado um incremento de produto potencialmente liberável para o uso. Sprints têm durações coerentes durante um esforço de desenvolvimento. Uma nova Sprint inicia imediatamente após a conclusão da Sprint anterior.

As Sprints são compostas por uma reunião de planejamento da Sprint, reuniões diárias, o trabalho de desenvolvimento, uma revisão da Sprint e a retrospectiva da Sprint.

Durante a Sprint:

- Não são feitas mudanças que possam colocar em perigo o objetivo da Sprint;
- As metas de qualidade não diminuem; e,
- O escopo pode ser esclarecido e renegociado entre o Product Owner e o Time de Desenvolvimento à medida que se aprende mais sobre o produto em construção.

Cada Sprint pode ser considerada um projeto com horizonte não maior que um mês. Como os projetos, as Sprints são utilizadas para realizar algo. Cada Sprint tem a definição do que é para ser construído, um plano flexível que irá guiar a construção, o trabalho e o incremento do produto resultante.

Sprints são limitadas a um mês corrido. Quando o horizonte da Sprint é muito longo, a definição do que será construído pode mudar, a complexidade pode aumentar e o risco pode crescer. Sprints permitem previsibilidade que garante a inspeção e adaptação do progresso em direção à meta pelo menos a cada mês corrido. As Sprints também limitam o risco ao custo de um mês corrido.

Cancelamento da Sprint

Uma Sprint pode ser cancelada antes que ela termine. Somente o Product Owner tem a autoridade para cancelar a Sprint, embora ele possa fazer isso sob influência das partes interessadas, do Time de Desenvolvimento ou do Scrum Master.

A Sprint poderá ser cancelada se o objetivo da Sprint se tornar obsoleto e se ela não fizer mais sentido às circunstâncias dadas. Isto pode ocorrer se a organização mudar sua direção ou se mudarem as condições do mercado ou das tecnologias. No entanto, devido a curta duração da Sprint, raramente cancelamentos acontecem.

Quando a Sprint é cancelada, qualquer item de Backlog do Produto completado e "Pronto" é revisado. Se uma parte do trabalho estiver potencialmente utilizável, tipicamente o Product Owner o aceita. Todos os itens de Backlog do Produto incompletos são re-estimados e colocados de volta no Backlog do Produto.

O cancelamento de Sprint consome recursos, já que todos têm que se reagrupar em outra reunião de planejamento da Sprint para iniciar outra Sprint. Cancelamentos de Sprints são frequentemente traumáticos para o Time Scrum, e são muito incomuns.

Planejamento da Sprint

O trabalho a ser realizado na Sprint é planejado na reunião de Planejamento da Sprint. Este plano é criado com o trabalho colaborativo de todo o Time Scrum.

O Planejamento da Sprint é um período com no máximo oito horas para uma Sprint de um mês de duração (ou seja, duas horas para cada semana da Sprint). Para Sprints menores, este evento é usualmente menor. O Scrum Master garante que o evento ocorra e que os participantes entendam seu propósito. O Scrum Master ensina o Time Scrum a manter-se dentro dos limites definidos para que a reunião aconteça.

O planejamento da Sprint responde às seguintes questões:

- O que pode ser entregue como Incremento resultante da próxima Sprint?
- Como o trabalho necessário para entregar o Incremento será realizado?

1. O que pode ser entregue nesta Sprint?

O Time de Desenvolvimento trabalha para prever as funcionalidades que serão desenvolvidas durante a Sprint. O Product Owner discute com o Time Scrum o objetivo que a Sprint deveria alcançar e os itens do Backlog do Produto que, se completados na Sprint, podem atingir a meta da Sprint. Todo o Time Scrum colabora na compreensão do trabalho a ser realizado na Sprint.

Os insumos para esta reunião são o Backlog do Produto, o Incremento do produto mais recente, a capacidade projetada do Time de Desenvolvimento durante a Sprint e o desempenho passado do Time de Desenvolvimento. O número de itens selecionados no Backlog do Produto para a Sprint depende exclusivamente do Time de Desenvolvimento, que são os únicos que podem avaliar o que pode ser realizado na próxima Sprint.

Durante o Planejamento da Sprint, o Time Scrum também determina a Meta da Sprint. A Meta da Sprint é o objetivo que será satisfeito dentro da Sprint através da implementação do Backlog do Produto, e que fornece a orientação para o Time de Desenvolvimento sobre o porquê de se construir o Incremento.

2. Como o trabalho escolhido será feito?

Tendo definido a Meta da Sprint e selecionado os itens do Backlog do Produto para a Sprint, o Time de Desenvolvimento decide como irá construir essas funcionalidades durante a Sprint e transformá-las em um Incremento de produto "Pronto". Os itens de Backlog do Produto selecionados para a Sprint, junto com o plano de entrega destes itens é chamado de **Backlog da Sprint**.

O Time de Desenvolvimento frequentemente começa desenhando o sistema e o trabalho necessário para converter o Backlog do Produto em um incremento funcional do produto. O trabalho pode ser de tamanho variável ou esforço estimado. No entanto, é planejado um trabalho suficiente durante o Planejamento da Sprint para o Time de Desenvolvimento prever o que este acredita que poderá fazer na próxima Sprint. O trabalho planejado pelo Time de

Desenvolvimento para os primeiros dias da Sprint é decomposto até o final desta reunião, frequentemente em unidades de um dia de duração ou menos. O Time de Desenvolvimento se auto-organiza para realizar todo o trabalho do **Backlog da Sprint**, tanto durante o Planejamento da Sprint e conforme necessário em toda a Sprint.

O Product Owner pode ajudar a esclarecer os itens de Backlog do Produto selecionados e fazer trocas. Se o Time de Desenvolvimento determina que tem excesso ou falta de trabalho, os itens do Backlog da Sprint podem ser renegociados com o Product Owner. O Time de Desenvolvimento também pode convidar outras pessoas para participar desta reunião para fornecer opinião técnica ou de domínios específicos.

No final do Planejamento da Sprint, o Time de Desenvolvimento deve ser capaz de explicar ao Product Owner e ao Scrum Master como pretende trabalhar como time auto-organizado para completar a Meta da Sprint e criar o incremento previsto.

Meta da Sprint

A Meta da Sprint é um objetivo definido para a Sprint que pode ser satisfeito através da implementação de itens do Backlog do Produto. Ela fornece orientação para o Time de Desenvolvimento sobre o porquê de estar construindo o Incremento. Este é criado durante a reunião de Planejamento da Sprint. A Meta da Sprint dá ao Time de Desenvolvimento alguma flexibilidade a respeito das funcionalidades implementadas na Sprint. Os itens do Backlog do Produto selecionados entregam uma função coerente, que pode ser a Meta da Sprint. A Meta da Sprint pode ser qualquer outra coerente que faça o Time de Desenvolvimento trabalhar em conjunto e não em iniciativas separadas.

Conforme o Time de Desenvolvimento trabalha, eles mantêm a meta da Sprint em mente. A fim de satisfazer a meta da Sprint, implementando funcionalidade e tecnologia. Caso o trabalho acabe por ser diferente do esperado pelo Time de Desenvolvimento, então eles colaboram com o Product Owner para negociar o escopo do Backlog da Sprint dentro da Sprint.

Reunião diária

A Reunião Diária do Scrum é um evento de 15 minutos para o Time de Desenvolvimento e é realizada todos os dias da Sprint. Nela o Time de Desenvolvimento planeja o trabalho para as próximas 24 horas. Isso otimiza a colaboração e a performance do time através da inspeção do trabalho desde a última Reunião Diária, e da previsão do próximo trabalho da Sprint. A Reunião Diária é mantida no mesmo horário e local todo dia para reduzir a complexidade de se usar Scrum (e facilitar a manutenção do hábito da reunião diária).

O Time de Desenvolvimento usa a Reunião Diária para inspecionar o progresso em direção à Meta da Sprint e para inspecionar se o progresso tende na direção de completar o trabalho do Backlog da Sprint. A Reunião Diária aumenta a probabilidade de o Time de Desenvolvimento atingir a meta da Sprint. Todos os dias, o Time de Desenvolvimento deve entender como pretende trabalhar em conjunto, como um time auto-organizado, para completar a Meta da Sprint e criar o Incremento previsto até o final da Sprint.

A estrutura da reunião é definida pelo Time de Desenvolvimento e pode ser conduzida de diferentes formas, desde que mantenham o foco no progresso em direção à meta da Sprint. Alguns Times de Desenvolvimento vão utilizar perguntas, outros vão se basear em discussões. Segue um exemplo do que pode ser utilizado:

- O que eu fiz ontem que ajudou o Time de Desenvolvimento a atingir a meta da Sprint?
- O que eu farei hoje para ajudar o Time de Desenvolvimento atingir a meta da Sprint?
- Eu vejo algum obstáculo que impeça a mim ou o Time de Desenvolvimento no atingimento da meta da Sprint?

O Time de Desenvolvimento pode se encontrar imediatamente após a Reunião Diária para discussões detalhadas, ou para adaptar, ou replanejar o restante do trabalho da Sprint.

O Scrum Master assegura que o Time de Desenvolvimento tenha a reunião, mas o Time de Desenvolvimento é responsável por conduzir a Reunião Diária. O Scrum Master ensina o Time de Desenvolvimento a manter a Reunião Diária dentro do período de 15 minutos. Muitos times preferem realizar essa reunião em pé e próximos ao kanban na parede com o Backlog da Sprint.

A Reunião Diária é uma reunião interna do Time de Desenvolvimento. Se outros estiverem presentes, o Scrum Master deve garantir que eles não perturbem a reunião.

Reuniões Diárias melhoram as comunicações, 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 de Desenvolvimento. Esta é uma reunião chave para **inspeção e adaptação**.

A Reunião Diária não é **reporte de status** nem **microgestão**, porque a ideia não é ter controle sobre o trabalho das pessoas ou pressioná-las por resultados. O objetivo é compartilhar a gestão e a responsabilidade pelos resultados do Time Scrum. Por isso, o Time de Desenvolvedores também precisa desenvolver uma mentalidade de gestão para que o compartilhamento da gestão e da responsabilidade pelos resultados faça sentido para todos.

Revisão da Sprint

A Revisão da Sprint é realizada no final da Sprint para inspecionar o Incremento e adaptar o Backlog do Produto se necessário. Durante a Revisão da Sprint o Time Scrum e as partes interessadas colaboram sobre o que foi feito na Sprint. Com base nisso e em qualquer mudança no Backlog do Produto durante a Sprint, os participantes colaboram nas próximas coisas que podem ser feitas para otimizar valor. Esta é uma reunião informal, não uma reunião de status, e a apresentação do Incremento visa **obter feedback e promover a colaboração**.

Esta é uma reunião de no máximo uma hora de duração para cada semana de uma Sprint. Para Sprints menores, este evento é usualmente menor. O Scrum Master garante que o evento ocorra e que os participantes entendam o seu propósito. O Scrum Master ensina todos os envolvidos a manter a reunião dentro do tempo previamente definido.

Como a Revisão da Sprint é conduzida:

- Os participantes incluem o Time Scrum e os Stakeholders chave convidados pelo Product Owner;
- O Product Owner esclarece quais itens do Backlog do Produto ficaram “Prontos” e quais não ficaram “Prontos”;
- O Time de Desenvolvimento discute o que foi bem durante a Sprint, quais problemas ocorreram dentro da Sprint, e como estes problemas foram resolvidos;
- O Time de Desenvolvimento demonstra o trabalho que está “Pronto” e responde às questões sobre o incremento;
- O Product Owner discute o Backlog do Produto tal como está e projeta os prováveis metas e datas de entrega baseado no progresso até a data (se necessário);
- O grupo todo colabora sobre o que fazer a seguir, e é assim que a Revisão da Sprint fornece valiosas entradas para o Planejamento da Sprint subsequente;
- Revisão de como o mercado ou o uso potencial do produto pode ter mudado e o que é a coisa mais importante a se fazer a seguir; e,
- Revisão da linha do tempo, orçamento, recursos potenciais e mercado para os próximos lançamentos previstos de funcionalidade ou capacidade do produto.

O resultado da Revisão da Sprint é um Backlog do Produto revisado que define os prováveis itens de Backlog do Produto para a próxima Sprint. O Backlog do Produto pode também ser ajustado completamente para atender novas oportunidades.

Retrospectiva da Sprint

A Retrospectiva 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.

A Retrospectiva da Sprint ocorre depois da Revisão da Sprint e antes do planejamento da próxima Sprint. Esta é uma reunião de no máximo três horas para uma Sprint de um mês. Para Sprint menores, este evento é usualmente menor. O Scrum Master garante que o evento ocorra e que os participantes entendam seu propósito.

O Scrum Master garante que o evento seja positivo e produtivo. O Scrum Master ensina todos a manter o evento dentro do tempo previamente combinado. O Scrum Master participa da reunião como um membro auxiliar do time devido a sua responsabilidade pelo processo Scrum.

O propósito da Retrospectiva da Sprint é:

- Inspecionar como foi a última Sprint em relação às pessoas, aos relacionamentos, aos processos e às ferramentas;

- Identificar e ordenar os principais itens que foram bem e as potenciais melhorias;
- Avaliar se foram eficazes as ações de melhorias que foram propostas em Retrospectivas de Sprint anteriores;
- Criar um plano para implementar melhorias no modo que o Time Scrum faz seu trabalho;
- Comemorar vitórias.

O Scrum Master encoraja o Time Scrum a melhorar dentro do processo do framework do Scrum, seu processo de desenvolvimento e suas práticas para torná-lo mais efetivo e agradável para a próxima Sprint. Durante cada Retrospectiva da Sprint, o Time Scrum planeja formas de aumentar a qualidade do produto melhorando o processo de trabalho ou adaptando, se apropriado, a definição de “Pronto”, mas sem entrar em conflito com os padrões do produto ou da organização.

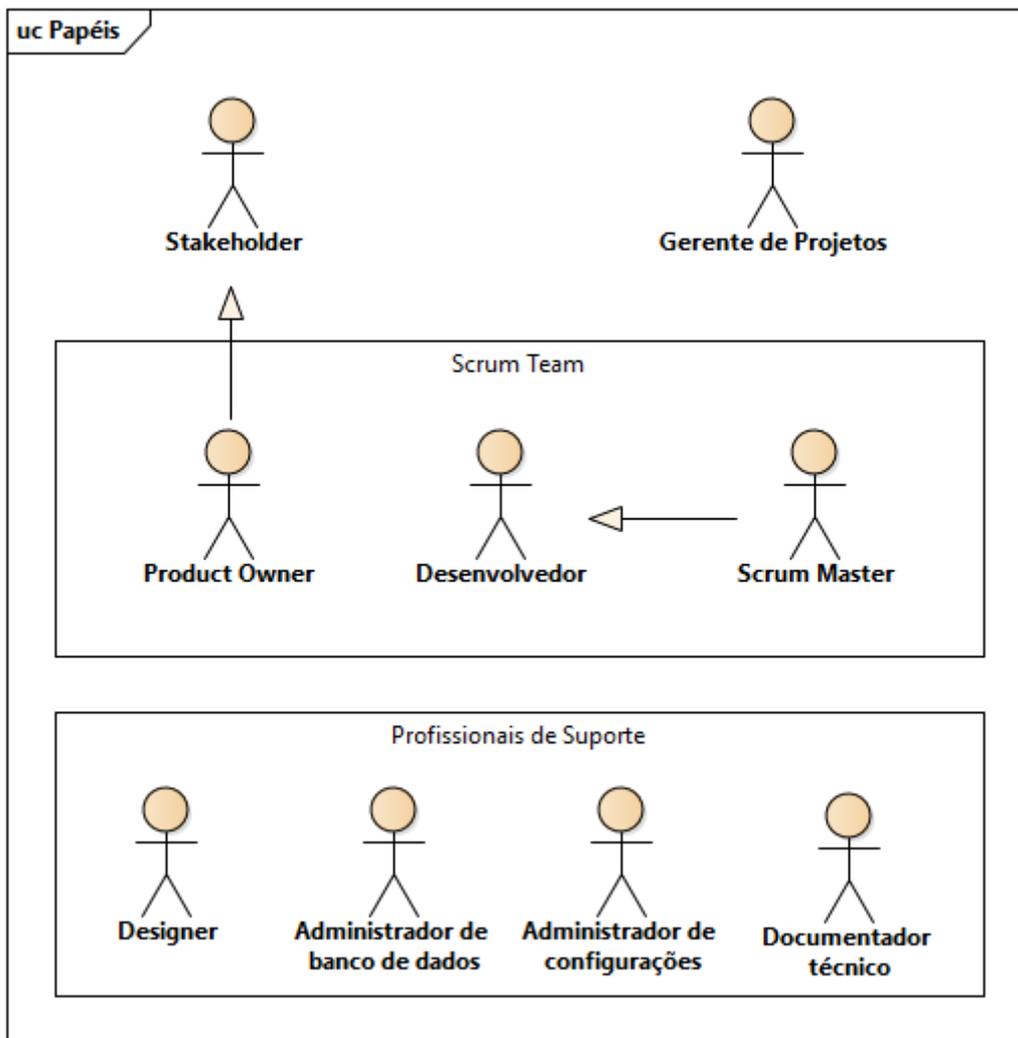
Ao final da retrospectiva da Sprint, o Time Scrum deverá ter identificado melhorias que serão implementadas na próxima Sprint. A implementação destas melhorias na próxima Sprint é a forma de adaptação à inspeção que o Time Scrum faz a si próprio. Apesar de que melhorias podem ser implementadas a qualquer momento, a Retrospectiva da Sprint fornece uma oportunidade formal para focar em inspeção e adaptação.

Práticas de Engenharia de Software

Papéis

O **ScrumForce** define o Time Scrum tal como está no Guia Scrum: Product Owner, Scrum Master e Desenvolvedores, sendo que o Scrum Master pode ser também um desenvolvedor. Mas são admitidos outros papéis que podem trabalhar junto do Time Scrum, de modo colaborativo. São profissionais como **designers, administradores de banco de dados, administradores de configurações e documentador técnico** que geralmente estão fora do Time Scrum e atendem sob demanda (veja a figura abaixo).

Mas o ponto a se considerar é o grau de dependência que o Time Scrum terá desses profissionais. Se a atuação deles estiver diretamente ligada à entrega e a sua disponibilidade para atendimento imediato não é garantida, talvez seja melhor que essas pessoas façam parte do Time Scrum como desenvolvedores. Isso garante que eles estejam não apenas envolvidos com o desenvolvimento do produto, mas também comprometidos com os resultados do Time Scrum.



Scrum Master versus Gerente de Projeto

De acordo com o PMBOK®, uma das principais referências no mundo quando se fala em gerenciamento de projetos, o *Gerente de Projeto* é responsável pelo que sua equipe produz. O Gerente do Projeto lidera a equipe do projeto para atender aos objetivos do projeto e às expectativas das partes interessadas e trabalha para equilibrar as restrições concorrentes no projeto com os recursos disponíveis.

O Gerente de Projeto também desempenha funções de comunicação entre o patrocinador do projeto, os membros da equipe e outras partes interessadas. Isso inclui fornecer orientação e apresentar a visão de sucesso do projeto.

Já o Guia Scrum™ define o Scrum Master como responsável por promover e apoiar o Scrum, ajudando todos a entender a teoria, práticas, regras e valores do Scrum.

O Scrum Master é um líder-servidor que apoia o time removendo impedimentos. O Scrum Master também ajuda as pessoas de fora do Time Scrum a entender quais de suas interações com a equipe Scrum são úteis e quais não são. O Scrum Master ajuda todos a mudar essas interações para maximizar o valor criado pelo Time Scrum.

Comparar Scrum Master com Gerentes de Projetos é como comparar goleiros do futebol com zagueiros do rugby. Quando o goleiro muda para o Rugby e começa a mapear o papel antigo no novo jogo, imagine o que vai acontecer? O goleiro começará a jogar futebol no rugby.

Enquanto nos métodos clássicos a gerência do projeto está concentrada em uma pessoa com muitas responsabilidades (inclusive de gerenciar o dia a dia das pessoas), no Scrum, todos compartilham a responsabilidade por gerenciar o projeto, cada um cuida de si e também uns dos outros - quase que eliminando a necessidade de gerenciar pessoas.

É possível que o Scrum Master acumule o papel de Gerente de Projeto. Contudo, é fundamental ter em mente que estas são funções diferentes, apesar de terem pontos em comum. E se o Scrum Master acumular o papel de Gerente de Projeto, poderia também acumular o de desenvolvedor? Teoricamente sim, mas a cada papel que se acumula aumenta-se o risco de a pessoa não ter tempo para desempenhar todos os papéis de maneira satisfatória.

Modelos de Sprint

O ScrumForce define três modelos de Sprint que comportam diferentes tipos de atividades de engenharia de software:

1. Design Sprint: atividades de levantamento e análise de requisitos, e desenho arquitetônico.
2. Build Sprint: atividades de construção do produto - desenho, implementação e testes.
3. Launch Sprint: atividades relacionadas à transição do produto para o ambiente de produção - testes beta e operação piloto.

Design Sprint

Na Design Sprint, são conhecidas as necessidades dos usuários e os conceitos do negócio. Os requisitos são identificados e analisados, e o produto é especificado o suficiente para modelar o domínio do problema e subsidiar o planejamento da construção do produto. São realizadas as seguintes atividades:

1. Levantamento de requisitos
 - a. **Abertura do projeto:** Definir de forma sucinta o cenário atual e o cenário futuro desejável, as partes interessadas, premissas e restrições, riscos e marcos preliminares.
 - b. **Definição do contexto:** Levantar os aspectos dos processos de negócio ou de um sistema maior que sejam relevantes para a determinação dos requisitos do produto.
 - c. **Identificação dos requisitos funcionais e não-funcionais:** Delimitar os problemas que o produto irá resolver. Produzir uma lista de todos os

requisitos funcionais e não-funcionais, descritos de forma sucinta, ainda sem entrar em detalhes.

- d. **Definição dos requisitos funcionais:** Definir as funcionalidades do produto na forma de histórias de usuário, identificar atores (papéis que os usuários desempenham nas histórias), descrever os critérios de aceitação, que podem ser regras de negócio, campos da tela de usuário e, se necessário, a descrição passo a passo de fluxos de eventos.
- e. **Análise do contexto:** (1) Caracterizar os usuários, as tarefas que eles realizam e os papéis que eles desempenham; o ambiente em que o produto será utilizado; e os produtos semelhantes ou concorrentes. (2) Investigar as características das tarefas realizadas pelos usuários e a forma como essas tarefas são realizadas visando a obtenção de informação que vai ajudar no desenho de uma interface mais adequada para a realização das tarefas com o apoio do produto. (3) Produzir recomendações para o desenho da interface com o usuário.
- f. **Prototipação dos requisitos:** Para as funcionalidades mais importantes, criar protótipos de interface para ajudar a definir os requisitos e compreender aspectos de informação, estilos de interação e de navegação da interface. Caso necessário, criar protótipos para testar tecnologia e explorar os riscos associados.
- g. **Detalhamento dos requisitos não-funcionais:** Detalhar os requisitos de desempenho, usabilidade e outros aspectos considerados necessários para que o produto atinja a qualidade desejada.
- h. **Criação da Estrutura Analítica do Projeto (EAP):** Criar a Estrutura Analítica do Projeto (EAP) organizando os requisitos funcionais e não-funcionais a fim de facilitar o desenho das liberações, o acompanhamento do progresso do projeto, a comunicação e o envolvimento das partes interessadas. Classificar os requisitos em relação a valor, benefícios e prioridade relativa para o cliente. Estimar também a estabilidade e a complexidade de implementação dos requisitos.
- i. **Inspeção dos requisitos:** Determinar se os requisitos satisfazem os critérios de qualidade definidos nas listas de conferência, e se a Especificação dos Requisitos do Software está clara e bem entendida por todas as partes interessadas nos tópicos referentes ao levantamento de requisitos.

2. Análise de requisitos

- a. **Análise de histórias de usuário:** (1) Identificar as classes do produto, com base nas histórias de usuário, descobertas obtidas nas entrevistas, leitura de documentos relacionados ao negócio e testes de protótipos do levantamento de requisitos. (2) Organizar as classes em pacotes lógicos e lhes atribuir estereótipos, dependendo do papel que desempenham no modelo. (3)

Determinar os relacionamentos de vários tipos que podem existir entre os objetos das classes identificadas.

- b. **Realização de histórias de usuário:** (1) Realizar os épicos e as histórias de usuário mais complexos por meio do desenho de diagramas de interação, para facilitar e até mesmo ampliar o seu entendimento. (2) Levantar os atributos e os métodos que fazem parte do conceito expresso pela classe.
- c. **Inspeção da análise:** Determinar se o Modelo do Problema satisfaz os critérios de qualidade de modelo e se a Especificação de Requisitos do Software está clara e bem entendida por todas as partes interessadas nos tópicos referentes à análise de requisitos.

3. Desenho

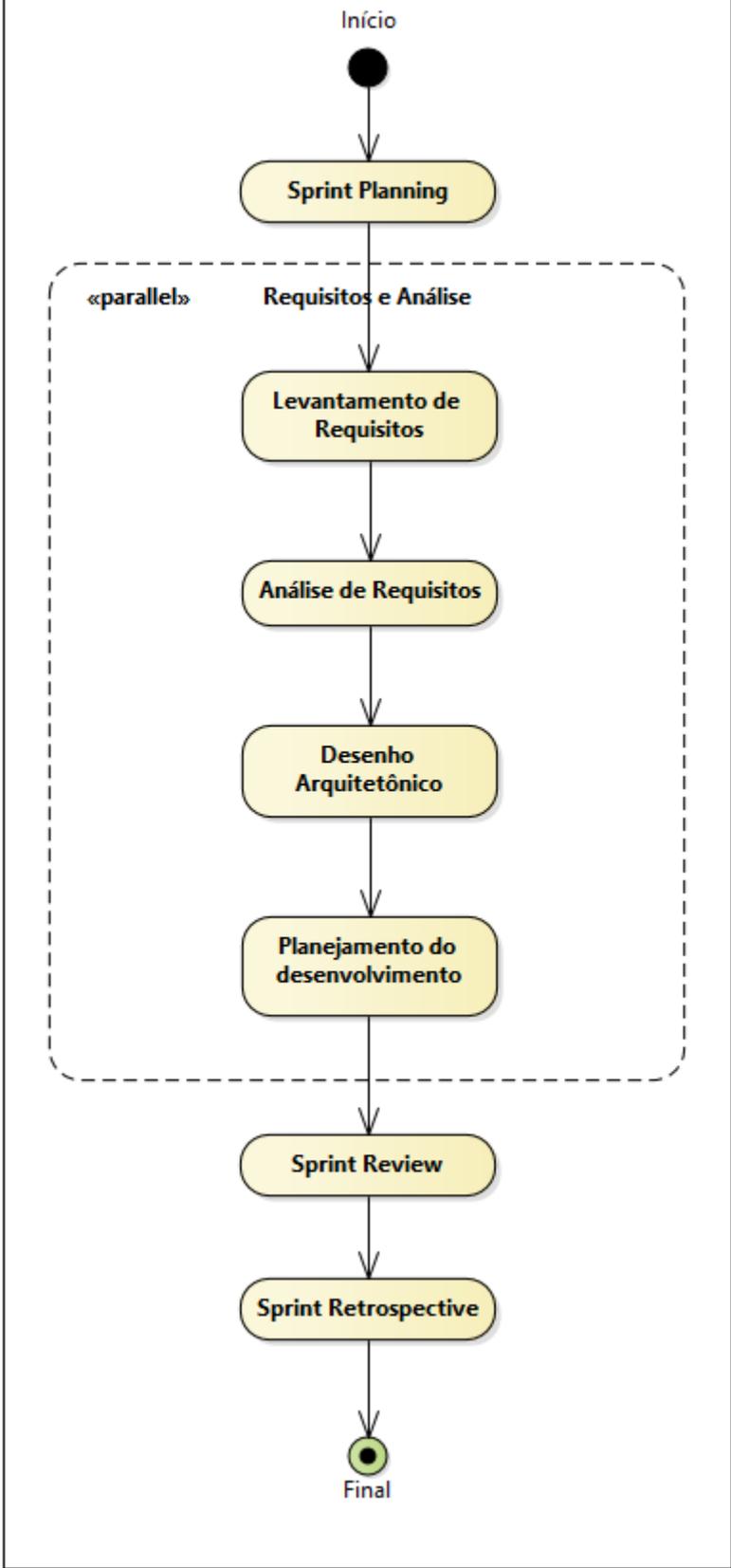
- a. **Desenho arquitetônico (preliminar):** Definições externa e interna dos componentes do produto de software, em nível suficiente para se decidir as principais questões de arquitetura e tecnologia e permitir o planejamento das liberações.

4. Planejamento do desenvolvimento

- a. **Estimativa de tamanho:** Estimar o tamanho do produto com alguma medida que atenda aos seguintes critérios: ser contável por meio de um procedimento bem definido; ser calculável a partir das informações contidas na Especificação de Requisitos do Software; apresentar uma boa correlação com o esforço de desenvolvimento.
- b. **Estimativa de esforço:** Estimar o esforço a partir da estimativa de tamanho e da produtividade registrada em bases históricas de projetos semelhantes. Caso os Desenvolvedores não tenham experiência com métodos de estimativa de tamanho de software ou as bases históricas não tenham dados significativos, pode-se usar Planning Poker e ajustar as estimativas no decorrer das Sprints.
- c. **Estimativa de recursos:** (1) Estimar tanto os recursos descritos na Especificação de Requisitos do Software quanto os necessários para o desenvolvimento e para a implantação do produto. (2) Estimar atividades de treinamento para suprir as competências requeridas para o desenvolvimento do software.
- d. **Estimativa de cronograma:** (1) Definir as relações de dependências entre as funcionalidades para estabelecer a ordem de desenvolvimento, contudo, considerando-se a prioridade apontada pelo Product Owner. (2) Estimar o cronograma levando-se em consideração o tempo efetivamente disponível para o projeto, os feriados, as férias, as folgas compensatórias, etc.
- e. **Estimativa de riscos:** (1) Identificar os riscos individuais em relação ao projeto, bem como as fontes de risco geral do projeto e documentar suas

características; (2) Analisar qualitativamente esses riscos, avaliando-lhes a probabilidade e o provável impacto; (3) Priorização dos riscos, organizando-os de acordo com a probabilidade e o impacto; (4) Planejar respostas aos riscos.

act Design Sprint



Build Sprint

Na Build Sprint é desenhada, implementada e testada uma versão completamente operacional do produto, que atende aos requisitos especificados.

São realizadas as seguintes atividades:

1. Desenho

- a. **Desenho arquitetônico:** Resolver os aspectos estratégicos de desenho externo e interno, e escolher as tecnologias mais adequadas para o desenvolvimento do produto, considerando aspectos de testabilidade. Desenhar como os testes serão implementados.
- b. **Detalhamento das histórias de usuários:** Para as funcionalidades mais críticas, detalhar as histórias de usuário o suficiente para que elas sejam realizáveis por meio de componentes de software e interfaces de usuário.
- c. **Desenho das interfaces:** Desenhar em detalhes as interfaces reais do produto na forma definitiva de implementação, levando-se em consideração a Especificação de Requisitos do Software em relação a requisitos não-funcionais de usabilidade e recomendações resultantes de prototipação.
- d. **Desenho das entidades:** Transformar as classes de entidade do Modelo do Problema nas classes correspondentes no Modelo da Solução.
- e. **Desenho da persistência:** Desenhar as estruturas externas de armazenamento persistente, como arquivos e bancos de dados (esquemas, tabelas, visões, restrições de integridade, etc.).
- f. **Realização das histórias de usuário:** Para as funcionalidades mais críticas, determinar como os objetos das classes do Modelo da Solução vão colaborar para a realização das histórias de usuário.
- g. **Desenho da estrutura do Manual do Usuário:** Rever a visão do produto, suas interfaces com o usuário, suas funcionalidades e as tarefas em que será usado. Caracterizar o público do Manual do Usuário para selecionar estilo de linguagem, estrutura e nível de detalhe da documentação. Definir a estrutura do Manual do Usuário.

2. Implementação e testes

- a. **Desenho interno, implementação e testes da história de usuário:** (1) Analisar os aspectos de eficiência e decidir se as unidades são críticas para o desempenho do produto, ou decidir por tratar esses aspectos posteriormente, em uma etapa de otimização do produto, desde que não seja necessário refatorar partes significativas do software, de modo a comprometer a sua arquitetura, além de prazo e orçamento do projeto; (2) Desenhar e implementar os testes de integração; (3) Desenhar e implementar os detalhes do comportamento das unidades: pré-condições, pós-condições,

diagramas de estado (se necessário), regras de negócio e critérios de aceitação, algoritmos e tratamentos de erros; (4) Integrar as unidades e os testes implementados com o trabalho dos outros desenvolvedores e produzir uma versão preliminar do produto. (5) Desenhar, implementar e executar testes de aceitação do produto por meio da extensão dos testes de integração das histórias de usuário que foram implementadas.

- b. **Redação do Manual de Usuário:** Redigir os tópicos do Manual de Usuário referentes aos itens que foram implementados.

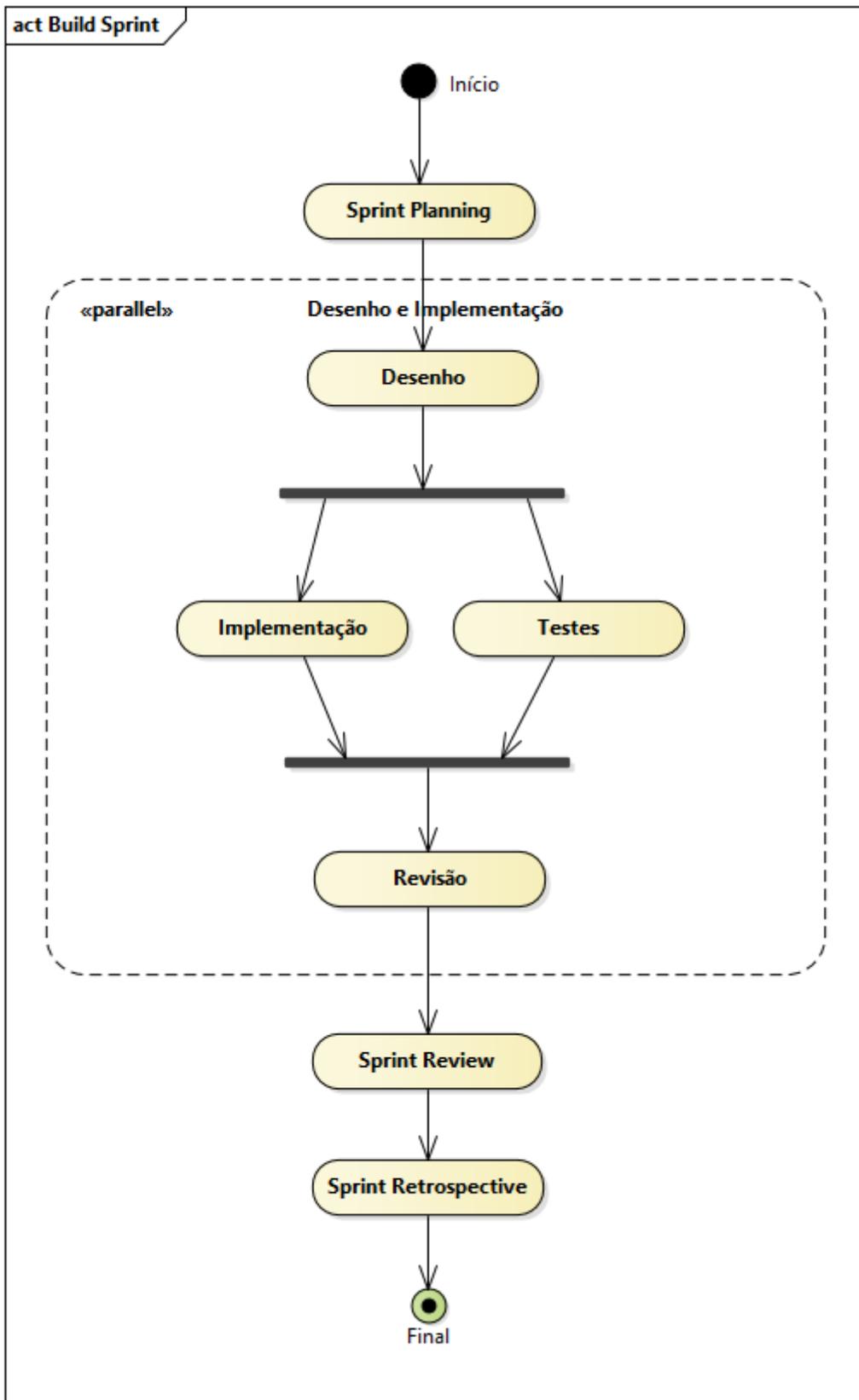
3. Revisão do desenvolvimento

- a. **Inspeção de usabilidade:** Inspeção das interfaces de usuário em relação aos requisitos não-funcionais de Usabilidade. Encaminhar para correção os defeitos encontrados em ordem de prioridade e gravidade. Verificar se as correções aplicadas eliminaram os defeitos apontados na inspeção.
- b. **Inspeção do desenho externo:** Inspeção dos artefatos de acordo com lista de conferência de Desenho Externo. Encaminhar para correção os defeitos encontrados em ordem de prioridade e gravidade. Verificar se as correções aplicadas eliminaram os defeitos apontados na inspeção.
- c. **Inspeção de desenho interno:** Inspeção dos artefatos de acordo com lista de conferência de Desenho Interno.
- d. **Inspeção dos testes:** Inspeção dos artefatos de acordo com lista de conferência de Testes.
- e. **Inspeção da implementação:** Inspeção dos artefatos de acordo com lista de conferência de Implementação.
- f. **Revisão técnica do Manual do Usuário:** (1) Revisão da consistência do Manual do Usuário em relação ao produto que foi implementado. (2) Revisão da clareza, completeza e conformidade com recomendações de outras normas aplicáveis.

4. Monitoramento do desenvolvimento

- a. **Monitoração do escopo:** Registrar as alterações feitas nos requisitos e medir a estabilidade destes.
- b. **Medição do progresso:** Comparar os valores adquiridos reais e os valores planejados para tamanho, esforços e prazo. A comparação entre uma dessas medidas de valor adquirido real e o respectivo valor planejado permite projetar a provável evolução do projeto.
- c. **Monitoração dos riscos:** (1) Verificar as premissas estipuladas no termo de abertura do projeto; (2) Monitorar os riscos em relação ao cenário atual do projeto; (3) Verificar se as políticas e os procedimentos de gerenciamento

dos riscos estão sendo seguidos; (4) Verificar a eficácia das respostas aos riscos; (5) Manter os registros de riscos atualizados.



Launch Sprint

Na Launch Sprint, o produto é colocado à disposição de uma comunidade de usuários para testes finais, treinamento e uso inicial.

São realizadas as seguintes atividades:

1. Testes beta

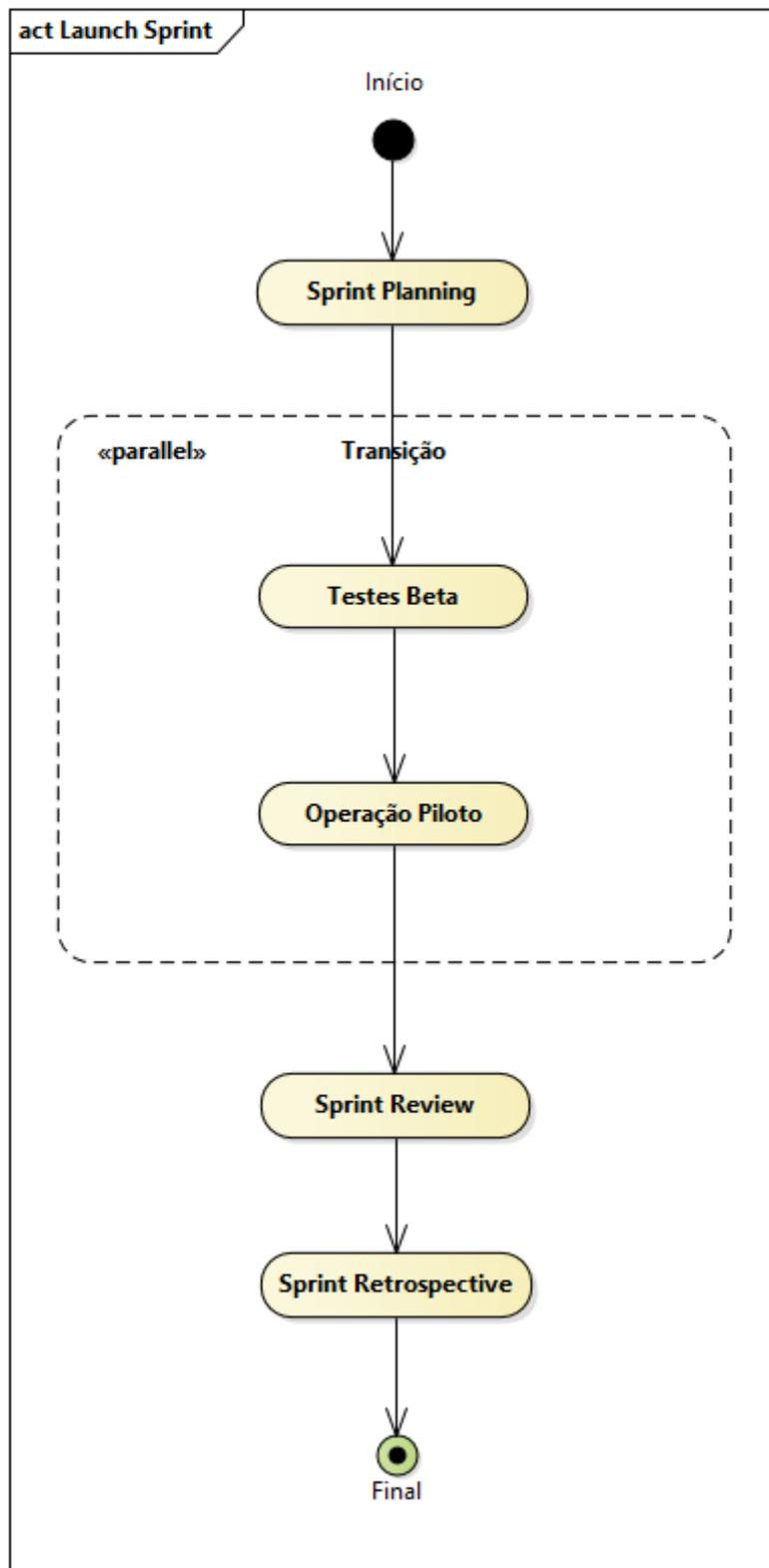
- a. **Implantação do produto em ambiente de homologação:** Realizar os procedimentos para implantação do produto em ambiente de homologação para que sejam realizados os testes de aceitação pelo cliente.
- b. **Realização de testes de aceitação do produto:** Realizar os testes de aceitação em ambiente de homologação para que o cliente ateste a conformidade do produto com os requisitos que foram especificados e acordados.
- c. **Revisão técnica do Manual do Usuário:** Realizar uma revisão de apresentação do Manual do Usuário com os usuários-chave.

2. Operação piloto

- a. **Implantação do produto em ambiente de produção:** Realizar os procedimentos para implantação do produto em ambiente de produção para uso inicial pelo cliente.
- b. **Treinamento dos usuários:** Capacitar os usuários para utilizarem o produto.
- c. **Monitoramento do desempenho do produto:** Monitorar o desempenho do produto, ajustar configurações, corrigir defeitos, se necessário.

3. Encerramento do projeto

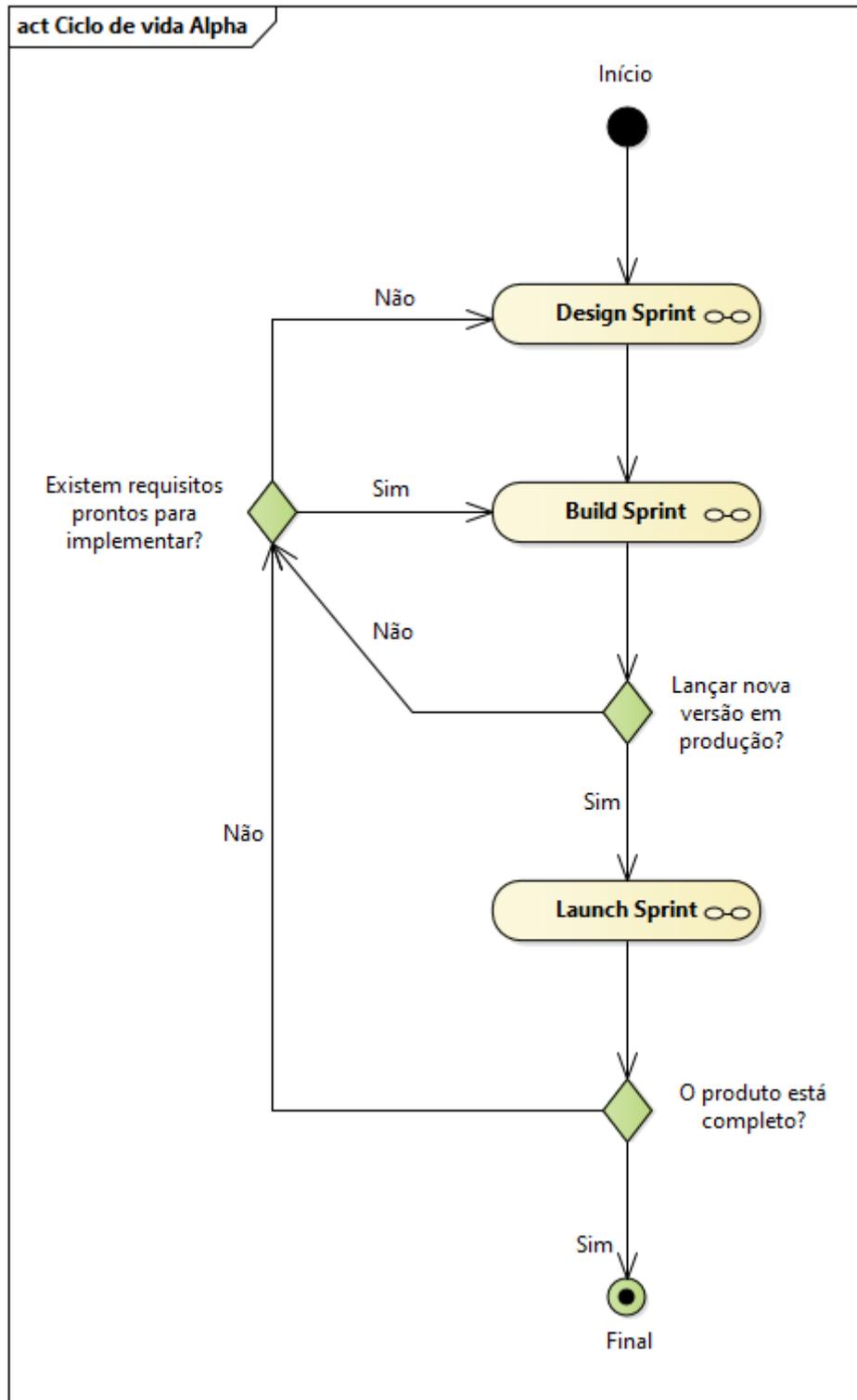
- a. **Monitoramento dos riscos:** (1) Verificar as premissas estipuladas no termo de abertura do projeto; (2) Monitorar os riscos frente ao cenário atual do projeto; (3) Verificar se as políticas e os procedimentos de gerenciamento dos riscos estão sendo seguidos; (4) Verificar a eficácia das respostas aos riscos; (5) Manter os registros de riscos atualizados.
- b. **Encerramento do projeto:** (1) Verificar se todas as atividades foram finalizadas; (2) Verificar se todos os artefatos estão armazenados no sistema de controle de versão; (3) Liberar recursos para serem utilizados em outros empreendimentos; (4) Formalizar o encerramento do projeto.



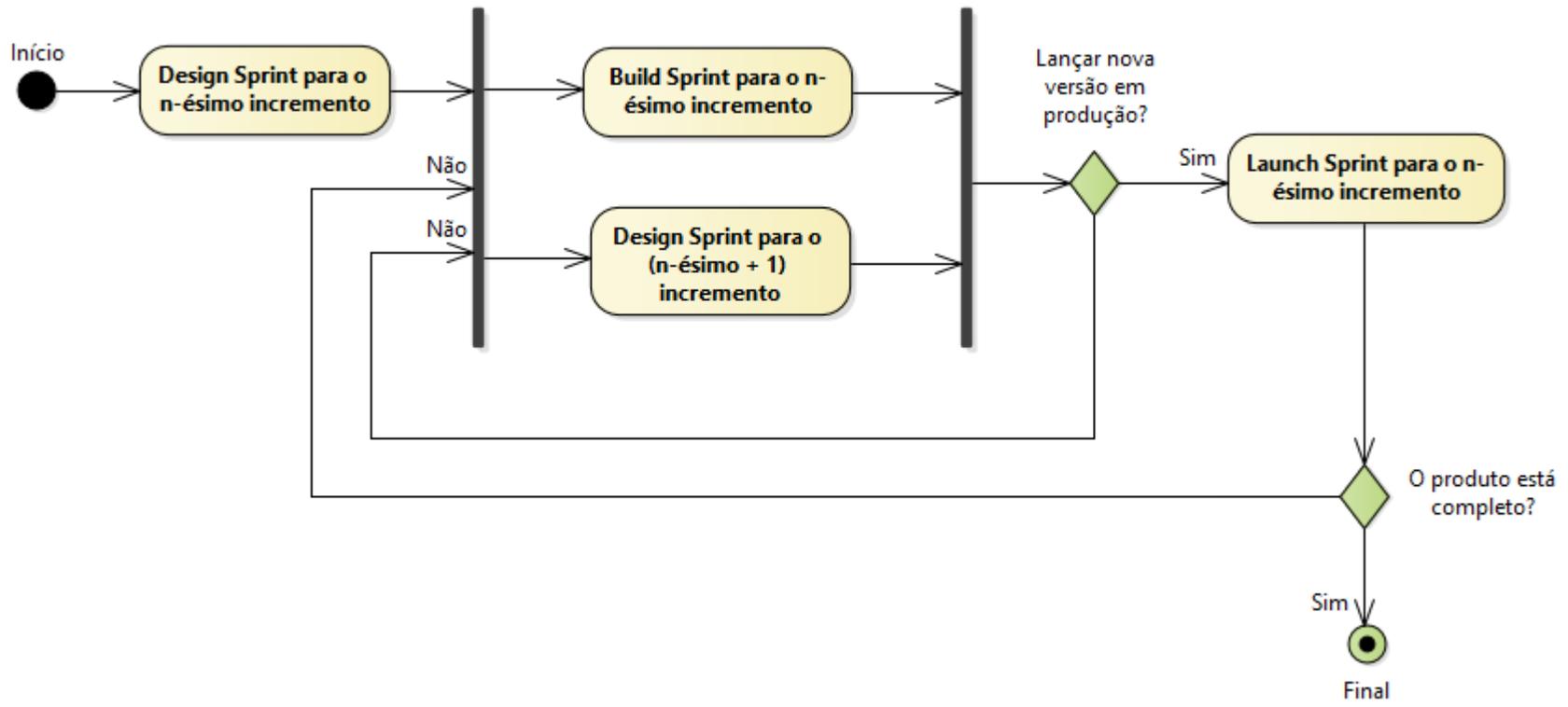
Modelos de Ciclo de Vida

O ScrumForce define dois modelos de ciclo de vida a serem escolhidos conforme a disponibilidade de desenvolvedores na equipe:

- **O Ciclo de Vida Alpha** é mais adequado às equipes pequenas em que provavelmente todos participam da *Design Sprint*, *Build Sprint* e *Launch Sprint*.
- **O Ciclo de Vida Beta** é interessante quando existem desenvolvedores suficientes para se ter dois times para trabalhar em paralelo e colaborativamente, um atuando nas Design Sprints e outro nas Build Sprints.



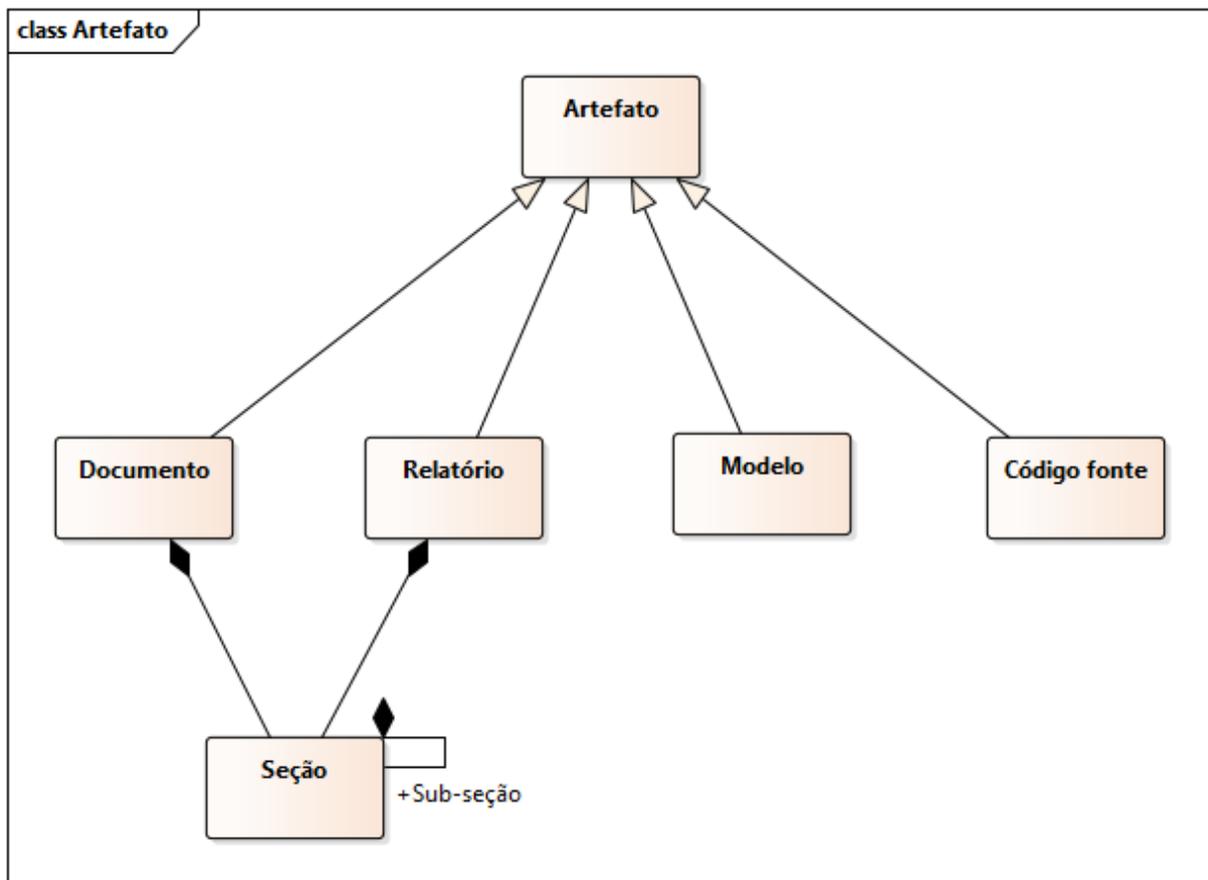
act Ciclo de vida Beta



Artefatos

Um artefato é um conjunto de informações que são produzidas, modificadas ou usadas em um processo, além de ser sujeito a controle de versão. Artefatos são os elementos tangíveis de um projeto e são usados como insumos pelo Time Scrum para desempenhar uma atividade e são também resultados ou saídas de outras atividades. [KROLL]

O Scrum define os seguintes artefatos: o **Product Backlog**, o **Sprint Backlog** e o **Incremento**. Além desses, o *ScrumForce* define outros artefatos que podem ser do tipo *Documento*, *Relatório*, *Modelo* ou *Código-fonte*.



Os artefatos do tipo *Documento* e *Relatório* podem ser produzidos manualmente por meio de editores de texto ou planilhas eletrônicas, ou podem ser extraídos automaticamente de ferramentas de software, que é o desejável principalmente para relatórios.

Já para os artefatos do tipo *Modelo* e *Código-fonte*, existem softwares específicos para a produção deles, como ferramentas de design e ambientes integrados de programação.

Outros artefatos podem ser documentados de maneira mais informal, como a captura de mensagens de e-mail e fotografias de quadros brancos.

Para documentar a descrição de requisitos funcionais, adota-se como padrão as **Histórias de Usuários**, com agrupamentos em **Épicos** e **Temas**, por serem mais flexíveis e mais

próximas da linguagem do negócio. Mas também é aceito o formato de **Casos de Uso**, cuja utilização é recomendada apenas para funcionalidades mais complexas.

Seguem abaixo as tabelas de artefatos sugeridos e de relacionamento entre as atividades e os artefatos.

Artefatos sugeridos

Sigla	Descrição	Disciplina
Documentos		
CDU	Caso de Uso: Documento que descreve requisitos funcionais na forma de caso de uso.	Requisitos, Análise e Desenho
DTSw	Descrição dos Testes do Software: Documento que descreve, de forma detalhada, os planos e especificações dos testes que serão executados.	Testes
ERSw	Especificação dos Requisitos do Software: Documento que delimita o escopo inicial e descreve o conjunto de requisitos especificados para o produto de software, em detalhes suficientes para se iniciar a construção dos componentes mais críticos do produto.	Requisitos, Análise e Desenho
HDU	História de Usuário: Documento que descreve requisitos funcionais na forma de história de usuário.	Requisitos, Análise e Desenho
MUSw	Manual do Usuário do Software: Documento que serve de referência para uso do produto por parte dos usuários.	Implementação
*PDSw	Plano de Desenvolvimento do Software: Documento que descreve, de forma detalhada, os compromissos assumidos em relação ao projeto, quanto a recursos, custos, prazos, riscos e outros aspectos gerenciais.	Gestão de Projetos
PLSw	Plano de Lançamento do Software: Documento que descreve em detalhes como implantar a versão atual do produto em ambiente de produção e como será comunicado o lançamento dessa versão.	Desenho e Implementação
*PQSw	Plano da Qualidade do Software: Documento que descreve, de forma detalhada, os procedimentos de garantia da qualidade que serão adotados no projeto.	Gestão da Qualidade
README	Arquivo leia-me do Software: Documento com informações sobre como compilar e executar o produto, bem como executar a bateria de testes automatizados.	Implementação e Testes

Release Notes	Notas de Lançamento: Documento com informações relevantes para o lançamento de uma versão do produto; deve destacar as novidades da versão e as melhorias e correções de defeitos que estão incluídas.	Implementação
RNG	Regras de Negócio Gerais: Documento que descreve as regras de negócio que são válidas para todos os componentes do produto.	Requisitos, Análise e Desenho
TAE	Termo de Aceitação de Entrega: Documento que formaliza o aceite da entrega parcial ou final do produto, considerando-a em conformidade com os requisitos.	Gestão de Projetos
TAP	Termo de Abertura de Projeto: Documento que formaliza o início do projeto.	Gestão de Projetos
TEP	Termo de Encerramento de Projeto: Documento que formaliza o encerramento do projeto.	Gestão de Projetos
Modelos		
MPSw	Modelo do Problema: Modelo que detalha os conceitos do domínio do problema a resolver que sejam relevantes para a validação dos requisitos.	Análise
MSSw	Modelo da Solução: Modelo que detalha a estrutura lógica e física do produto, em termos de seus componentes.	Desenho
Burndown	Gráficos Burndown: Gráficos que mostram os pacotes de trabalho em pontos que ficam prontos no decorrer da Sprint.	Gestão de projetos
EAP	Estrutura Analítica do Projeto: Decomposição das entregas do projeto em pacotes de trabalho (histórias de usuários ou casos de uso).	Requisitos, Gestão de Projetos
Relatórios		
RDSw	Relatório de Inspeção de Desenho do Software: Relatório dos defeitos encontrados na inspeção dos artefatos de desenho em relação à Especificação de Requisitos, com as devidas justificativas e categorização.	Desenho
RISw	Relatórios de Inspeção de Implementação de Software: Relatório dos defeitos encontrados na inspeção dos artefatos de desenho e implementação em relação às listas de conferência de desenho interno e implementação e à Especificação de Requisitos e Modelo da Solução, com as devidas justificativas e categorização.	Implementação
RTSw	Relatórios dos Testes do Software: Relatório dos defeitos encontrados nos testes do software em relação à	Testes

	Especificação de Requisitos do Software, com as devidas justificativas e categorização.	
RUSw	Relatório de Inspeção de Usabilidade: Relatório dos defeitos encontrados na inspeção das interfaces de usuário em relação às heurísticas de usabilidade, com as devidas justificativas e categorização.	Usabilidade
Código		
CFSw	Código-fonte do Software	Implementação
CESw	Código Executável do Software	Implementação

* Artefatos que serão definidos nas próximas versões do ScrumForce.

Relacionamento entre as atividades e os artefatos

Atividade	Insumos	Resultados	Disciplina
Design Sprint			
Abertura do projeto			
Abertura do projeto	Entrevistas com o cliente; documentos do negócio; textos normativos relacionados aos problemas a serem resolvidos.	TAP	Gestão de Projetos
Levantamento de Requisitos			
Definição do contexto	Entrevistas e oficinas de requisitos com usuários; observação do trabalho dos usuários; documentos do negócio; textos normativos relacionados com os problemas a serem resolvidos.	ERSw	Requisitos, Usabilidade
Identificação dos requisitos funcionais e não-funcionais	Entrevistas e oficinas de requisitos com usuários; observação do trabalho dos usuários; documentos do negócio; textos normativos relacionados com os problemas a serem resolvidos.	ERSw/Product Backlog	Requisitos

Definição dos requisitos funcionais	ERSw/Product Backlog	ERSw (Histórias de usuário), MPSw (preliminar)	Requisitos
Análise do contexto	Anotações das entrevistas, oficinas de requisitos com usuários, observação do trabalho dos usuários e testes de usabilidade; documentos do negócio; textos normativos relacionados com os problemas a serem resolvidos.	ERSw (atores, definições adicionais sobre requisitos funcionais e não-funcionais, principalmente relacionados com usabilidade)	Requisitos, Usabilidade
Prototipação dos requisitos	ERSw	Protótipos das interfaces	Requisitos, Usabilidade, Implementação
Detalhamento dos requisitos não-funcionais	ERSw	ERSw	Requisitos, Usabilidade
Criação da Estrutura Analítica do Projeto (EAP)	ERSw/Product Backlog	EAP	Requisitos, Gestão de Projetos
Inspeção dos requisitos	ERSw/Product Backlog	Relatório da inspeção dos requisitos, ERSw/Product Backlog (corrigido)	Requisitos
Análise de requisitos			
Análise de histórias de usuário	ERSw	MPSw	Análise
Realização de histórias de usuário	ERSw	MPSw (modificado)	Análise
Inspeção da análise	MPSw	MPSw (corrigido)	Análise

Desenho			
Desenho arquitetônico (preliminar)	MPSw	MPSw (modificado)	Desenho
Planejamento do desenvolvimento			
Estimativa de tamanho	ERSw, MPSw	PDSw (Estimativa de tamanho)	Gestão de Projetos
Estimativa de esforço	ERSw, MPSw, PDSw (Estimativa de tamanho)	PDSw (Estimativa de esforço)	Gestão de Projetos
Estimativa de recursos	ERSw, MPSw	PDSw (Estimativa de recursos)	Gestão de Projetos
Estimativa de cronograma	ERSw, PDSw (Estimativas de tamanho, esforço e recursos)	PDSw (Estimativa de cronograma)	Gestão de Projetos
Estimativa de riscos	ERSw, PDSw, histórico dos projetos anteriores, experiência do Time Scrum	PDSw (Estimativa de Riscos)	Gestão de Projetos
Build Sprint			
Desenho			
Desenho arquitetônico	ERSw, EAP, Product Backlog, MPSw	MSSw, CFSw, DTSw	Desenho e Testes
Desenho da estrutura do Manual do Usuário	ERSw, EAP, Product Backlog	MUSw (Estrutura do manual)	Desenho e Implementação

Desenho das interfaces	ERSw, Protótipos das interface	CFSw (Interfaces de usuário)	Desenho e Implementação
Detalhamento das histórias de usuários	ERSw (Histórias de usuário)	ERSw (Histórias de usuário)	Desenho
Desenho das entidades	MPSw	MSSw, CFSw	Desenho e Implementação
Desenho da persistência	MSSw, CFSw	CFSw	Desenho
Realização das histórias de usuário	MSSw	CFSw	Desenho
Inspeção de usabilidade	CESw, ERSw, Heurísticas de Usabilidade	Relatório da inspeção de usabilidade, CFSw (Corrigido)	Usabilidade, Desenho e Implementação
Inspeção do desenho externo	MSSw, CFSw, LC-Desenho-Externo	Relatório de inspeção de desenho externo (Defeitos, Informação sobre correção e validação da correção), CFSw (corrigido), MSSw (corrigido)	Desenho e implementação
Implementação e testes			
Desenho interno, implementação e teste de história de usuário	ERSw, MSSw, CFSw	MSSw, CFSw	Implementação e Testes
Redação do Manual do Usuário	ERSw, MSSw, CESw	MUSw (Tópicos dos itens que foram implementados)	Implementação
Revisão da implementação e dos testes			

Inspeção de desenho interno	ERSw, MSSw, CFSw, LC-Desenho-Interno	Relatório de inspeção de desenho interno (Defeitos, Informação sobre correção e validação da correção), MSSw (Corrigido), CFSw (Corrigido)	Desenho e Implementação
Inspeção da implementação	ERSw, MSSw, CFSw, LC-Implementação	Relatório de inspeção de implementação (Defeitos, Informação sobre correção e validação da correção), MSSw (Corrigido), CFSw (Corrigido)	Implementação
Inspeção dos testes	ERSw, MSSw, CFSw, LC-Testes	Relatório de inspeção dos testes (Defeitos, Informação sobre correção e validação da correção), CFSw (Corrigido)	Implementação e Testes
Revisão do Manual do Usuário	ERSw, CESw, MUSw	Relatório de inspeção do manual do usuário (Defeitos, Informação sobre correção e validação da correção)	Implementação
Monitoramento do desenvolvimento			
Monitoração do escopo	PDSw, EAP, Product Backlog, Sprint Backlog, Burndown Chart	PDSw	Gestão de Projetos
Medição do progresso	PDSw	PDSw	Gestão de Projetos

Monitoração dos riscos	PDSw, EAP, Product Backlog, Sprint Backlog, Burndown Chart, Relatórios de inspeção	PDSw	Gestão de Projetos
Launch Sprint			
Testes Beta			
Implantação do produto em ambiente de homologação	CFSw	PLSw, CESw (modo de homologação)	Implementação e Gestão de Mudanças
Realização de testes de aceitação do produto no ambiente de homologação	ERSw, MSSw, DTSw, CESw	Relatório dos testes de aceitação (Defeitos, Informação sobre correção e validação da correção), CFSw (Corrigido), TAE	Testes
Revisão técnica do Manual do Usuário	MUSw	Relatório de revisão técnica do Manual do Usuário, MUSw (corrigido)	Implementação
Operação Piloto			
Implantação do produto em ambiente de produção	PLSw, CFSw	CESw (modo de produção)	Implementação e Gestão de Mudanças
Treinamento dos usuários	MUSw	Lista de presença dos participantes	Implementação
Monitoramento do desempenho do produto	CESw	Relatório de defeitos encontrados e corrigidos, CFSw (corrigido)	Implementação

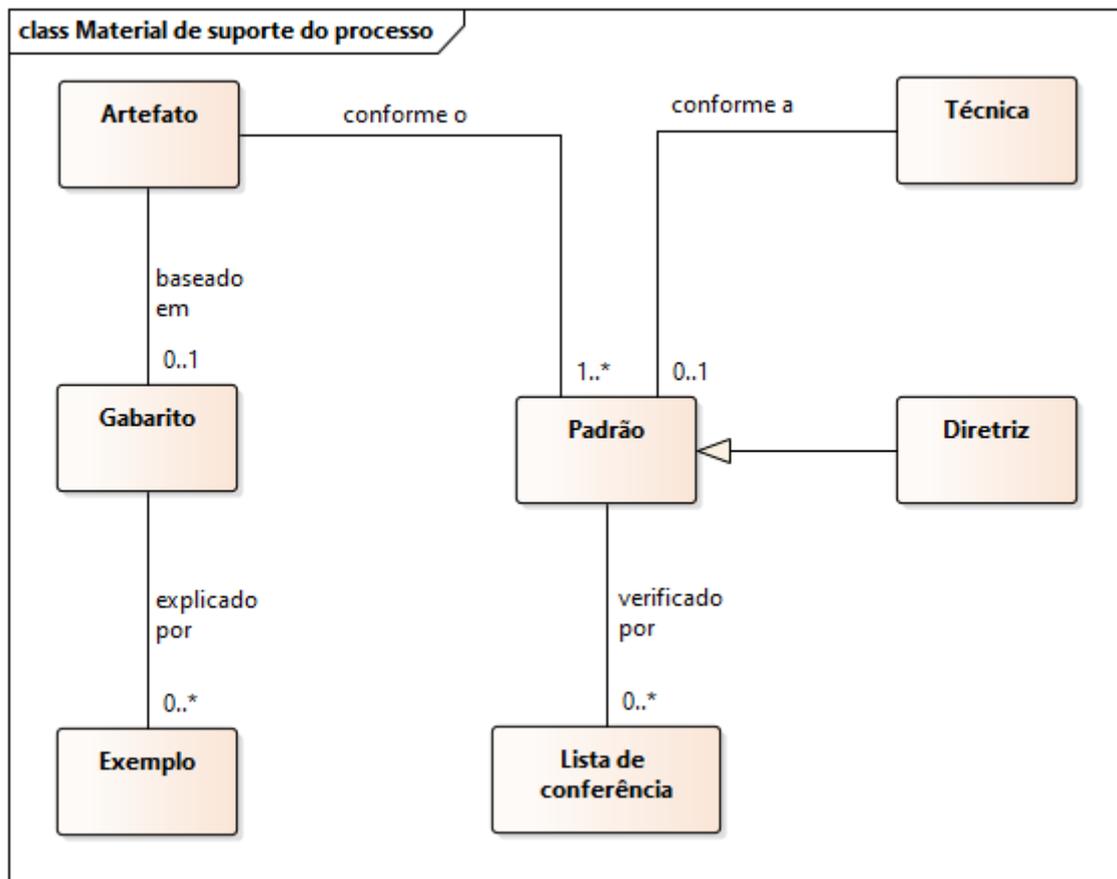
Encerramento do projeto			
Monitoração dos riscos	PDSw	PDSw	Gestão de Projetos
Encerramento do projeto	PDSw, EAP, TAP, TAEs, e outros documentos e relatórios relacionados ao projeto.	PDSw (atualizado), TEP	Gestão de Projetos

Material de suporte do método

O ScrumForce possui material de suporte para orientar o uso do método.

- Artefatos, como documentos e relatórios, devem ser criados com base em [gabaritos](#) (ou *templates*), que por sua vez podem ser explicados por meio de exemplos ou comentários explicativos em cada seção do gabarito.
- Os artefatos devem ser produzidos conforme os padrões estabelecidos, que são definidos de acordo com alguma técnica.
- Os padrões podem ser normas técnicas, convenções ou diretrizes. Verificação da qualidade dos artefatos quanto à aderência aos padrões deve ser feita por meio de [listas de conferência](#).
- As [diretrizes](#) fornecem orientação detalhada sobre a elaboração de artefatos, inclusive, por meio de exemplos.
- O Modelo do Processo apresenta os elementos do ScrumForce e seus relacionamentos graficamente por meio da UML (Unified Modeling Language).
- A Descrição do Processo detalha os principais conceitos do processo e estabelece o relacionamento entre as atividades sugeridas e os artefatos que são insumos e resultados das atividades.

O material de suporte do [ScrumForce](#) está disponível no [Codex](#), a base de conhecimento da Coordenadoria de Soluções Corporativas do TRE-MT.



Referências bibliográficas

SUTHERLAND, Jeff; SCHWABER, Ken. O Guia do Scrum - O Guia Definitivo para o Scrum: As Regras do Jogo. Novembro de 2020. Disponível em <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-2.0.pdf>.

PAULA FILHO, Wilson de Pádua. Engenharia de software; fundamentos, métodos e padrões. Segunda Edição. Editora LTC, 2003.

KROLL, Per; KRUCHTEN, Philippe. The Rational Unified Process Made Easy - A Practitioner's Guide to the RUP. Addison-Wesley, 2003.

PALMER, Stephen R.; FELSING, John M. A Practical Guide to Feature-Driven Development. Prentice Hall. 2002.