

Abordagem Ágil:

Papéis, responsabilidades, definições e conceitos

<i>Product Owner (PO)</i>	1
<i>Scrum Master</i>	1
<i>Equipe Ágil [multifuncionalidade Autogerenciada]</i>	2
<i>Agile Coach</i>	2
<i>Definição de Pronto (DoD) e Critério de Aceite (AC)</i>	5
<i>Sprint Zero</i>	7
<i>Tamanho da Sprint Zero:</i>	8
<i>O que NÃO deve ser feito:</i>	8
<i>Técnica MoSCoW (técnica de priorização)</i>	8
<i>Desafios de Medição em Projetos Híbridos ou Ágeis</i>	9

PAPEIS E RESPONSABILIDADE

PRODUCT OWNER (PO)

- **Papel:** O Product Owner é responsável por definir a visão do produto e garantir que o backlog do produto esteja priorizado com base no valor para o negócio e nos requisitos dos stakeholders. O PO é a ponte entre a equipe de desenvolvimento e os stakeholders, assegurando que o produto final atenda às necessidades do usuário final.
- **Responsabilidades:** Definir e priorizar o backlog do produto, especificar os critérios de aceitação para as user stories e garantir que a equipe de desenvolvimento entenda as necessidades do usuário.
- **User Stories:** Geralmente, é responsabilidade do Product Owner escrever as user stories ou garantir que elas sejam escritas de forma adequada, embora possa colaborar com stakeholders e a equipe para refiná-las.

SCRUM MASTER

- **Papel:** O Scrum Master atua como um facilitador para a equipe de desenvolvimento, assegurando que a equipe siga as práticas e princípios do Scrum. O Scrum Master ajuda a equipe a trabalhar de maneira eficaz dentro do framework Scrum.
- **Responsabilidades:** Facilitar reuniões (como o Daily Scrum, Sprint Planning, Sprint Review e Sprint Retrospective), ajudar a equipe a remover impedimentos e garantir uma colaboração efetiva.

- **User Stories:** Embora o Scrum Master possa ajudar a facilitar o processo de criação de user stories, especialmente durante o planejamento do sprint e as sessões de refinamento do backlog, normalmente *não é sua responsabilidade* escrevê-las.

EQUIPE ÁGIL [MULTIFUNCIONALIDADE | AUTOGERENCIADA]

- **Papel:** A equipe de desenvolvimento é o núcleo que transforma requisitos do projeto (como user stories) em incrementos de produto utilizáveis e de alta qualidade. Composta por profissionais de diversas áreas (desenvolvedores, testadores, designers, etc.), a equipe é responsável por todas as tarefas necessárias para entregar o produto, desde a análise até a implementação, testes e lançamento.
- **Responsabilidade:** A equipe colabora para definir os critérios específicos que devem ser atendidos para que uma tarefa ou história de usuário seja considerada completa, isso incluir definir is critério de pronto (**DoD**), alinhados com os critérios de aceite definidos pelo Product Owner (**PO**) inclui aspectos como qualidade, testes, documentação e funcionalidade. Garantir que o produto ou funcionalidade desenvolvida atenda aos padrões de qualidade estabelecidos.
- **User Stories:** Antes de um sprint, a equipe participa de sessões de refinamento de backlog para discutir, estimar e priorizar as user stories, garantindo que elas sejam claras, testáveis e viáveis dentro do contexto do sprint. Trabalha para transformar as user stories em incrementos de produto funcionais, aplicando a DoD como um guia para garantir que cada história esteja completa e pronta para entrega.

AGILE COACH

- **Papel:** O Agile Coach ajuda organizações, equipes e indivíduos a adotar e melhorar métodos ágeis em seus processos de trabalho. Este papel é mais sobre orientação e treinamento do que sobre gerenciamento direto de projetos.
- **Responsabilidades:** Fornecer treinamento e orientação sobre práticas ágeis, ajudar a equipe a superar desafios na adoção do ágil e trabalhar para melhorar a cultura e os processos da equipe de desenvolvimento.
- **User Stories:** O Agile Coach *não é diretamente responsável* por escrever user stories, mas pode orientar o Product Owner e a equipe sobre as melhores práticas para criá-las.

Portanto, dentre esses quatro papéis, o **Product Owner** é quem tem a responsabilidade primária de escrever as user stories, ou pelo menos garantir que elas sejam escritas de forma a refletir os requisitos do usuário e os objetivos do negócio.

REQUERIMENTO E REQUISITOS ÁGEIS

Na metodologia ágil, uma hierarquia padronizada acima de um épico pode incluir "**Tema -> Iniciativa -> Épico**". Um exemplo: :

- **Tema:** O nível mais alto de objetivo estratégico. Por exemplo, "Melhorar a experiência do usuário na plataforma de e-commerce."

- **Iniciativa:** Uma coleção de épicos relacionados que trabalham em direção a um objetivo mais amplo. Por exemplo, "Aprimorar o desempenho do site, otimizar o processo de checkout e personalizar as recomendações de produtos."
- **Épico:** Um grande volume de trabalho que pode ser dividido em histórias menores. Neste caso, um épico poderia ser "Implementar um novo motor de recomendação para sugestões de produtos personalizados."

Esta hierarquia ajuda na organização e priorização do trabalho, fornecendo uma estrutura clara para gerenciar projetos grandes e complexos dentro do framework ágil.

Esta estrutura é comumente usada no gerenciamento de projetos ágeis, incluindo em ferramentas como **Jira**, para **planejar roadmaps** para prazos mais longos e para gerenciar o trabalho em diferentes níveis de granularidade.

No contexto das metodologias ágeis, um **EPIC** (Épico) é um termo usado para descrever um grande corpo de trabalho que pode ser dividido em várias menores tarefas **User Story** (histórias de usuário) que são mais gerenciáveis e mais facilmente compreendidas.

Os Épicos são usados para agrupar trabalhos que compartilham um único objetivo amplo ou que são parte de uma grande funcionalidade ou objetivo do projeto. Eles são uma maneira de organizar e planejar o trabalho em níveis mais altos de abstração, permitindo que as equipes mantenham uma visão geral do projeto enquanto detalham e executam suas partes constituintes.

CARACTERÍSTICAS DE UM ÉPICO:

- **Amplo em Escopo:** Um Épico é tipicamente mais amplo e menos definido do que as histórias de usuário. Ele abrange uma grande funcionalidade ou objetivo do projeto que não pode ser completado em uma única iteração (sprint).
- **Divisível:** Épicos são divididos em histórias de usuário menores, que são tarefas mais específicas e gerenciáveis. Este processo ajuda a transformar um objetivo amplo em partes executáveis que podem ser priorizadas e agendadas em sprints.
- **Visão de Longo Prazo:** Ao trabalhar com Épicos, as equipes podem planejar e priorizar o trabalho que contribuirá para objetivos de longo prazo, garantindo que todos os aspectos de uma grande funcionalidade sejam considerados.
- **Rastreabilidade:** Os Épicos permitem que as equipes rastreiem o progresso em direção a grandes objetivos ou funcionalidades ao longo do tempo, proporcionando uma maneira clara de ver como o trabalho diário contribui para os objetivos maiores do projeto.

Como os Épicos são usados:

1. **Planejamento e Estratégia:** No início de um projeto ou ao planejar novas funcionalidades, os Épicos servem para estruturar o pensamento em torno de objetivos amplos ou funcionalidades significativas. Eles são fundamentais nas discussões estratégicas sobre a direção e as prioridades do projeto.
2. **Desenvolvimento Incremental:** Conforme o projeto avança, os Épicos são detalhados em histórias de usuário menores e mais concretas. Estas são priorizadas e agendadas para sprints específicos, garantindo uma abordagem sistemática para cobrir todos os elementos do Épico.

3. **Avaliação do Progresso:** Épicos funcionam como marcadores para o acompanhamento do avanço em direção a objetivos estratégicos ou lançamentos significativos. O cumprimento das histórias de usuário vinculadas permite medir o desenvolvimento do Épico globalmente.
4. **Escopo Abrangente:** Na metodologia ágil, um Épico engloba um vasto conjunto de trabalho, similar a uma grande narrativa literária. Ele inclui múltiplas histórias de usuário e tarefas, estendendo-se possivelmente por vários sprints, e aborda diversas funcionalidades ou aspectos do produto para alcançar um objetivo extenso.
5. **Conexão e Significado:** Os Épicos na agilidade unem diversas histórias relacionadas, formando uma meta coletiva que reflete um significado profundo para o projeto, assim como os épicos literários entrelaçam personagens e eventos em uma trama unificada. Eles representam metas de grande importância, com impacto considerável no sucesso do projeto.
6. **Fragmentação Estratégica:** Analogamente à divisão de um épico literário em capítulos, um Épico em agilidade é decomposto em histórias de usuário. Essa estratégia permite que a equipe aborde partes do trabalho incrementalmente, mantendo a visão do objetivo maior que o Épico representa.

Esta reorganização fornece uma descrição coesa e focada da função e importância dos **Épicos** dentro das metodologias ágeis, esclarecendo seu papel no planejamento estratégico, desenvolvimento incremental e avaliação do progresso de projetos ágeis.

A escolha do termo '**Épico**' reflete a sua natureza abrangente e significativa, destacando a importância estratégica desses blocos de trabalho e sua capacidade de agrupar histórias de usuário menores em direção à realização de objetivos ou funcionalidades significativas do produto.

Portanto, os **Épicos** emergem como ferramentas vitais para planejar, organizar e monitorar o progresso em direção a metas de larga escala, facilitando uma gestão de projeto que equilibra visão geral com execução detalhada.

A terminologia '**Épico**' é inspirada na literatura, onde representa narrativas de grande escala, e é apropriadamente empregada nas metodologias ágeis para descrever trabalhos que são essenciais para o sucesso e evolução do projeto.

As user stories anteriormente mencionadas foram construídas seguindo um padrão que foca em expressar claramente as necessidades e os desejos dos usuários finais, facilitando a compreensão da equipe sobre o que precisa ser desenvolvido e por quê. A construção dessas user stories se baseia em conceitos fundamentais para garantir que sejam eficazes e valiosas para o processo de desenvolvimento. Vamos explorar como essas user stories foram construídas, com base em definições e conceitos-chave:

ESTRUTURA DE UMA USER STORY

A estrutura típica de uma user story segue o formato:

"Como [tipo de usuário], eu quero [ação ou recurso] para que [benefício ou valor]."

Essa estrutura ajuda a focar no usuário, na funcionalidade e no valor que a funcionalidade traz, garantindo que o desenvolvimento esteja alinhado com as necessidades reais dos usuários.

Conceitos Utilizados na Construção das User Stories

Persona do Usuário: Cada user story começa com a identificação de uma persona específica ("**Como um cliente regular**", "**Como um novo cliente**", "**Como um usuário online**"). Isso ajuda a equipe a entender para quem eles estão desenvolvendo e a personalizar a solução para atender às necessidades desse usuário específico.

Necessidade ou Desejo: A parte central da user story ("**eu quero ver um logotipo modernizado**", "**eu quero navegar facilmente pelo novo site**") descreve o que o usuário precisa ou deseja. Isso direciona o foco da equipe para as funcionalidades e características específicas que devem ser implementadas.

Justificativa ou Explicação: O final da user story ("**para que eu possa sentir que a marca está evoluindo**", "**para que eu possa encontrar informações sobre serviços sem esforço**") explica o motivo pelo qual o usuário tem essa necessidade ou desejo. Este é o valor ou benefício que o usuário obtém, orientando a equipe a entender o impacto do que será desenvolvido.

Importância dos Critérios de Aceitação (*explicados logo abaixo*): - Os critérios de aceitação são um componente crucial. Eles definem as condições específicas que uma user story deve satisfazer para ser considerada concluída, garantindo que todos tenham a mesma compreensão do que é esperado;

Construção Focada no Valor: - As user stories foram construídas com um foco claro no valor para o usuário, garantindo que o desenvolvimento seja guiado pelo que é mais importante para os usuários finais. Isso ajuda a priorizar o trabalho e assegura que a equipe esteja sempre trabalhando em funcionalidades que ofereçam o maior valor;

Flexibilidade e Colaboração: - A construção dessas user stories também incentiva a flexibilidade e colaboração, permitindo ajustes baseados no feedback dos usuários e na evolução do projeto. Ao manter o foco nas necessidades dos usuários, a equipe pode adaptar-se para melhor atender essas necessidades à medida que mais informações se tornam disponíveis; e

Esses conceitos e a estrutura utilizada nas user stories ajudam a garantir que o projeto seja desenvolvido de forma a maximizar o valor entregue aos usuários, mantendo a equipe alinhada e focada nas metas do projeto.

DEFINIÇÃO DE PRONTO (DoD) E CRITÉRIO DE ACEITE (AC)

Entender a diferença entre "definição de pronto" (Definition of Done - DoD) e "Critério de aceite" em histórias de usuário no contexto ágil **é crucial**.

- **Definição de Pronto (DoD):** Refere-se aos critérios específicos que uma história de usuário ou tarefa deve atender para ser considerada completa. Estes critérios são estabelecidos pela equipe e geralmente incluem aspectos como qualidade, funcionalidade, testes e documentação. A DoD assegura que todos na equipe

tenham uma compreensão clara do que significa concluir uma tarefa, garantindo consistência e qualidade no trabalho entregue.

Entendido. Ajustando a abordagem para refletir um entendimento mais específico de como o conceito de Pronto (DoD) se aplica dentro do contexto de equipes ágeis e a sua aplicabilidade em qualquer tipo de projeto, é crucial focar na verificação interna pela equipe antes da validação pelo Proprietário do Produto (Product Owner - PO). O DoD deve ser um conjunto de critérios objetivos e mensuráveis que garantem que a User Story (História de Usuário) ou tarefa atende a todos os requisitos e funcionalidades esperados, sem subjetividade, permitindo uma validação interna efetiva antes da apresentação ao PO. Aqui está uma proposta de critérios de Pronto adaptável a qualquer projeto:

- **Compreensão Clara:** A equipe tem uma compreensão completa e unânime da User Story ou tarefa, incluindo seus objetivos, requisitos e impacto.
 - **Requisitos Atendidos:** Todos os requisitos e funcionalidade da User Story ou tarefa foram atendidos. Esses critérios são acordados antecipadamente pela equipe e são mensuráveis.
 - **Qualidade Assegurada:** O trabalho passou por todas as verificações de qualidade necessárias, incluindo, mas não se limitando a, testes (conforme aplicável ao projeto), revisões de desempenho, e verificações de conformidade com padrões relevantes.
 - **User Story Pronta:** A User Story ou tarefa está em um estado que não requer trabalho adicional para ser testada/validada para entrega a equipe de testes.
 - **Adaptação aos Projetos:** Este conjunto de critérios deve ser adaptado pela equipe para cada projeto específico, garantindo relevância e aplicabilidade aos diferentes contextos e necessidades do projeto.
 - **Revisão e Atualização:** O DoD deve ser revisado regularmente pela equipe para garantir que permaneça alinhado com os objetivos do projeto e as necessidades da equipe e partes interessadas.
 - **Comunicação com o PO:** Embora o DoD seja um critério interno de validação pela equipe, é importante manter uma comunicação clara e contínua com o Proprietário do Produto para assegurar que as expectativas estejam alinhadas.
- **Critério de Aceite:** Está mais relacionado à aprovação do cliente ou do proprietário do produto. Uma história de usuário é "aceita" quando o cliente ou proprietário do produto confirma que a tarefa atende às expectativas e as especificações definidas. O aceite é geralmente feito após a conclusão da tarefa, após o de acordo com a DoD, e é uma verificação de que o trabalho realizado atende às necessidades do Product Owner (PO) ou cliente.
 - **Definição Clara e Específica:** Os Critérios de Aceite são definidos de maneira clara e específica pelo Proprietário do Produto, com a colaboração da equipe, antes do início do trabalho na User Story. Eles devem descrever funcionalidades específicas, comportamentos e condições que o produto ou serviço deve cumprir.
 - **Mensuráveis e Testáveis:** Cada critério deve ser mensurável e testável, permitindo uma avaliação objetiva de se a User Story atende ou não às expectativas. Isso significa que cada AC deve ser formulado de forma a permitir uma verificação clara e inequívoca.

- **Alinhamento com Objetivos do Projeto:** Os Critérios de Aceite devem estar alinhados com os objetivos gerais do projeto e as necessidades do cliente, assegurando que a entrega contribua efetivamente para a visão geral do produto ou serviço.
- **Compreensível por Todos:** Embora criados pelo PO, os Critérios de Aceite devem ser compreensíveis por toda a equipe e outras partes interessadas relevantes, garantindo que todos estejam na mesma página quanto ao que se espera do resultado final.
- **Base para Testes:** Servem como a base para o desenvolvimento de casos de teste, garantindo que todos os testes realizados estejam alinhados com o que é importante para o PO e, por consequência, para o cliente.
- **Flexibilidade para Ajustes:** Embora definidos no início, os Critérios de Aceite podem ser ajustados com o consentimento do PO, refletindo aprendizados e feedbacks durante o ciclo de desenvolvimento, desde que mantendo o alinhamento com os objetivos do projeto.
- **Colaboração na Definição:** O PO lidera a definição dos AC, mas deve buscar input e colaboração da equipe para garantir que os critérios sejam realistas, compreensíveis e alinhados com a capacidade técnica e os prazos do projeto.
- **Revisão e Confirmação:** Antes de uma User Story ser submetida ao processo de AC, é obrigatório que o DoD tenha sido confirmado, revisado e aprovado pela equipe do projeto.
- **Abordagem clara:** adotando essa abordagem objetiva e clara para os Critérios de Aceite, as equipes podem melhorar significativamente a precisão na entrega de User Stories, aumentando a satisfação do cliente e a eficiência do processo de desenvolvimento.
 - *(similar) Durante o encerramento do projeto, o gerente do projeto revisa a documentação da fase anterior, documentação de aceitação pelo cliente com base no processo Validar Escopo (Seção 5.5), by PMI*

SPRINT ZERO

A Sprint Zero, embora não oficialmente reconhecida no Scrum Guide, é uma prática adaptada por muitas equipes Scrum para estabelecer uma base sólida antes de começarem as sprints de desenvolvimento iterativo e incremental. Ela serve para preparar o terreno para o projeto, garantindo que todos os elementos necessários estejam em lugar para facilitar um processo de desenvolvimento suave. Aqui está uma visão mais detalhada do que normalmente ocorre durante a Sprint Zero, bem como o que deve e não deve ser feito.

O que ocorre na Sprint Zero:

Definição da Equipe: Assegurar que todos os papéis necessários estejam preenchidos, incluindo a contratação ou alocação de recursos adicionais necessários para o projeto.

Estabelecimento de Infraestrutura e Ferramentas: Configurar servidores, ambientes de desenvolvimento, ferramentas de CI/CD (Integração Contínua/Entrega Contínua), repositórios de código e outras plataformas de colaboração essenciais para o projeto.

Definição do Product Backlog Inicial: Embora detalhado e priorizado ao longo do projeto, um backlog inicial deve ser criado para guiar as primeiras sprints.

Planejamento de Releases e Sprints: Esboçar um plano de release preliminar e uma visão geral das sprints, incluindo marcos importantes e entregas esperadas.

Arquitetura e Design Inicial: Estabelecer uma visão geral da arquitetura do sistema e do design, focando em soluções escaláveis e sustentáveis que atendam aos requisitos do projeto.

Preparação para o Desenvolvimento: Garantir que a equipe tenha acesso a todos os recursos necessários, incluindo documentação, acessos e treinamentos específicos sobre as ferramentas e processos a serem utilizados.

TAMANHO DA SPRINT ZERO:

O tamanho de uma Sprint Zero pode variar dependendo da complexidade do projeto e da equipe, mas geralmente dura entre **uma a duas semanas**. O objetivo é dedicar tempo suficiente para estabelecer uma fundação sólida sem atrasar o início do desenvolvimento efetivo.

O que deve ser feito:

- Estabelecer objetivos claros e expectativas para a Sprint Zero.
- Preparar a equipe e o ambiente de desenvolvimento.
- Criar um backlog inicial que guiará o desenvolvimento futuro.
- Definir processos e ferramentas de trabalho.

O QUE NÃO DEVE SER FEITO:

- ↘ Adiar decisões críticas de design ou arquitetura que impactarão o desenvolvimento futuro.
- ↘ Tentar detalhar completamente o product backlog ou planejar todo o projeto até o menor detalhe.
- ↘ Iniciar o desenvolvimento de funcionalidades sem ter a infraestrutura ou diretrizes básicas estabelecidas.
- ↘ Considerar a Sprint Zero como uma extensão do desenvolvimento, onde o trabalho em funcionalidades específicas começa.

A Sprint Zero é, portanto, um período crítico para a preparação e planejamento, estabelecendo uma base sólida para as sprints subsequentes e para o sucesso geral do projeto.

TÉCNICA MoSCoW (TÉCNICA DE PRIORIZAÇÃO)

MoSCoW, é uma ferramenta de priorização usada em gestão de projetos, especialmente útil em metodologias ágeis. MoSCoW é um acrônimo que representa quatro categorias de prioridade:

- **M - Must Have (Deve Ter):** Esses itens são essenciais para o sucesso do projeto. Sem eles, o projeto não pode ser considerado completo ou bem-sucedido. Eles são não negociáveis.
- **S - Should Have (Deveria Ter):** Itens importantes que devem ser incluídos no projeto se possível, mas cuja ausência não inviabiliza o projeto. Eles são valiosos, mas não críticos.
- **C - Could Have (Poderia Ter):** Esses são itens desejáveis que poderiam melhorar a entrega do projeto de maneira significativa, mas podem ser facilmente dispensados sem um impacto substancial no resultado final. São as "boas de ter" se houver tempo e recursos.
- **W - Won't Have (Não Terá Neste Momento):** Itens que foram identificados como os menos críticos, menos importantes ou menos viáveis neste momento do projeto. Isso não significa que eles nunca serão feitos; apenas não serão realizados na entrega atual.

A técnica MoSCoW ajuda as equipes a entender quais recursos, funcionalidades ou tarefas são cruciais para o sucesso do projeto, permitindo uma alocação de recursos mais eficiente e focada. Isso também facilita a comunicação clara das prioridades para todas as partes interessadas do projeto.

DESAFIOS DE MEDIÇÃO EM PROJETOS HÍBRIDOS OU ÁGEIS

Para ilustrar esses parâmetros e problemas dentro de um contexto específico, podemos considerar um projeto de revisão de website para um Doggy Daycare que está sob nova direção. A equipe está utilizando uma abordagem ágil para desenvolver um novo logotipo, melhorar a funcionalidade do site e preparar materiais para um evento importante. Eles estão mais de dois meses no projeto e enfrentam problemas técnicos como fotos que não carregam e falhas na página de pagamento.

- **Interpretação da Duração do Sprint:** A duração do sprint em metodologias ágeis pode variar conforme a interpretação. Alguns consideram um sprint de duas semanas como 10 dias úteis, excluindo finais de semana, enquanto outros podem incluir todos os 14 dias corridos. Essa divergência afeta diretamente a construção de gráficos de Burndown e Burnup, que são instrumentos visuais para rastrear o progresso do trabalho.
- **Inclusão de Feriados e Finais de Semana:** A contabilização de feriados e finais de semana nos sprints é outro ponto de variação. Alguns métodos consideram esses dias como 'custos operacionais' e os incluem nos gráficos, o que pode alterar a percepção do progresso e da velocidade da equipe.

- **Ferramentas de Gerenciamento de Projetos:** Plataformas como Asana, Trello e Jira oferecem soluções automatizadas para capturar dados e gerar gráficos de Burndown e Burnup com maior precisão e eficiência. Elas permitem que equipes se concentrem mais na interação e colaboração, essenciais em abordagens ágeis, ao invés de se prenderem a processos estruturados e manutenção manual de gráficos.

Pontos para Contextualização:

- **Flexibilidade vs. Rigidez:** Em ambientes ágeis, a flexibilidade é favorecida em detrimento de processos rígidos. Isso permite adaptações às variações de duração de sprints devido a feriados e outras interrupções.
- **Visualização de Progresso:** O desafio é criar prompts eficazes que refletem o progresso da equipe de forma precisa, mesmo com essas variações.
- **Automatização de Dados:** A automatização através de ferramentas especializadas alivia a carga de criar e atualizar gráficos de progresso manualmente, permitindo mais tempo para trabalho colaborativo.

Conclusão:

A eficácia na gestão de projetos ágeis depende não apenas da escolha das ferramentas certas, mas também de uma compreensão clara e comum dos processos e parâmetros de trabalho da equipe. Ao priorizar a interação entre pessoas e flexibilidade, as equipes podem navegar pelos desafios de maneira eficiente e manter o foco em entregar valor.