

Fundamentos de ALM

CICLO DE VIDA DE SOFTWARE
CARLOS CASALICCHIO



Conteúdo

Sinopse.....	5
Público Alvo.....	5
Material didático.....	5
Tópicos abordados	6
Introdução	6
O que é ALM?.....	6
Etapas do ALM.....	6
Introdução.....	7
Ciclo de vida de Desenvolvimento de Software.....	8
O Processo ALM.....	11
Estratégias para reduzir o tempo do ciclo ALM	14
Identificar métricas que realçam engarrafamentos.....	15
Tamanho das Filas	16
Tamanho dos Batches.....	16
Velocidade.....	16
Custos de Retrabalho	17
Tendências de Bug	17
Demora no Feedback	17
Criar soluções em potencial.....	18
Entender WIP e Ciclos.....	18
Identificar passos dos processos que introduzem defeitos	22
Entender o processo de qualidade do começo ao fim.....	24
Utilizar esforços de qualidade logo no início do ciclo	26

Reduzir o Desperdício	27
Identificando Desperdício.....	27
Lidando com Desperdício.....	27
Automatizando Tarefas Repetitivas	28
Criar um plano de Melhorias	28
Desenvolver o valor com Qualidade.....	29
Implementando Scrum.....	29
Funções Scrum	29
Equipe Scrum	30
O Scrum Master	30
Scrum Product Owner.....	31
Artefatos Scrum.....	32
Backlog do Produto	32
User Stories (Estórias de Usuário).....	32
Sprint Backlog	33
Técnicas de Estimativa.....	35
Estimativas de Esforço.....	35
Planning Poker® / Poker Scrum.....	36
Gráfico Burndown	38
Gráfico simples de Burndown	38
Definição de Pronto	39
Planejamento	40
Sprint	40
O Processo Sprint	41
Planejando um lançamento.....	45

Etapas ALM	47
Gerenciamento de projetos.....	47
Determinando o esforço.....	48
Metodologia incremental e iterativa.....	48
Requisitos	49
Arquitetura.....	50
Codificação	51
Testes e Qualidade.....	52
Gestão de implantação	52
Operações	53
Integrar Desenvolvimento e Operações.....	53
Atividades.....	55
Atividade 1 [20 minutos].....	55
Atividade 2 [25 minutos].....	55
Atividade 3 [15 minutos].....	58
Atividade 4 [15 minutos].....	58
Atividade 5 [15 minutos].....	59
Atividade 6 [15 minutos].....	59
Atividade 7 [15 minutos].....	60
Avaliação.....	60
Para as questões 1 a 4.....	60
Questão 1.....	60
Questão 2.....	61
Questão 3.....	62
Questão 4.....	62
Questão 5.....	62

Questão 6.....	63
Questão 7.....	63
Questão 8.....	63
Questão 9.....	64
Questão 10.....	64
Referencias.....	64

Sinopse

Este treinamento oferece aos alunos uma completa visão sobre o gerenciamento do ciclo de vida das aplicações (ALM – Application Lifecycle Management), englobando os aspectos de sua concepção, construção, evolução, declínio e desativação. Apresenta aos participantes a importância dos atores envolvidos na criação e manutenção de uma aplicação: Pessoas, Processos e Tecnologias.

Público Alvo

Este curso deve ser aplicado a todos os profissionais que trabalham ou trabalharão em projetos que adotam o processo de ALM. Desde o analista de negócios até o profissional desenvolvedor, passando pelos PO's e gerentes de produtos e projetos.

Material didático

A apostila de ALM será utilizado como referência e as aulas serão ministradas utilizando apresentações. Os alunos também receberão uma cópia digital das apresentações para realizarem as anotações necessárias.

Tópicos abordados

Introdução

Software é complexo; Software é abstrato; Requisitos são incompletos; Tecnologia muda rapidamente; as melhores práticas ainda não são maduras; A tecnologia é um domínio vasto; A tecnologia está sempre incompleta; Desenvolvimento de software é pesquisa; O trabalho repetitivo é automatizado; A construção é o projeto; mudar é considerado fácil; Mudanças são inevitáveis.

O que é ALM?

O processo de ALM (desenvolvimento); Governança do ciclo de vida da aplicação; Componentes importantes para a governança.

Etapas do ALM

Gerenciamento de projetos; estabelecendo um processo de desenvolvimento; Metodologia incremental e iterativa; Requisitos; Arquitetura; Codificação; Testes e Qualidade; Gestão de implantação; Operações.

Introdução

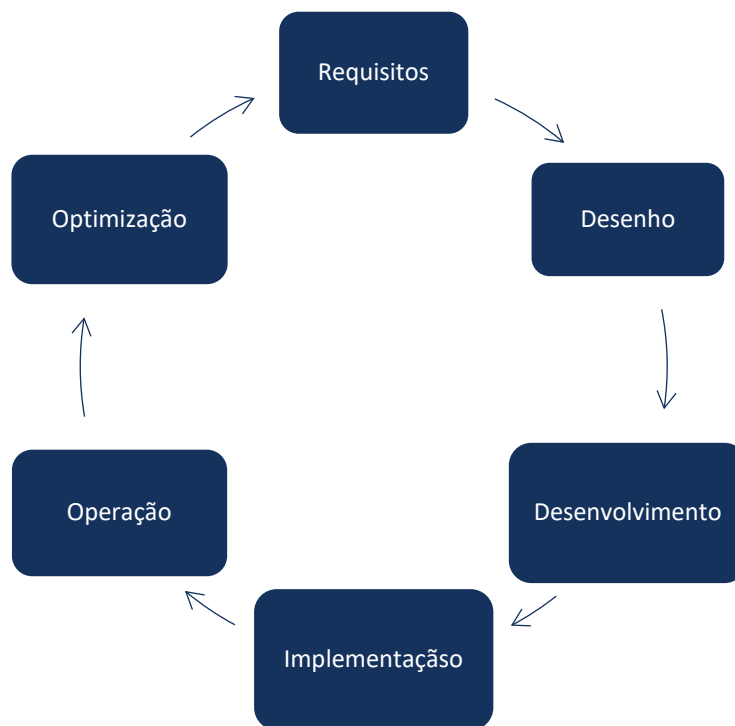
Qual o propósito do Gerenciamento do Ciclo de Vida da Aplicação (Application Lifecycle Management)?

Neste curso serão apresentados o valor de uma visão completa das práticas de ciclo de vida. Discutiremos os benefícios de feedback rápido.

Aprenderemos como implementar estratégias para reduzir o tempo nos ciclos da ALM; a implementar estratégias para melhorar a qualidade do software; implementar estratégias para reduzir o desperdício e iremos criar um plano de melhoria de processos.

A habilidade de produzir com qualidade, no tempo esperado e com preço competitivo é importante para qualquer organização. A globalização força as empresas a operarem em um ambiente de desenvolvimento distribuído. Hoje em dia, empresas estão buscando meios mais eficientes e sistemáticos de superar seus desafios. Uma resposta a esses desafios foi o crescimento dos métodos Ágeis, tal como XP (Extreme Programming), Scrum, etc. Originalmente esses métodos focavam em equipes de desenvolvimento locais. Recentemente, o uso de métodos Ágeis em ambientes distribuídos de desenvolvimento têm sido pesquisados ativamente. Na literatura e entre as ferramentas, o termo Gerenciamento do Ciclo de Vida do Produto tem sido discutido abertamente. Uma solução de Ciclo de Vida do Produto pode compreender vários sistemas que são usados para criar e gerenciar os dados relacionados do produto, tais como requisitos, configuração, recursos da Empresa (ERP), sistemas de auxílio de engenharia, etc. Interfaces e aplicações de integração podem ser necessárias para habilitar todos esses sistemas a trabalharem juntos. Estima-se que no futuro, o Gerenciamento do Ciclo de Vida do Produto proporcione melhor integração de produtos multidisciplinares, não somente produtos mecânicos ou elétricos. O conceito de Gerenciamento do Ciclo de Vida da Aplicação emergiu para indicar a coordenação de atividades e do gerenciamento de artefatos (requisitos, código fonte, casos de testes) durante o ciclo de vida do produto em software. Há a crença que soluções bem integradas de ALM tem

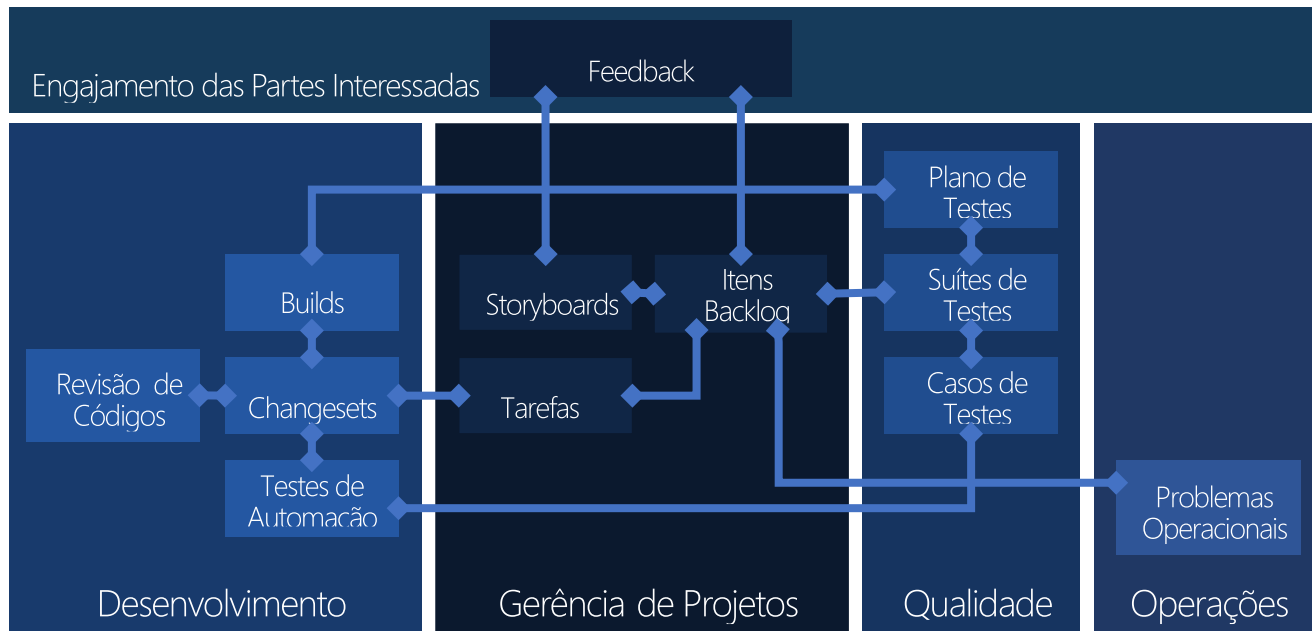
como alvo o desenvolvimento tradicional de produtos. No entanto, recentemente, o mercado de ferramentas ALM tem crescido significativamente. As raízes de soluções ALM estão no histórico de gerenciamento de configuração. Essas soluções são geralmente a fundação das estruturas ALM que proporcionam armazenamento, versionamento e rastreabilidade entre diferentes artefatos do ciclo de vida. No desenvolvimento de produtos multidisciplinares, ALM se encaixa em uma visão maior do Ciclo de Vida do Produto. Nesses produtos, o ALM se foca em gerenciar a porção do software de um produto multidisciplinar. [a]



Ciclo de vida de Desenvolvimento de Software

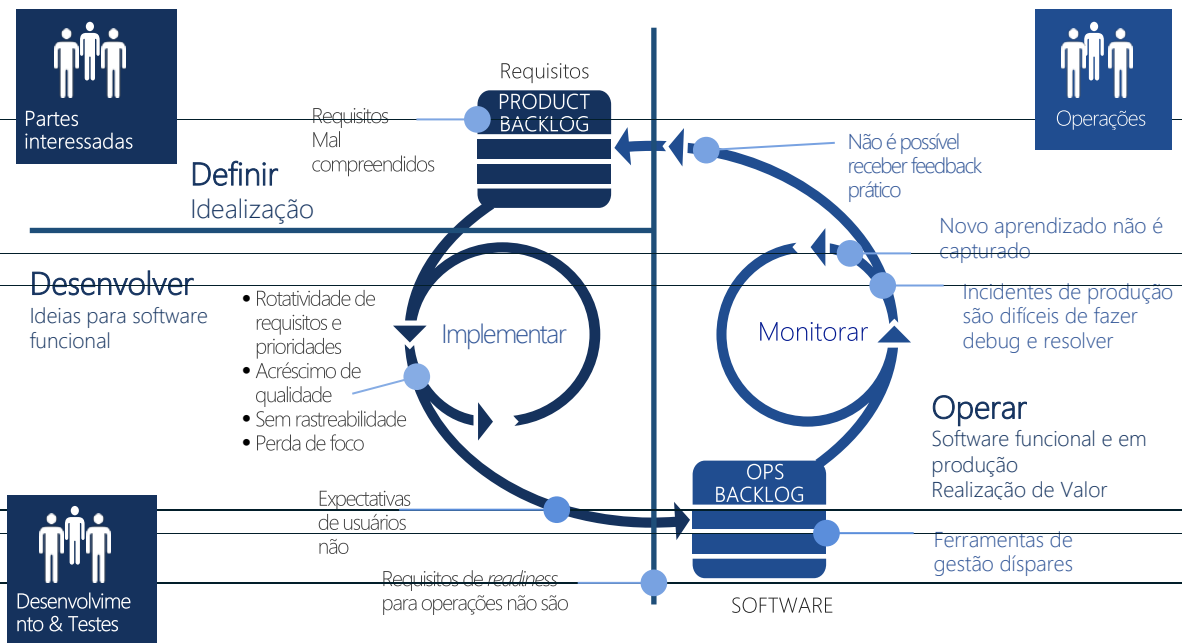
O propósito do gerenciamento do ciclo de vida é garantir que a aplicação sempre irá satisfazer as necessidades do negócio. ALM compreende todas as tomadas de decisão e gerenciamento da aplicação. Norteia e orienta como serão abordados todos os processos do ciclo de vida do

produto (neste caso, o software), desde o levantamento e priorização dos requisitos até a entrega do produto final.

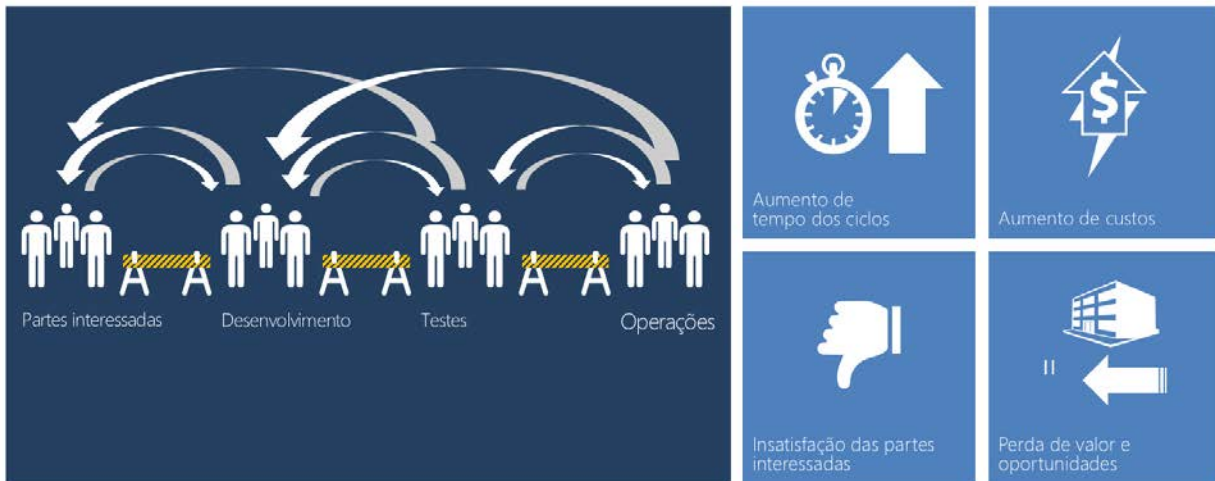


Para que se alcance sucesso em qualquer metodologia Ágil, é vital que os artefatos do projeto sejam facilmente rastreáveis e transparentes. Quando os artefatos são transparentes e rastreáveis, a fase de inspeção e adaptação possibilita encontrar os problemas e impedimentos que podem estar causando desperdícios ou atrasos. Por outro lado, essa transparência pode potencializar o que está funcionando corretamente, a fim de melhorarem as outras fases do fluxo.

Consequentemente, o fluxo deve ser bem desenhado e organizado, para que essa transparência e rastreabilidade possam ser mantidas em longo prazo. Dúvidas, confusões e incertezas podem desestabilizar o fluxo, provocando possíveis adulterações da metodologia e retrocedendo aos processos tradicionais de gerenciamento de projetos.



Equipes integradas de forma ineficiente resultam em impedimentos que impactam todo ciclo de vida do software desde a definição dos requisitos de desenvolvimento até a publicação em produção e operações. O possível impacto negativo resultante inclui aumentos no ciclo de tempo de entrega para agregar o valor de negócio esperado, e na velocidade necessária.

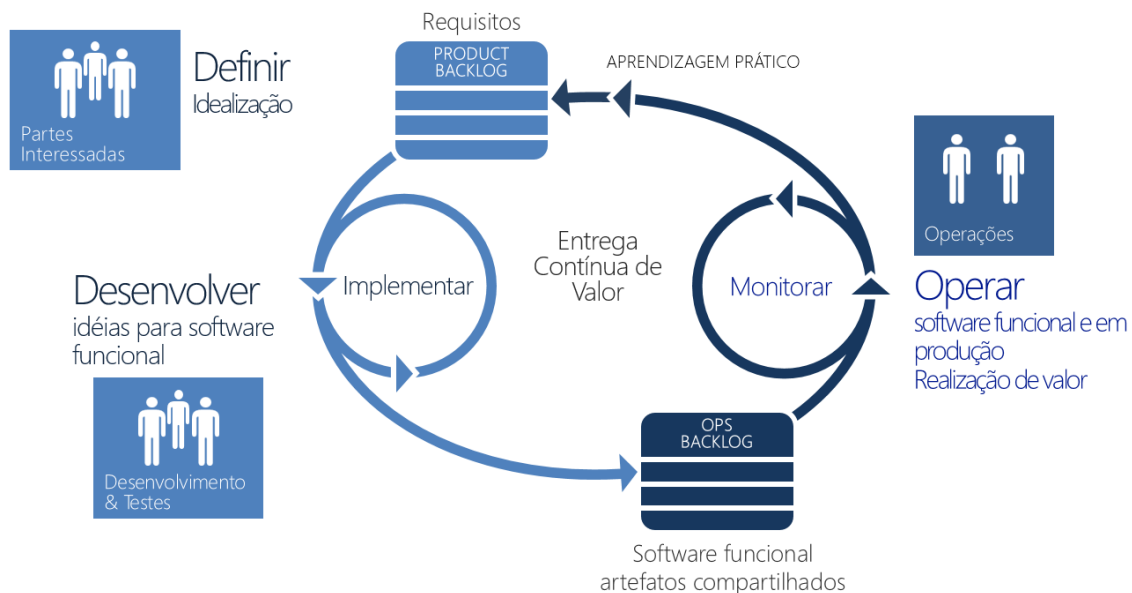


Desenvolvimento de software em empresas e organizações é uma tarefa multifuncional. Barreiras e lacunas entre equipes multifuncionais, que precisam estar integradas para entregar, estão entre os impedimentos principais que entram as entregas de valor. Tais impedimentos são geralmente uma manifestação de processos rígidos, ferramentas de colaboração ineficientes e práticas de desenvolvimento que não usam os avanços de tecnologia e oportunidades para integrar melhor as equipes.

Desde o momento de concepção do produto (software), até o momento em que o produto deixa de ser usado (aposentado), todas as fases constituem o ciclo de vida do software.

O Processo ALM

Equipes que trabalham sem barreiras entregam continuamente e agregam valor ao produto.



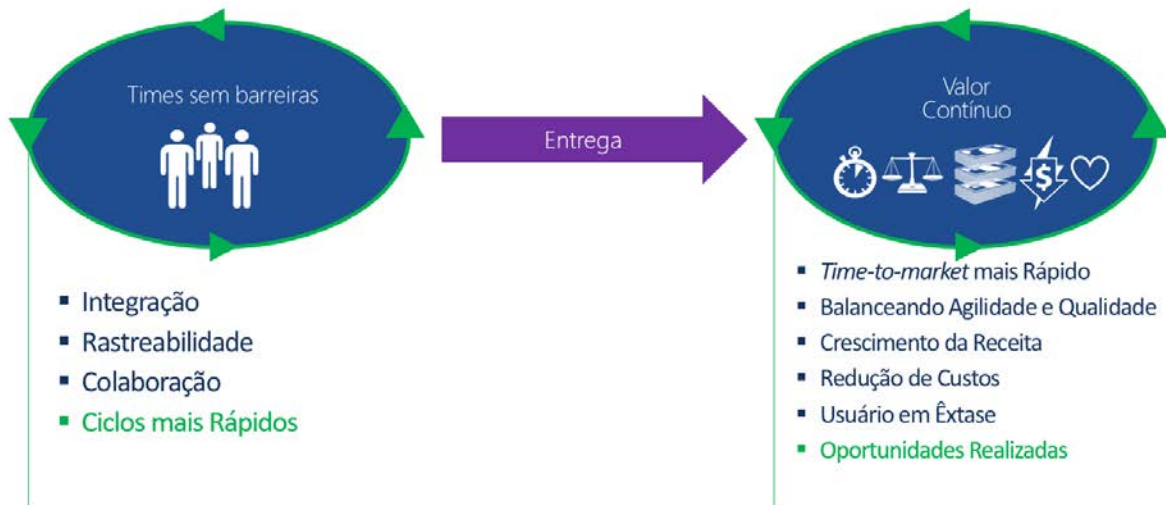
O primeiro passo em ALM é o desenvolvimento do caso de negócio. Essa análise ocorre antes do processo de desenvolvimento começar. Uma vez que o *business case* é aprovado, o desenvolvimento da aplicação se inicia, e a governança é nesse momento implementada, através do gerenciamento do *portfólio* do projeto. Em algumas organizações, isso é simples: um gerente de projeto pode estar diretamente ligado ao time de desenvolvimento, ou um dos técnicos da equipe poderá assumir esse papel. Em outras organizações, pode ser utilizada uma abordagem mais formal, contando com o auxílio de um escritório de gerenciamento de projetos (PMO), para reforçar a prática dos procedimentos estabelecidos.



Uma vez que a aplicação completa é entregue e implantada, isso se torna parte do *portfólio* de aplicações da organização. Uma aplicação é um *ativo* como qualquer outro, de modo que a organização precisa constantemente avaliar seu custo e benefício. O gerenciamento do *portfólio* de aplicações fornece essa visão, oferecendo uma maneira de evitar funções duplicadas por meio de aplicações diferentes.

Até o momento não existe um processo consolidado de como implementar governança em ALM, ficando a critério de cada organização a escolha da melhor estratégia. ([b] p. 21)

Práticas de ALM bem aplicadas habilitam equipes a trabalharem eficazmente sem barreiras e entregando valor continuamente. A organização se beneficia de práticas eficazes de ALM reduzindo desperdícios, com ciclos mais rápidos de entregas, e oportunidades de negócio.



Estratégias para reduzir o tempo do ciclo ALM

Metodologias Ágeis incentivam a melhoria contínua. Para que essas melhorias aconteçam, é vital o hábito de inspeção e adaptação. Essa inspeção proporciona a oportunidade para todos os Stakeholders avaliarem como está seu trabalho, o que está dando certo e o que deve ser corrigido. Após a inspeção, deve-se criar um plano de ação para que tais melhorias possam ser implementadas de modo eficaz. A implementação de melhorias que reduzem o tempo dos ciclos de entrega exige atenção e esforço, tirando todos de sua zona de conforto.

Zona de conforto é uma série de ações, pensamentos e comportamentos que uma pessoa está acostumada a ter e que não a causam nenhum tipo de medo, ansiedade ou risco. É uma região onde nenhum indivíduo se sente ameaçado.

Na zona de conforto, as pessoas realizam sempre um determinado tipo de comportamento que lhe dá um desempenho constante, porém limitado e com uma sensação de segurança. Essa segurança é uma falsa segurança, uma vez que, quando ocorre uma grande mudança, quem está muito confortável leva um choque maior, e estará menos preparado para sobreviver do que os outros. Os indivíduos em geral, necessitam saber operar fora de sua zona de conforto para realizar avanços, melhorar seu desempenho seja ele no trabalho, na vida pessoal etc. A zona de conforto é um tema sempre muito debatido na psicologia.

[c]

Para ser capaz de identificar os pontos que podem ser melhorados, listamos a seguir alguns itens que são comuns e que afetam o desempenho de equipes Ágeis.

Identificar métricas que realçam engarrafamentos

Engarrafamentos ou gargalos são pontos do ciclo que encontram problemas comuns para prosseguir, por exemplo, na passagem de tarefas para outras fases (testes, QA, etc.).

Geralmente, os indicadores mais comuns de engarrafamentos estão nos tamanhos das filas e nos tamanhos dos batches.

Outra forma de apontar engarrafamentos é monitorar a velocidade de conclusão de atividades. Essa velocidade é o número de atividades completadas durante uma iteração ou Sprint.

Também é possível encontrar engarrafamentos calculando os custos de retrabalho, em instâncias onde os requisitos não estavam claros, ou estavam incompletos; e também buscando as tendências de defeitos ou bugs.

Tamanho das Filas

Filas se acumulam sempre que alguma atividade é passada para algum indivíduo ou equipe que já está muito atarefada e precisa deixar algum trabalho em espera. Isso acontece muito quando existem deficiência ou déficit técnico em atividades especializadas como testes, bancos de dados, análise de negócios ou especificação de requisitos.

Quando tarefas demais se acumulam na fila de qualquer indivíduo ou fase, imediatamente deve-se avaliar a razão para o acúmulo de atividades e, se possível, tomar medidas imediatas para mitigar qualquer atraso.

Tamanho dos Batches

Batches são o conjunto de itens do Backlog que serão entregues ou publicados ao final da Iteração ou Sprint. Se o número de itens a ser entregue é muito grande, a quantidade de itens que cada membro da Equipe deve completar é tamanha que a probabilidade de atraso aumenta consideravelmente. Conseqüentemente, filas acabam se acumulando à espera de cada membro da equipe.

Velocidade

A velocidade de uma iteração compreende o número (ou média) de itens completados durante cada Sprint. Se os gráficos de Burndown mostram que a velocidade está caindo em cada iteração, significa que a Equipe está encontrando dificuldades para completar mais atividades. Provavelmente existe algum tipo de engarrafamento, seja devido à dívida Técnica, ou mesmo por impedimentos que podem existir, mas que não foram identificados.

Custos de Retrabalho

Quando problemas são encontrados, em forma de defeitos (sejam bugs ou requisitos mal implementados), atividades precisam ser revisadas, reabertas e retrabalhadas. Esse processo de retrabalho não é contado na velocidade da iteração. E, a não ser que sejam novas funcionalidades de um requisito já existente – nesse caso, um novo item de Backlog é necessário – esse retrabalho causa atrasos e engarrafamentos (filas) para membros da equipe que tem que terminar o retrabalho antes de iniciar novas tarefas.

Tendências de Bug

Quando os relatórios indicam tendências a defeitos, isso significa possivelmente que alguém pode não estar fazendo seu trabalho corretamente. Esse trabalho pode não estar sendo bem feito por vários motivos, incluindo filas, déficit técnico, impedimentos, dificuldade na comunicação ou compreensão dos requisitos, etc. Sempre que defeitos são encontrados, as funcionalidades precisam ser refeitas (causando custos de retrabalho) e testadas novamente. Tudo isso causa engarrafamentos para as outras tarefas que já estavam agendadas.

Demora no Feedback

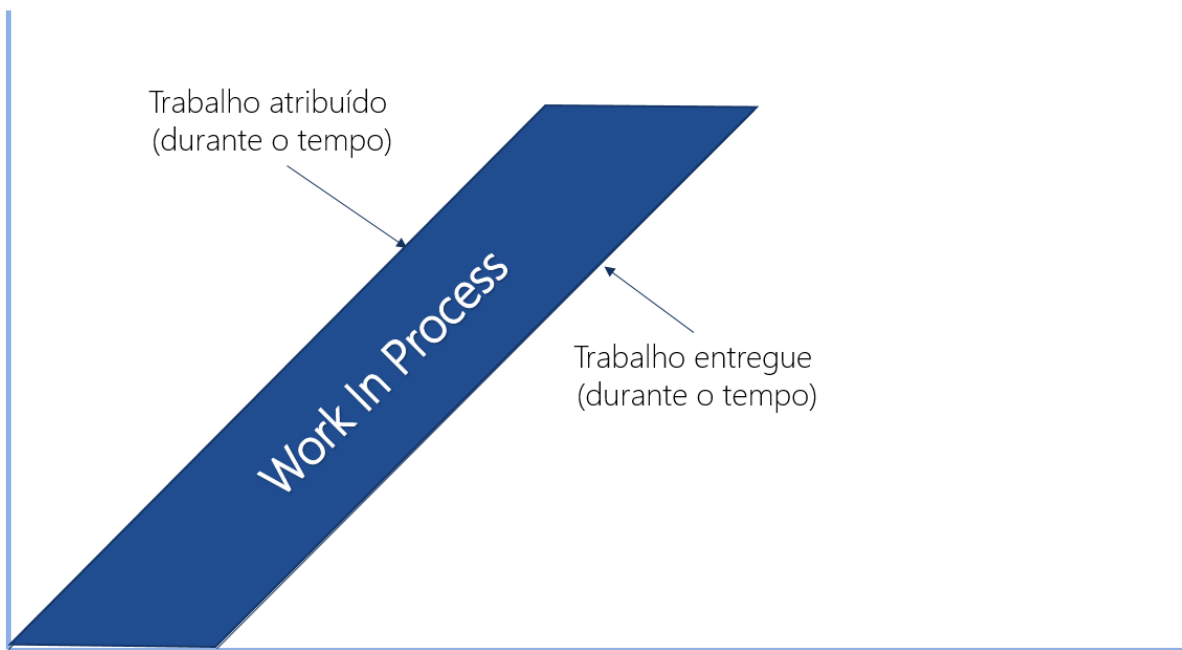
O ciclo de feedback deve ser rápido e dinâmico a fim de informar a equipe se o que está sendo feito está de acordo com as expectativas do Product Owner e do cliente. Se esse feedback demora, a equipe acaba decidindo por prosseguir com outras funcionalidades – o que causa o risco de construir sobre uma funcionalidade incorreta, sem feedback – ou param o trabalho dessa funcionalidade ou funcionalidades a fim de evitar retrabalho. Em ambos os casos, perde-se tempo essencial do desenvolvimento, causando desperdícios.

Criar soluções em potencial

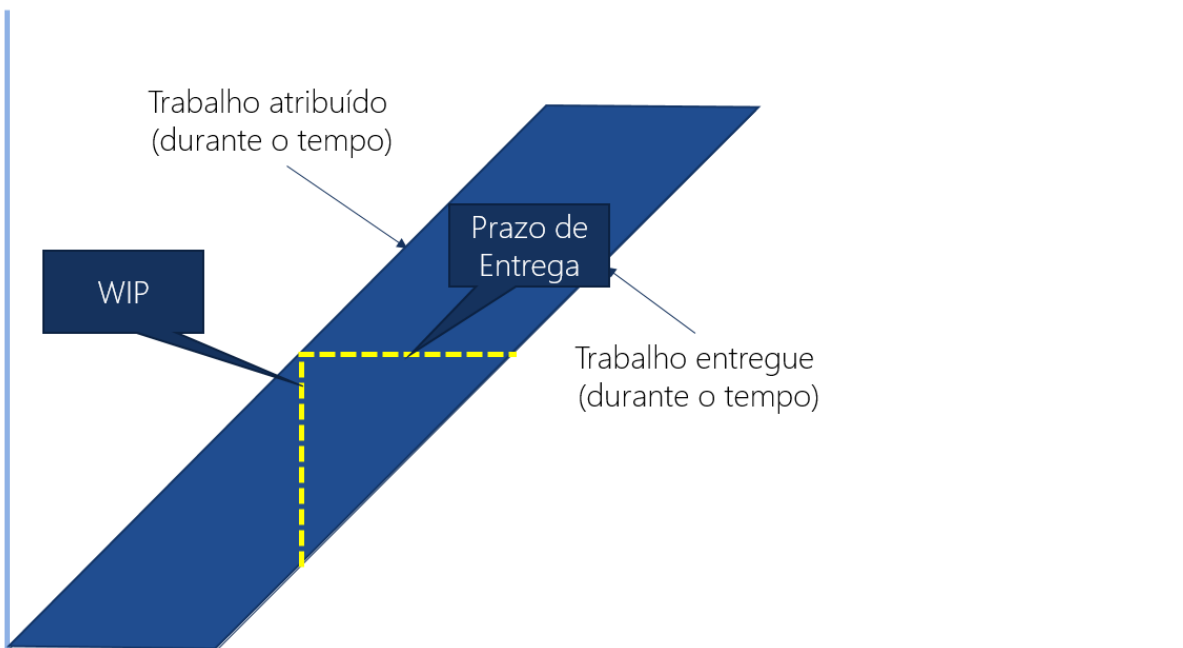
Após a inspeção, a fase de adaptação proporciona liberdade em investigar soluções que melhor se adaptem a organização de forma a realçar o processo ágil e tornar a equipe mais eficiente e de maior desempenho. Tais soluções podem não ser convencionais e podem ser criativas e originais, e devem ser consideradas como potencial de solução para a equipe. A solução pode incluir novas ferramentas de integração e fluxo, ou mesmo em coisas simples como sistemas mais transparentes de comunicação e trabalho. No entanto, cada equipe pode adaptar-se melhor a uma solução do que a outra. Portanto, deve-se focar-se na equipe como propósito da melhoria, otimizando a equipe para produtividade, e não para a simples utilização da solução.

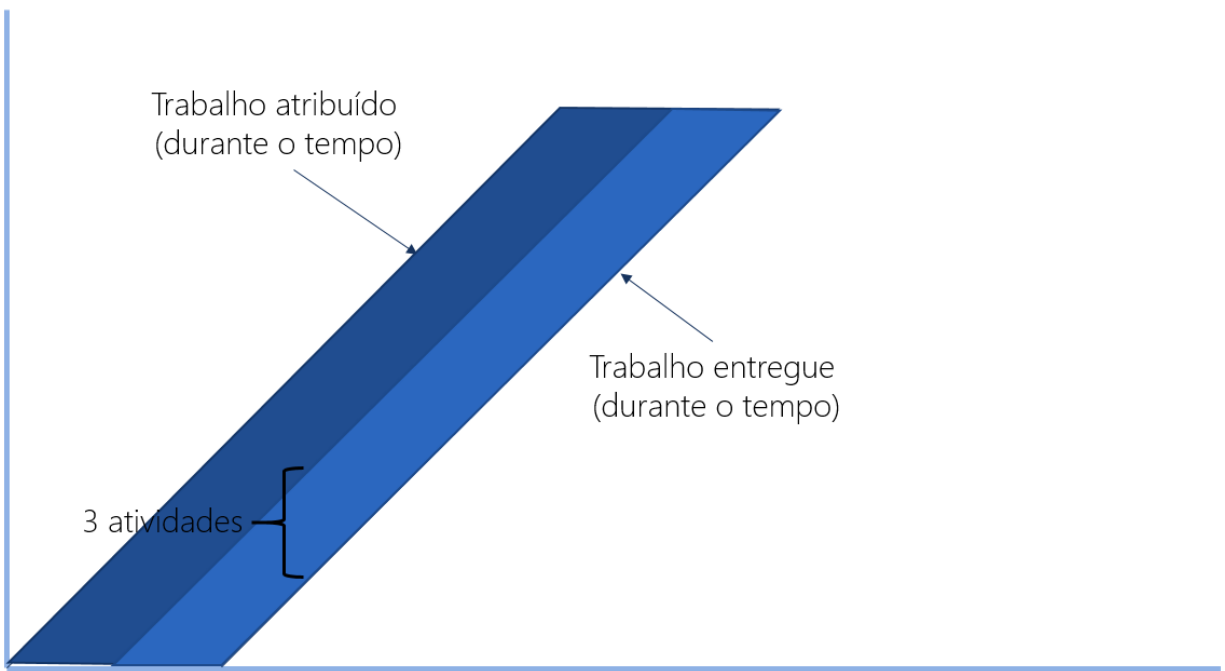
Entender WIP e Ciclos

WIP ou *Work in Progress* (trabalho em andamento) é a quantidade de itens sendo trabalhados em qualquer ponto do tempo durante o projeto. Para compreender melhor o conceito de WIP, vemos os diagramas a seguir.



Cada item a ser entregue demora um certo tempo para ser completado.

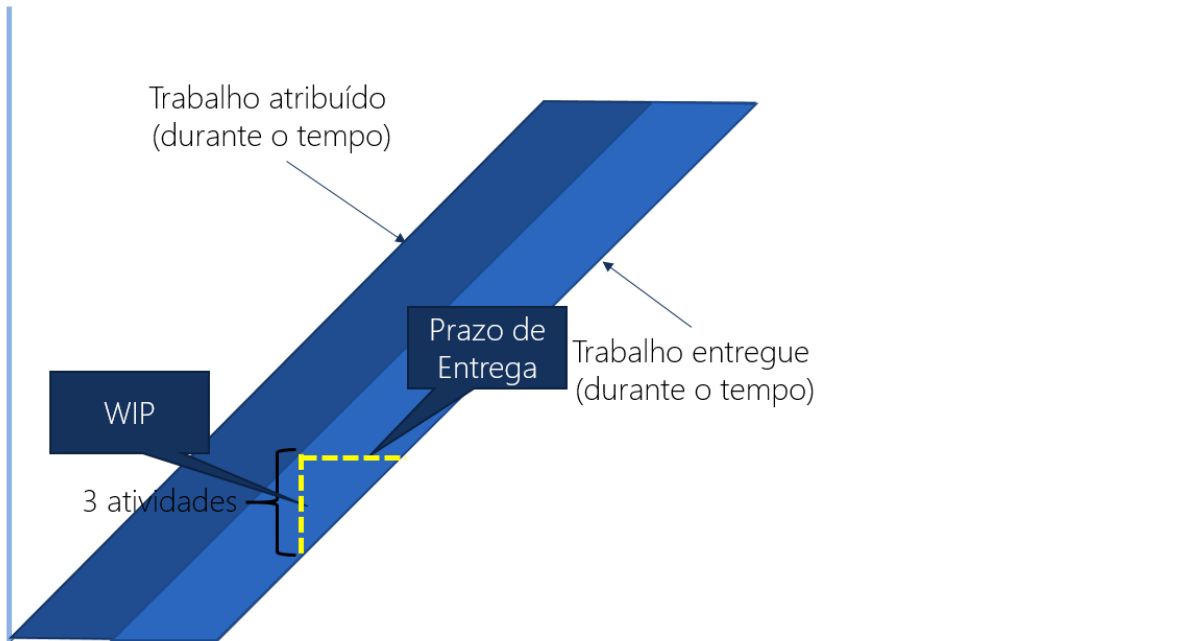




Se ajustarmos a quantidade de atividades a serem entregues em um batch, nosso WIP baixa para 3 por período, então teremos uma entrega em menos tempo. Claro, teremos menos itens entregues, mas o prazo de entrega também é reduzido pela metade.

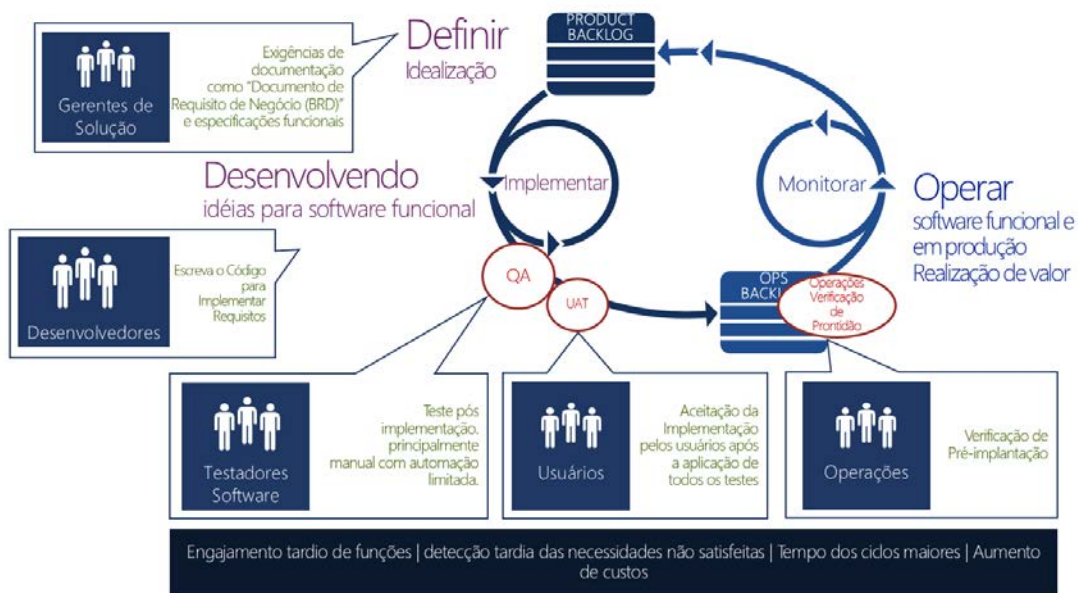
Portanto, atribuindo-se um WIP menor, o prazo de entrega automaticamente também é menor. Em longo prazo, menos itens a serem entregues por iteração auxilia na estimativa e velocidade da equipe, tornando-a mais eficiente. WIP menor também implica em maior satisfação do cliente, pois o mesmo vê entregas com prazo muito menor, e é capaz de prover feedback mais

significativo e atual, sem ter que esperar meses para poder ver melhorias reais no projeto.



Identificar passos dos processos que introduzem defeitos

A imagem a seguir descreve um fluxo generalizado e o envolvimento dos Stakeholders em cada fase. Note que, em sua organização, pode haver outros Stakeholders envolvidos e outros processos, de acordo com as necessidades do projeto.



Em uma metodologia ágil, a fase da definição ou idealização também é um ciclo. Ou seja, embora o cliente seja capaz de descrever o propósito do seu produto, o projeto sempre está definindo e idealizando os requisitos e funcionalidades que deverão ser entregues, ajustadas ou removidas. Essa definição se traduz em itens do Backlog do Produto, que são inseridos por prioridade de valor agregado e pela ordem que o Product Owner definir, de acordo com as necessidades do projeto. Cada item é, então, estimado pela equipe de desenvolvimento, que discute as tarefas necessárias para que cada funcionalidade seja implementada.

Durante essa fase de desenvolvimento, os membros da Equipe de uma equipe multifuncional trabalham juntos para concluir todos os itens que se comprometeram a completar. Esse batch de itens do Backlog do Produto é inserido no Backlog da Sprint. Cada Sprint pode durar entre 2 a quatro semanas, e inclui todos os requisitos estabelecidos pela definição de "pronto".

Somente após todos os itens serem integrados, testados e aprovados pelos usuários finais (ou testadores de qualidade) esse batch segue para o Backlog de Lançamento, ou Backlog de Operações.

Dentro desse Backlog de Operações, a equipe de infraestrutura aplica as mudanças do último batch de entrega em produção. Produção pode ser um produto para download, um site, ou um banco de dados.

Após o produto “entregável” ter sido confirmado como finalizado, uma nova iteração ou Sprint se inicia para implementar novas ou ajustar funcionalidades. E o ciclo se reinicia.

Após definir um processo que inclua todas as fases do desenvolvimento para cada iteração, a inspeção deve ocorrer durante cada iteração, sempre buscando pontos do processo que causem quaisquer tipos de atraso, ou que causem defeitos – sejam testes mal escritos, algoritmos pobres, etc.

Note que os problemas mais comuns estão listados a seguir:

- **Engajamento tardio de funções.** Stakeholders postergam suas tarefas até o último minuto.
- **Deteção tardia das necessidades não satisfeitas.** Feedback não é retornado.
- **Tempo dos ciclos maiores.** Pode ser causado por vários motivos.
- **Aumento de custos.** Mudanças ou impedimentos que provocam remanejamento no orçamento.

Entender o processo de qualidade do começo ao fim

Há várias maneiras de implementar código assim como há vários desenvolvedores que podem implementá-los e, por isso, desenvolvimento de software é uma profissão divertida. Também significa que há várias opiniões sobre o que é software de boa ou alta qualidade. O que um desenvolvedor considera código de qualidade pode ser bem diferente da opinião de outro desenvolvedor.

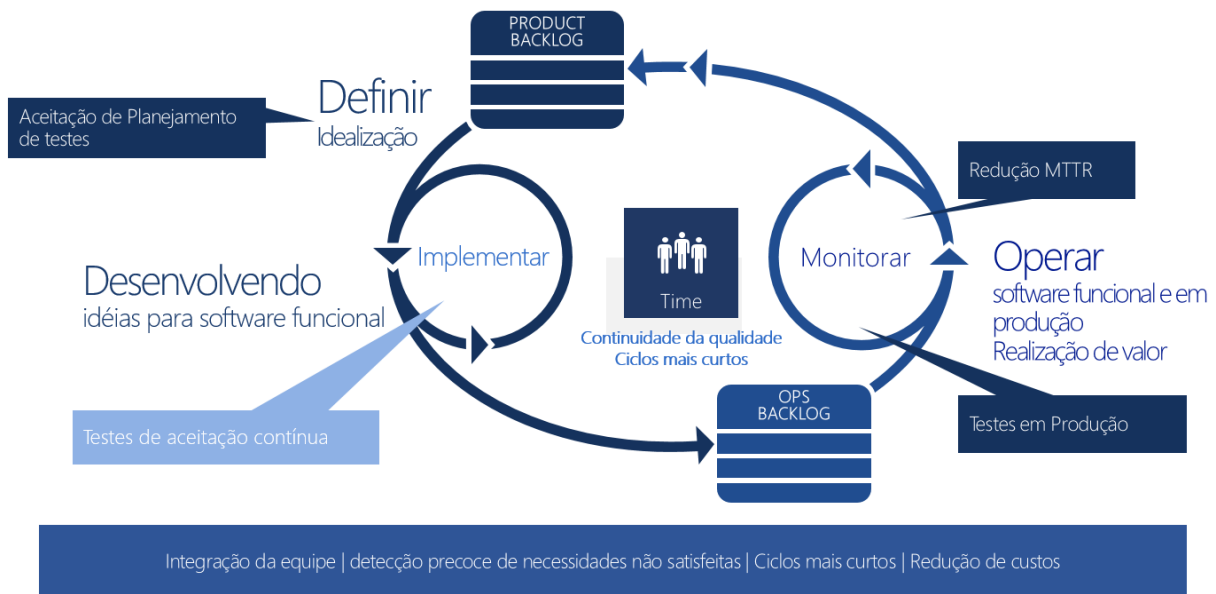
Um simples exemplo disso é um método escrito de forma compacta, usando sintaxe compacta e o menor número de linhas possível. Enquanto para um isso pode parecer código de qualidade, para outro desenvolvedor que prefere código prolixo e de estrutura de leitura mais fácil, pode parecer conciso demais e até esotérico.

Faz mais sentido começar a pensar em código em termos da qualidade que possui, ao invés de tentar expressar qualidade com uma única definição.

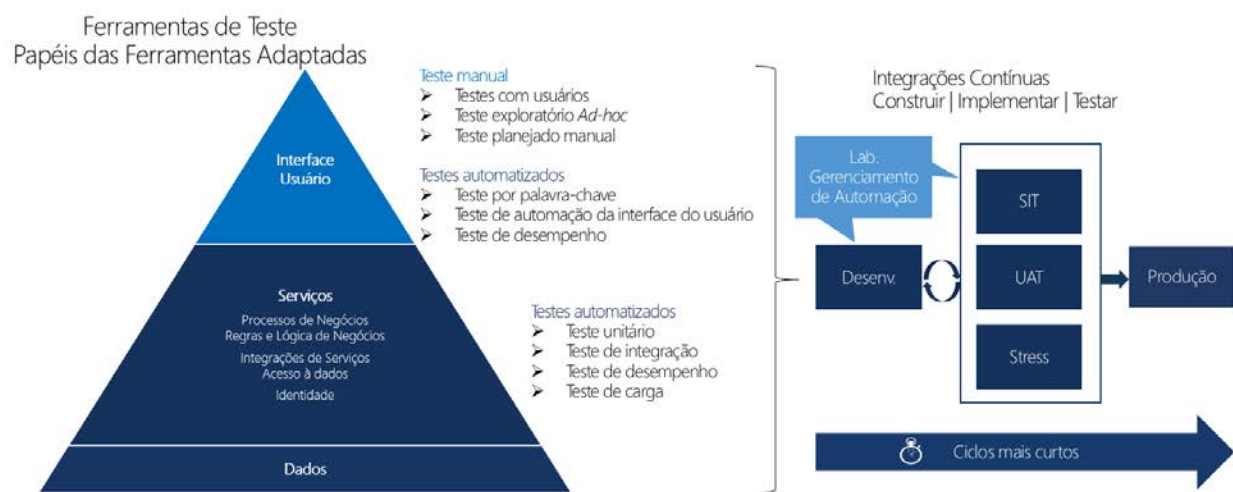
Indiferente de quais padrões de desenvolvimento que sejam usados para auxiliar no desenvolvimento e qualidade do código, os pontos mais importantes a serem estressados sobre software de qualidade são:

1. Qualidade consiste em funcionalidades do produto que suprem as necessidades dos clientes e, portanto, produzem satisfação ao cliente.
2. Qualidade consiste em estar livre de deficiências.

Outro ponto importante sobre qualidade é a capacidade do código em minimizar o tempo de leitura e a curva de aprendizado. Ou seja, quanto mais fácil o código for de ser lido e compreendido, esse código também consiste em software de qualidade. [d]



A QUALIDADE PODE SER APLICADA EM TODOS OS PONTOS DO FLUXO ALM.



TESTES VARIADOS E CONTÍNUOS AUMENTAM A QUALIDADE DAS ENTREGAS E DIMINUEM OS CICLOS

Um dos propósitos do ALM inclui a implementação de ferramentas de automação em quaisquer fases do ciclo que exijam esforço repetitivo. Entre elas, os melhores candidatos para automação estão em testes e lançamentos. Ambos possuem ferramentas disponíveis na solução Microsoft Team Foundation Server e Visual Studio.

Utilizar esforços de qualidade logo no início do ciclo

Considerando que os esforços de qualidade devem fazer parte constante do dia-a-dia da equipe, é aconselhável gastar mais tempo, no início de cada ciclo, para garantir a qualidade – seja por criar testes unitários e automatizados, ou gated check-ins, do que “correr atrás” dos defeitos e retrabalhos que são encontrados no final do ciclo.

Em outras palavras, mesmo que a curva de aprendizado e o tempo de entrega sejam maiores, aparentemente tornando a velocidade menor, em longo prazo os desperdícios com retrabalho serão menores, aumentando o desempenho da equipe e proporcionando um projeto mais fluido.

Reduzir o Desperdício

Identificando Desperdício

Qualquer situação que causa tempo perdido ou gastos desnecessários, Esses desperdícios podem se manifestar em:

- Inventário – trabalhos incompletos; trabalhos não publicados
- Superprodução – com funcionalidades raramente usadas
- Processamento – documentação desnecessária ou excessiva
- Transporte – entregas com mudanças de contexto
- Movimentação excessiva – membros da equipe se dividindo em vários projetos
- Espera – atrasos causados por filas
- Retrabalhos – defeitos, bugs e requisitos mal compreendidos

Lidando com Desperdício

Investigue e valide a causa do desperdício. Procure por atrasos ou engarrafamentos no desenvolvimento do software; valide os passos anteriores, a fim de identificar a causa do mesmo; ao confirmar sua causa, confirma-se o desperdício.

Elimine desperdícios de forma criativa, possivelmente consultando os membros da equipe e outros Stakeholders, ou mesmo, especialistas. Foco na melhoria contínua, sempre inspecionando e adaptando o ciclo e ferramentas, para alcançar a redução no ciclo de vida.

Preste atenção nas métricas que auxiliam validar e confirmar desperdícios:

- Aumento nos prazos de entrega dos ciclos
- Custos de retrabalho

- Tendências a defeitos e bugs

Automatizando Tarefas Repetitivas

A fim de minimizar ou extinguir desperdício, recomenda-se a automação de quaisquer atividades repetitivas que podem tomar tempo desnecessário. Entre essas tarefas, encontram-se:

- Testes unitários
- Build e merges
- Teste de Integração
- Testes exploratórios
- Publicação em QA e produção

Automatizando processos repetitivos, diminui-se o risco de erro – principalmente humano – da repetição. Também se diminui o tempo de execução dessas tarefas, que de outra forma teriam que ser feitas por um indivíduo. Automatizar também proporciona a coleta de dados metódica e detalhada para criação de relatórios.

Criar um plano de Melhorias

Para melhorar o desempenho da equipe e diminuir os ciclos de entrega é necessário criar um plano de melhoria de processos. Nesse plano, cria-se estratégias para incentivar mudanças na organização, identificando métricas chave que podem ser acompanhadas durante o processo de esforço de melhorias, a fim de criar um consenso favorável às mudanças.

Dessas métricas, recomenda-se:

- **O valor do negócio** – identifique os ganhos do negócio com a eliminação dos desperdícios

- **Custos de retrabalho** – aponte os custos e tempo gastos com atividades que poderiam ter sido evitadas
- **O tempo do ciclo** – com ciclos menores, os riscos são menores, os custos são menores.
- **Tendências de defeitos e bugs** – maiores vilões dos retrabalhos e maiores tempos de ciclo
- **Tamanho de Filas** – filas apontam para atrasos de atividades
- **Tamanhos de Batches** – quantidade de atividades entregadas ao final de cada Sprint pode mostrar que pouco está sendo feito, ou atividades demais estão sendo planejadas.

Desenvolver o valor com Qualidade

Toda funcionalidade concluída está relacionada a um certo valor que foi agregado ao produto, à organização, e ao cliente final. Por esse motivo, toda atenção com a qualidade garante que esse valor seja visível e palpável. Quando a funcionalidade exige retrabalhos, esse valor, por mais que seja grande, perde a qualidade ao provocar experiências mais pobres, resultando em impressões negativas do produto. Qualquer experiência negativa, por menor que seja, subtrai da expectativa e da confiança no projeto, podendo até, em casos extremos, levar ao fracasso total do mesmo.

Implementando Scrum

Funções Scrum

Dentro da Estrutura Scrum temos as seguintes funções:

- A equipe Scrum;

- O Scrum Master;
- O Product Owner Scrum (ou PO).

Cada uma dessas funções tem um número de responsabilidades definidas, interagem de perto com os outros integrantes e contribuem para o sucesso do projeto somente se cumprirem suas responsabilidades.

Equipe Scrum

Na estrutura Scrum todo o trabalho entregue ao cliente é feito por equipes Scrum. Uma equipe Scrum é um grupo de indivíduos trabalhando juntos para entregar os requisitos e funcionalidades em incrementos do produto.

Para trabalharem eficazmente é importante que cada membro da Equipe:

- Siga uma meta em comum;
- Siga as mesmas normas e regras;
- Demonstrem respeito uns com os outros.

O Scrum Master

Em geral o trabalho do Scrum Master é garantir que a equipe Scrum siga a metodologia Scrum, suas práticas e regras.

O Scrum Master faz parte da equipe Scrum e atua como um líder-servo da equipe. Em princípio, este pode ser um trabalho de tempo integral forçando o Scrum Master a não contribuir diretamente com os resultados da Sprint. Com o passar do tempo, após algumas Sprints os processos irão se solidificar e o fardo do Scrum Master irá diminuir e o mesmo poderá contribuir diretamente para a Meta da Sprint.

Sendo crucial a confiança entre a equipe e o Scrum Master, é ideal que a equipe escolha o Scrum Master. No entanto, é muito mais comum que a gerência escolha o mesmo. Para que o Scrum Master consiga a confiança do resto da equipe é importante que o mesmo não tenha nenhuma superioridade autoritária sobre o resto da equipe. De outra forma, a comunicação aberta e decisões terão deficiência.

Scrum Product Owner

O Product Owner (ou PO) tem função central dentro da estrutura Scrum. A maioria das responsabilidades do gerente de produtos clássico e do gerente de projetos se combinam em uma função única.

Ele representa o usuário final e/ou os Stakeholders, e é responsável por maximizar o valor do produto ao garantir que o trabalho correto é feito e no tempo certo.

Como consequência, o Product Owner tem que trabalhar muito próximo à Equipe Scrum e coordena suas atividades por toda vida do projeto. A ninguém mais é permitido indicar as prioridades de trabalho para a equipe Scrum.

O Product Owner tem várias responsabilidades:

- Gerenciar o Product Backlog (Backlog do Produto);
- Gerenciar os lançamentos ou releases;
- Gerenciar os Stakeholders (envolvidos/interessados);
- Trabalhar de perto à Equipe Scrum.

Claro que o Product Owner pode delegar certas atividades (como fisicamente manter o Backlog do Produto), mas no final ele continua o responsável.

Artefatos Scrum

Backlog do Produto

A definição mais simples do Backlog do produto é uma lista com tudo que o projeto precisa, sendo uma lista viva que pode crescer ou diminuir durante a vida útil do projeto. Essa lista substitui os artefatos tradicionais de especificação e requisitos. Esses itens podem ter certa natureza técnica e podem até serem centralizados no usuário final, na forma de estórias de usuário (user stories). O dono do Backlog do Produto é o Product Owner. O Scrum Master, a equipe Scrum e outros Stakeholders podem contribuir para definir uma lista completa de tarefas para os itens do Backlog do Produto.

Outros artefatos podem ser usados pela equipe Scrum para facilitar a compreensão dos itens do Backlog do Produto, no entanto, esses artefatos não substituem o Backlog do Produto, seus detalhes ou conteúdo.

O Product Owner usa o Backlog do Produto durante a reunião de planejamento da Sprint para descrever seus itens para a equipe. Então, a equipe Scrum determina quais itens podem completar durante a próxima Sprint.

User Stories (Estórias de Usuário)

Os itens do Backlog do Produto são geralmente escritos na forma de Estórias de Usuário. Uma Estória conta brevemente sobre alguém usando o produto. Contém um nome, uma breve narrativa, um critério de aceite e condições para que a estória seja completada. A vantagem de estórias de usuário são que focam exatamente no que o usuário precisa/quer sem entrar em detalhes de como fazê-lo.

Sprint Backlog

Dentro do Sprint Backlog estão todas as atividades requeridas para completar os itens comprometidos e listados no Backlog do Produto. Todos os itens devem ser estimados na base de horas por pessoa para que o progresso e esforço pendentes sejam acompanhados.

O Sprint Backlog é um artefato vivo e deve ser atualizado diariamente. Se um membro da Equipe começar a trabalhar em alguma atividade, seu nome deve ser gravado nessa tarefa do Backlog. Novas atividades podem ser adicionadas ao Sprint Backlog durante a Sprint. Ao final do dia todos os esforços remanescentes devem ser atualizados e isso irá definir quanto trabalho ainda falta até que a Meta da Sprint seja alcançada. A "Definição de Pronto" (abaixo) é usada para decidir quais itens estão prontos ou não.

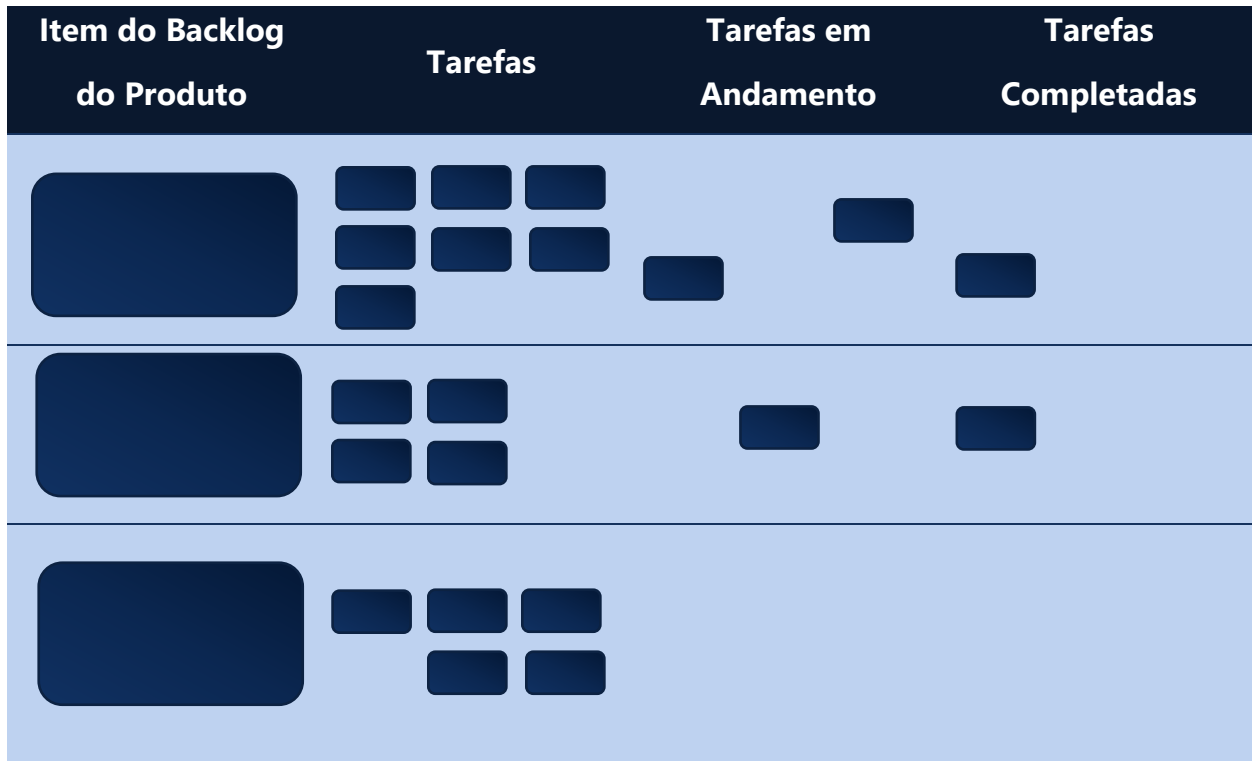
O Sprint Backlog pode ser mantido eletronicamente em uma planilha de Excel ou em cartões de um quadro. O último tem mais vantagens (transparência e fácil acesso) mas também adiciona certa complexidade se a Equipe Scrum estiver distribuída em múltiplos lugares físicos (assunto discutido posteriormente).

Existem muitas outras ferramentas disponíveis no mercado, inclusive ferramentas proprietárias e comerciais:

- Atlassian
- Apprenda
- Borland
- CA Technologies
- HP
- IBM
- Polarion
- RallyDev
- Serena

- Smart Bear
- Team Forge
- Tech Excel
- Thoughtworks
- Visual Studio ALM/DevOps
- Visual Studio Online

A figura abaixo mostra um exemplo de como tal quadro pode ser organizado. A estrutura deve se adaptar para refletir as necessidades do projeto.



Técnicas de Estimativa

Estimativas de Esforço

Todos os itens do Backlog do produto têm uma estimativa que permite ao Product Owner priorizar os itens e planejar os lançamentos. Isso significa que o Product Owner necessita de avaliações honestas sobre a dificuldade do trabalho a ser feito. Apesar disso, é recomendado que o Product Owner não participe da estimativa para evitar pressionar (intencionalmente ou não) a Equipe Scrum.

A estrutura Scrum não prescreve um modo único de estimar o esforço. No entanto, dentro da estrutura Scrum a estimativa não é normalmente feita em termos de tempo – uma métrica mais

abstrata de quantificação de esforço é usada. As estimativas mais comuns usadas incluem tamanhos entre 1 e 10, tamanhos de camisetas (PP, P, M, G, GG, GGG) ou a sequência Fibonacci (1,2,3,5,8,13,21,34, etc.).

O importante é que a equipe compartilhe um entendimento comum com a escala que usam, para que cada membro da equipe se sinta confortável ao usá-la.

Planning Poker® / Poker Scrum

Em um ambiente de desenvolvimento de software, constantemente nos deparamos com a necessidade de estimar o esforço necessário para a realização de uma tarefa.

O problema é que numa área de atuação que envolve **trabalho criativo** e soluções baseadas em **esforço subjetivo** como é a área de Desenvolvimento, Design ou Marketing, é bem provável que você terá divergências quanto à estimativa de esforço dada por cada pessoa do seu time.

Para facilitar este processo, **Mike Cohn** em seu livro *Agile Estimating and Planning* [n] fala sobre esta técnica de estimativas de tamanho de tarefas voltadas para as metodologias ágeis e Desenvolvimento de Software.

O Planning Poker® (também chamado de Poker Scrum) se baseia na ideia de estimar ideias através de um jogo de cartas, onde todos os envolvidos no processo devem fazer uma estimativa pessoal com base no seu conhecimento pessoal e na análise de uma tarefa, e o objetivo final é que haja um consenso através de discussão e argumentação rápida para tal.

Ao usar o Planning Poker®, influências entre participantes são minimizadas resultando em uma estimativa mais precisa. [m]

Para se jogar Planning Poker® os seguintes itens são necessários:

- A lista de funcionalidades a serem estimadas;
- Um baralho de cartas numeradas.

Um típico baralho contém a sequência Fibonacci incluindo o zero: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89; outras progressões similares também são possíveis. A razão para usar a sequência Fibonacci é refletir a incerteza em estimar itens grandes. Uma estimativa grande geralmente significa que a estória não está bem compreendida em detalhes ou deveria ser dividida em estórias menores. Estórias menores podem ser estimadas em maiores detalhes. Seria um desperdício de tempo discutir se é 19, 20, ou 25; A estória é simplesmente (muito) grande.

O jogo pode ser jogado seguindo os seguintes passos:

- O Product Owner apresenta a estória a ser estimada. A equipe Scrum faz perguntas e o Product Owner explica em mais detalhes. Se muitas estórias precisam ser estimadas, uma caixa de tempo (ex. somente um minuto por explicação) pode ser especificada. Se a restrição de tempo for alcançada e a Equipe Scrum ainda não entendeu a estória então é sinal de que a estória precisa ser reescrita;
- Cada membro da Equipe Scrum escolhe secretamente a carta representando sua estimativa;
- Depois que todos escolherem uma carta, todas as seleções são reveladas;
- Indivíduos com estimativas altas ou baixas explicam suas estimativas;
- O jogo se repete até que todas as estórias estejam estimadas;
- Se a discrepância numérica das estimativas for muito grande, possivelmente a estória não tenha sido completamente compreendida;
- O ideal é que todos escolham estimativas iguais ou similares, sendo priorizado o valor numérico da estimativa com maior repetição escolhida (ex. 5 membros da equipe escolheram 3, e 2 membros da equipe escolheram 1, a estimativa usada será a maioria, ou seja, 3).

Gráfico Burndown

O gráfico de Burndown Scrum é uma ferramenta de medição visual que mostra o trabalho completado diariamente em comparação ao ritmo projetado para o lançamento atual do projeto. Seu propósito é permitir que o projeto permaneça dentro do planejado e que o lançamento fique dentro da agenda pretendida.

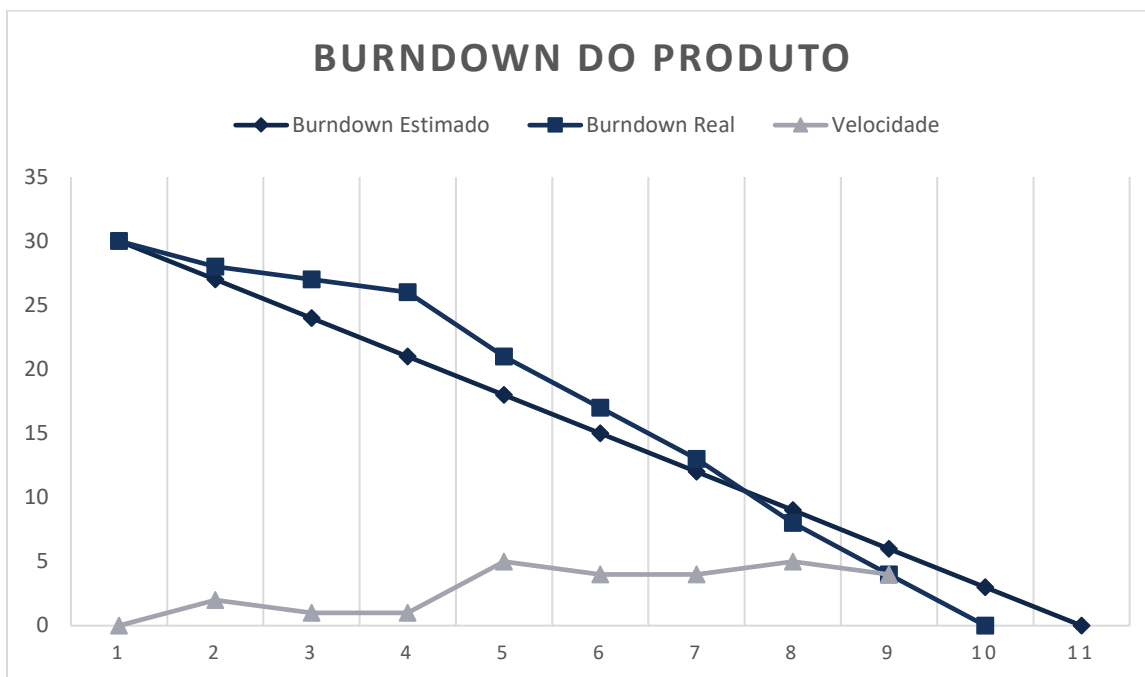


Gráfico simples de Burndown

O ritmo do progresso de uma equipe Scrum é chamada de "velocidade". Essa, expressa a quantidade de estórias ou pontos que foram completados por iteração. Uma regra importante para calcular a velocidade é que somente são contadas as estórias que forem completadas ao final da iteração. A contagem de trabalho parcialmente entregue (ex. somente código sem testes) é estritamente proibida.

Depois de algumas Sprints a velocidade de uma Equipe Scrum certamente será previsível e permitirá uma estimativa mais precisa sobre o tempo necessário até que todos os itens do Backlog do Produto sejam completados. Se a velocidade da Equipe Scrum é de 30 estórias e o total de trabalho restante é 155, poderemos prever que serão necessárias 6 Sprints para completar todas as estórias do Backlog.

Apesar de tudo isso, os itens do Backlog do Produto irão mudar durante a duração do projeto. Novas estórias serão adicionadas e outras estórias serão mudadas ou até removidas. No gráfico de Burndown simples a velocidade da Equipe Scrum e as mudanças de escopo não podem ser distinguidas. Para refletir isso, outros tipos de diagramas são usados.

Definição de Pronto

Para que se decida quando uma atividade do Sprint Backlog foi completada, a "Definição de Pronto" (DP) é usada. É uma lista de verificação completa com as atividades necessárias para garantir que somente as funcionalidades realmente completadas são entregues, não somente em termos de funcionalidade, mas também de qualidade. A Definição de Pronto pode variar de uma Equipe Scrum para outra, mas deve ser consistente dentro da Equipe.

Podem haver "Definições de Pronto" diferentes em vários níveis:

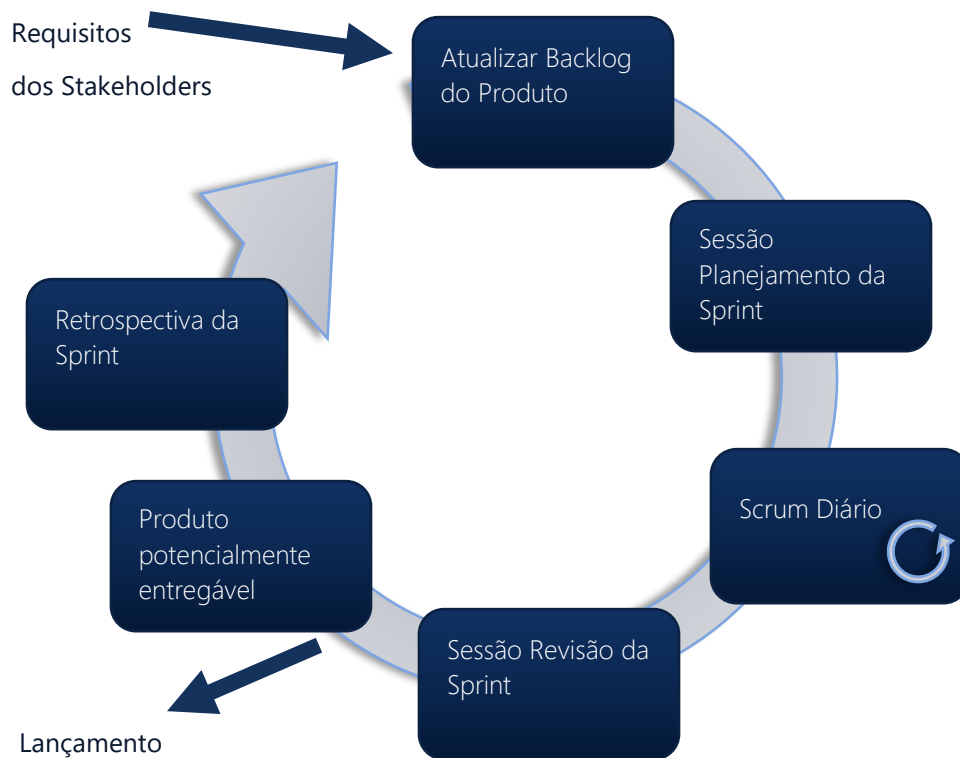
- DP para um item do Backlog do Produto (ex. escrevendo código, testes e toda documentação necessária);
- DP para uma Sprint (ex. instalação do novo sistema demo para revisão);
- DP para lançamentos (ex. redigir notas de lançamento).

Planejamento

Sprint

Na estrutura Scrum todas as atividades necessárias para a implementação dos itens do Backlog do Produto são desempenhadas dentro de uma Sprint (também chamada iteração). Sprints são sempre curtas: normalmente entre 2 a 4 semanas.

Cada Sprint segue um processo definido, como abaixo:



O Processo Sprint

Cada Sprint começa com duas sessões de planejamento para definir o conteúdo da Sprint: a reunião O QUE, e a reunião COMO. A combinação dessas duas reuniões também é conhecida como Reunião de Planejamento da Sprint. Na reunião O QUE a Equipe Scrum se compromete com as estórias do Backlog do Produto e usa a reunião de COMO para dividir essas estórias em tarefas menores e concretas. Então as implementações começam.

Ao final da Sprint, uma reunião de Revisão da Sprint é conduzida a fim de que o Product Owner possa verificar se todos os itens foram completados e implementados corretamente. Além disso, a Retrospectiva da Sprint é conduzida para verificar e melhorar os processos de execução do projeto: O que foi bom durante a Sprint, o que deve continuar como está e o que deve ser melhorado.

Durante a Sprint uma reunião curta chamada Scrum Diário (ou Reunião de Pé) é feita para atualizar os estados de cada item e para ajudar a auto-organização da equipe.

Reunião de Planejamento de Sprint

Cada Sprint e cada Reunião de Planejamento da Sprint começam com a reunião do O QUE. A meta para essas sessões é definir um Backlog da Sprint realístico que contenha todos os itens que possam ser completamente implementados até o final da Sprint.

Preparação

Para que uma reunião de O QUE obtenha sucesso, certos preparativos são necessários:

- O Product Owner define a Meta da Sprint;
- Com base na meta os itens relevantes do Backlog do Produto são escolhidos pelo Product Owner;

- Os itens são atualizados e divididos em estórias menores para que sejam completadas durante uma Sprint;
- Os itens são estimados e priorizados;
- A equipe define sua capacidade para a Sprint.

Meta da Sprint

O Product Owner define a meta da Sprint. Essa meta é uma curta descrição do que a Sprint irá tentar alcançar e, deve ser realista e compreensível a todos.

Capacidade da Equipe

A capacidade total da Equipe Scrum pode mudar de uma Sprint para outra. Para que os compromettimentos sejam realistas é necessário saber a capacidade total da equipe para a próxima Sprint considerando férias, feriados, esforços das reuniões de Sprint e tempo necessário para completar outras atividades durante a Sprint.

A sessão de Reunião

Durante a sessão o Product Owner apresenta a Meta da Sprint e discute sobre a mesma com a equipe. Depois disso, a Equipe Scrum itera por todos os itens do Backlog do Produto que sejam relevantes e se compromete com aqueles que pensa serem capazes de completar durante a Sprint. A decisão deve ser baseada na capacidade e compreensão disponíveis sobre os itens.

Ao final da sessão a lista de todos os itens comprometidos do Backlog do Produto dão base para a reunião de COMO e do Backlog da Sprint.

A Reunião de COMO

A meta da reunião de COMO é preencher o Backlog da Sprint identificando as tarefas necessárias a serem completadas para a implementação dos itens do Backlog do Produto. Tarefas normalmente incluem atividades de design, implementação, testes e documentação.

A reunião de COMO pode ser feita separadamente e depois da reunião de O QUE, ou mesmo durante a reunião de O QUE, após se comprometerem aos itens. Ou ambos.

Depois de identificar as atividades necessárias estas são estimadas pela equipe. A base dessa estimativa deve ser horas por pessoa. A equipe deve decidir quanto irá demorar para completar tudo que for requerido a fim de completar cada atividade.

Scrum Diário

A reunião de Scrum diário é curta e acontece todos os dias, de preferência no início do dia de trabalho. Cada membro da equipe que trabalha para completar uma tarefa da Sprint precisa participar. Durante a reunião, cada membro da equipe deve brevemente descrever respostas às seguintes três questões:

- O que ele/ela completou desde a última reunião de Scrum Diário?
- O que ele/ela irá completar até a próxima reunião de Scrum diário?
- Quais são os impedimentos para que ele/ela complete suas tarefas?

Todos os membros da equipe devem participar e devem estar de pé durante a reunião. O ideal é que a reunião de Scrum Diário dure não mais que 15 minutos. Nenhuma dúvida ou problema mencionado durante a reunião são ignorados por falta de tempo. Esses problemas ou dúvidas devem ser registrados pelo Scrum Master e devem ser abordadas mais especificamente após a reunião.

Reunião de Revisão da Sprint

Ao final de cada Sprint ocorre uma Reunião de Revisão da Sprint. Durante essa reunião a Equipe Scrum mostra quais itens do Backlog do Produto foram completados (de acordo com a definição de pronto) durante a Sprint. Isso deve ocorrer em forma de demonstração das novas funcionalidades.

É importante notar que os itens do Backlog que não foram completados não devem ser demonstrados. De outra forma isso pode sugerir que os itens estejam completados. Ao invés disso os itens incompletos/remanescentes devem voltar ao Backlog do Produto, estimados novamente e completados em uma Sprint posterior.

A Reunião de Revisão da Sprint deve ser informal. Não se aconselha uso de slides de PowerPoint® tampouco uso de muito tempo para preparação dessa apresentação. Durante a reunião o Product Owner inspeciona os itens de Backlog implementados e aceita a solução, ou adiciona novas estórias ao Backlog do Produto, para adaptar a funcionalidade.

Participantes na reunião de revisão da Sprint geralmente incluem o Product Owner, a Equipe Scrum e o Scrum Master. Ocasionalmente a gerência, clientes ou desenvolvedores de outros projetos podem participar.

Reunião de Retrospectiva da Sprint

Depois que a reunião de Revisão da Sprint terminar, o Scrum Master e a Equipe Scrum se juntam para a Retrospectiva da Sprint. Nessa reunião todos os membros refletem sobre a última Sprint e avaliam três coisas: o que deu certo durante a Sprint; o que não deu certo; e o quais melhorias podem ser feitas na próxima Sprint. Essa reunião deve ter tempo limitado. (Ex. 3 horas)

A retrospectiva da Sprint é parte integral do processo de “inspecionar e adaptar”. Sem essa reunião a equipe nunca será capaz de melhorar seu desempenho e não conseguirá focar na

melhoria coletiva da equipe. Portanto, sugestões de ações de melhoria de desempenho devem ser disponibilizadas ao final da reunião.

Planejamento de Lançamento (release)

O plano de lançamento é uma definição genérica e de alto nível com base em múltiplas Sprints. Esse plano reflete a expectativa sobre cada funcionalidade que será implementada e quando estarão disponíveis. Também serve como parâmetro de monitoramento do progresso do projeto. Os lançamentos podem ser pequenos pacotes intermediários durante o projeto, ou um pacote completo ao final dessa coleção de Sprints.

Para criar um plano de lançamento é necessário:

- Uma lista priorizada do Backlog do Produto;
- A velocidade (estimada) da Equipe Scrum;
- Condições de satisfação (metas para a agenda, escopo e recursos).

Dependendo do tipo de projeto (funcionalidade ou dados) o plano de lançamento pode ser criado da seguinte forma:

Para um projeto de funcionalidades, por exemplo, a soma de todas as funcionalidades dentro do lançamento pode ser dividida pela velocidade esperada. Isso irá resultar no número de Sprints necessárias para completar as funcionalidades requeridas.

Planejando um lançamento

Com base nos princípios de gerenciamento de projetos, o triângulo de ferro inclui três focos principais: Escopo, Recursos, Agenda.

A matriz de flexibilidade auxilia visualizar a maleabilidade de ajustes em cada um desses focos.

	Maior Flexibilidade	Flexibilidade Moderada	Menor Flexibilidade
Escopo	✘		
Recursos			✘
Agenda		✘	

Nota-se que dos três itens, o escopo tem maior flexibilidade, sendo ajustado constantemente, seja para adicionar ou para negociar funcionalidades.

O foco de menor flexibilidade são os recursos. Recursos incluem membros da Equipe e orçamento financeiro. Nesse caso, os recursos são os que causam maiores problemas se qualquer ajuste necessitar ser feito. Seja na contratação de novos recursos – que exige esforços de recursos humanos, recrutamento, negociação de salários, etc. – quanto nos gastos do projeto, que sempre são vistos negativamente por qualquer organização.

Para identificar lançamentos fundamentados nos itens prioritários da matriz de flexibilidade e critérios de lançamento podemos assumir que:

- **Número de funcionalidades de cada batch** – funcionalidades que visivelmente agregam valor tem maior impacto do que o número de funcionalidades lançadas.
- **Ciclos curtos e “magros”** – a maior frequência de lançamentos promove a positividade de aceitação e expectativa.
- **Agregar rápido, em pequenos incrementos** – funcionalidades, mesmo em pequenos números, que são importantes e são agregadas rapidamente, e com qualidade, causam maior impacto do que grandes entregas.
- **Atrasos, velocidade e previsão de Sprints** – quanto mais complexos ou maior o número de funcionalidades a serem lançadas em um batch, maior possibilidade de atrasos,

retrabalho (afetando a velocidade) e incertezas no planejamento de futuros batches. É melhor trabalhar com números pequenos e manter a margem de erro também pequena.

- **Tradicional, mas eficaz** – caso tudo mais falhe, o uso de atributos de metodologia tradicionais pode auxiliar o projeto a voltar aos trilhos.

Etapas ALM

Gerenciamento de projetos

Existem várias ferramentas de gerenciamento de projetos disponíveis no mercado ou mesmo na comunidade. Algumas dessas tem seu código aberto ou não possuem custos. A Microsoft oferece a solução Team Foundation Server e Visual Studio tanto em versões comerciais - para médias e grandes empresas – quanto versões de comunidade, que podem ser usadas por equipes pequenas, e não tem qualquer custo.

Para projetos que necessitam do uso de predecessores e gráficos Gantt, por exemplo, a Microsoft possui o MS Project.

O TFS é robusto para lidar com a maioria dos esforços de desenvolvimento de software, mas não possui funcionalidades relacionadas a predecessores / sucessores, ou caminho crítico.

Microsoft Project se aplica aos pontos em que o TFS é fraco, mas não possui funcionalidades para coleta de dados e na maioria das necessidades de desenvolvimento e monitoramento de projetos de software.

O TFS possui vários relatórios que podem ser usados por padrão, ou mesmo novos relatórios que podem ser personalizados e criado de acordo com as necessidades:

- Status de Bugs
- Tendências de Bugs
- Indicadores de Qualidade

- Builds com Sucesso / período
- Resumo do build
- Burndown e Burn Rate
- Reativações
- Trabalhos Restantes
- Visão geral de requisitos
- Progresso dos Requisitos
- Estados das iterações
- Progresso dos planos de Teste
- Trabalhos não planejados

Determinando o esforço

Em Scrum, o esforço de cada item do Backlog é definido pela Equipe de Desenvolvimento. Esse esforço é atribuído à complexidade de cada tarefa e não ao número de horas que cada item tomará. Para mais detalhes, vide tópico sobre Técnicas de Estimativa, p. 35

Metodologia incremental e iterativa

As metodologias ágeis, especialmente em Scrum, se estressa o caráter dinâmico de todos os fatores e estruturas dos projetos. Embora um item do Backlog possa estar concluído, novos itens de Backlog podem incrementar funcionalidades já existentes, ou mesmo ajustar seus propósitos. Portanto, não espere que um item de Backlog seja único e permanente. Funcionalidades podem evoluir durante o projeto e cada iteração pode acrescentar algum atributo novo a funcionalidades já existentes, ou mesmo criar novas funcionalidades que são dependentes de uma já implementada.

Não espere “adivinhar” funcionalidades futuras, permita que o Backlog do Produto evolua conforme as necessidades da organização, do mercado, e do projeto.

Requisitos

O refinamento do Backlog do Produto é um processo em que os itens do Backlog são revisados pela equipe Scrum, inserindo mais detalhes e garantindo o esclarecimento dos requisitos para esse item.

Qualquer item do Backlog do Produto deve ser refinado como e quanto informações adicionais forem necessárias. No entanto, a prioridade deve ser dada a refinar itens do Backlog que provavelmente serão incluídos na próxima Sprint. Isso garante que informações suficientes são conhecidas sobre cada item e permitem que sejam estimadas corretamente – geralmente durante a próxima reunião de planejamento da Sprint – e planejadas como serão entregues, e permitem que a equipe se comprometa a entregar esse e outros itens na próxima Sprint.

O refinamento do Backlog deve ser um processo constante. No entanto, algumas equipes descobrem ser útil planejar uma reunião no meio da Sprint que permita que os itens candidatos à próxima Sprint sejam discutidos. Essa sessão auxilia nas atividades de refinamento do Backlog, com revisões sendo feitas durante a sessão de Planejamento da Sprint.

Todos os membros da equipe devem estar envolvidos no refinamento do Backlog. Isso garante que toda equipe entenda o que é requerido de cada item do Backlog que eles podem incluir em uma Sprint. Além disso, podem incluir quaisquer especialistas em UI/UX, DBA, infraestrutura, etc. que podem ter envolvimento na entrega do item enquanto o mesmo estiver sendo refinado. [o] O produto final do refinamento do Backlog do Produto são itens claros e completos que definem exatamente o que se espera ao finalizar a entrega desse item, e que todos os membros da equipe saibam exatamente o que precisa ser feito. Esse refinamento inclui, mas não se restringe à:

- Critério de Aceite
- Comportamento

- Propósito
- Alternativas de comportamento
- Meta a ser alcançada

É importante lembrar que, mesmo depois de ser incluído em uma Sprint, um item pode ainda ser mais refinado, de acordo com as necessidades de entendimento e esclarecimento que possam surgir durante o decorrer da Sprint. No entanto, esse refinamento deve ser discutido e acordado entre a Equipe e Product Owner, com intuito de manter a visibilidade.

Pode ocorrer instâncias em que os requisitos possam ser muito complexos para que sejam mantidos em um único item de Backlog. Nesse caso, quebra-se o requisito em pedaços menores e gerenciáveis.

Para facilitar a priorização dos requisitos, identifique quais funcionalidades fazem parte do “caminho crítico” ou aquelas que **devem existir** (*must have*). Inclua o resto da equipe para que discutam quais funcionalidades são consideradas mais importantes e defina as dependências de cada funcionalidade.

Arquitetura

A fim de facilitar a compreensão dos requisitos e organizar o desenvolvimento das funcionalidades, é recomendado a criação de diagramas arquitetônicos relacionados aos objetos que serão criados, desenvolvidos e aprimorados. O Visual Studio oferece várias opções para criação de designs arquitetônicos, que podem ser incluídos nos requisitos, afim de esclarecer e documentar os itens de Backlog do Produto.

Designs arquitetônicos são:

- Bons para compartilhar conhecimento sobre o projeto
- Bons para colaboração
- Meios de impor padrões arquitetônicos, limitando ou que pode ser feito em cada camada

- Requisitos de arquitetura a serem validados durante um build
- Diagramas DGML podem ser estendidos para incluir mais do que elementos arquitetônicos

Codificação

O TFS e Visual Studio possuem várias ferramentas já inclusas que auxiliam no processo de codificação, a fim de aumentarem o desempenho e maximizarem a qualidade. Além dessas ferramentas já inclusas, há inúmeras outras ferramentas disponíveis para download – a maioria em código aberto e grátis – além de frameworks alternativas.

Dentre essas ferramentas, encontramos:

- Atalhos de teclado
- Navegadores de Código e arquivos
- Debuggers e IntelliTrace
- Métricas de Código
- Tarefas (Internas e TFS)
- Marcadores
- Integração com gerenciadores de repositório de código (Source Safe e Git)
- Revisão de Código, etc.

Para gerenciar o código produzido pela equipe, o TFS e Visual Studio contam com ferramentas de gerenciamento de código fonte que podem ser acessados tanto pelo portal do TFS quanto pelo próprio Visual Studio.

A fim de maximizar a qualidade do código produzido, recomenda-se:

- **Escrever código de fácil manutenção.** Código que seja claro e fácil de ser compreendido
- **Definir padrões de codificação.** Padrões já conhecidos como SOLID, TDD, etc.
- **Definir arquitetura do aplicativo.** Definir quais tecnologias ou frameworks serão usadas

- **Definir o processo de gerenciamento de configuração.** Como será a estrutura do código fonte, branching, merges, políticas de check-in, etc.

Testes e Qualidade

Testes unitários e testes automatizados auxiliam grandemente na qualidade do projeto pois, por estarem sempre sendo efetuados, encontram defeitos e Bugs rapidamente, dando tempo para os desenvolvedores de mitigarem os problemas enquanto o contexto do código ainda está vívido em sua lembrança e outras mudanças são mínimas. Ou seja, quanto mais rápido os consertos são aplicados, menores as chances de atraso no futuro e maior qualidade no produto lançado.

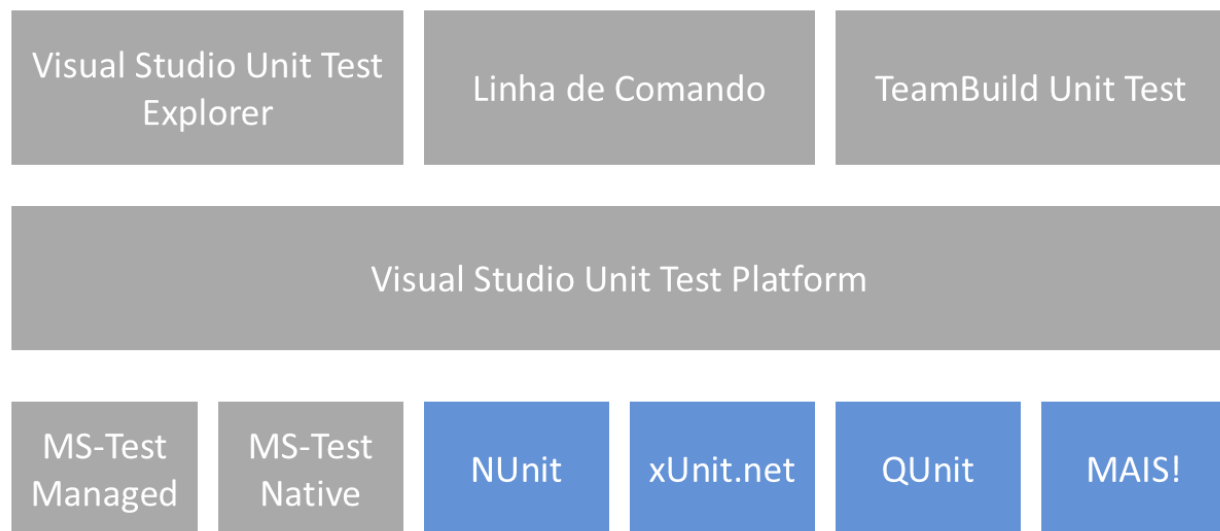


Figure 1 Ferramentas Disponíveis para Teste na Solução TFS / Visual Studio

a

Gestão de implantação

Além de gerenciar o desenvolvimento do produto, assegurando que seja testado e aprovado, o produto também deve ser gerenciado em seu processo de lançamento. Esse lançamento pode

ser em um servidor web, em um eCommerce, ou mesmo em um produto físico enviado a um cliente. Todas essas formas de lançamento exigem controle e organização, da mesma forma que nas fases anteriores do projeto. Se a implantação não for gerenciada corretamente, filas também podem ocorrer e o lançamento real do produto pode ser comprometido.

Há várias ferramentas de lançamento (deploy) automatizadas disponíveis comercialmente e pela comunidade. O TFS oferece a ferramenta de Release Management.

Dentre as atividades de implantação, podemos citar:

- Implementação de processos automatizados;
- Implementação de ambientes de pré-produção;
- Gerenciamento de feedback entre desenvolvimento e operações;
- Solucionar problemas em produção (em conjunto com operações).

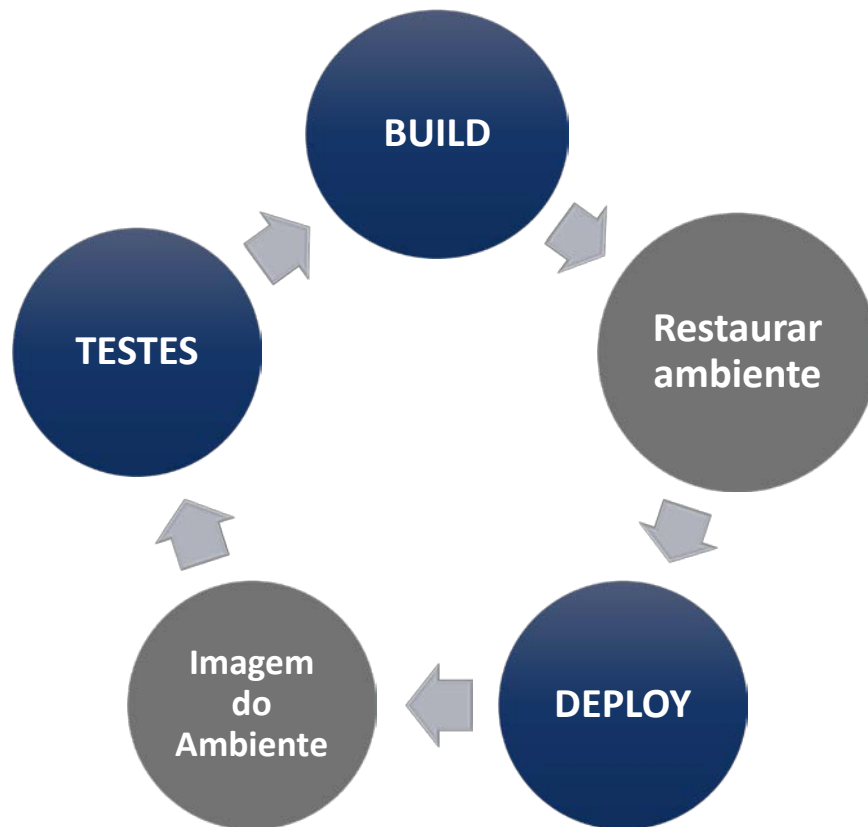
Operações

Operações gerenciam o produto após seu lançamento, ou seja, o produto em produção. No caso de problemas de disponibilidade (serviços) ou defeitos encontrados pós-lançamento. Operações incluem técnicos de suporte e, em algumas instâncias, desenvolvedores de plantão que fazem ajustes no código (dependendo da severidade). O método como as operações são gerenciadas pode causar enorme satisfação ou frustração aos usuários finais, dependendo de como os incidentes são tratados.

Integrar Desenvolvimento e Operações

Em produção, temos, por exemplo, servidores múltiplos, controles de patches, firewalls, scripts de defesa de servidores, balanceamento de carga, etc.

Em conjunto com o gerenciamento de lançamentos, criam um ciclo ou fluxo de gestão de ambiente, que engloba tanto o desenvolvimento, quanto o produto em produção.



xxx

Portanto, temos dois tipos de ambiente principais: **Desenvolvimento** (que inclui pré-produção) e **Produção**. Essa divisão pode variar e a nomenclatura ser diferente, no entanto, gerencia os ambientes de antes e depois do lançamento.

É aconselhável criar ambientes de laboratório que sejam livres de vícios, dependências ou defeitos que influenciem nos testes e na integração. A solução Microsoft inclui ferramentas para gerenciamento de ambientes.

Atividades

Atividade 1 [20 minutos]

Discuta as necessidades do seu projeto atual e defina a lista de ferramentas usada atualmente

- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____

Atividade 2 [25 minutos]

Discuta as necessidades do seu projeto atual e faça uma pesquisa no de possíveis ferramentas que possam auxiliar na melhoria contínua do seu projeto

Desenvolvimento

- _____
- _____

- _____
- _____

Lançamento

- _____
- _____
- _____
- _____

Operações

- _____
- _____
- _____
- _____

FAÇA O ESTUDO DO CASO A SEGUIR E RESPONDA AS ATIVIDADES 3 A 7 [60 MINUTOS]

Empresa PentaSoft é uma nova organização que foca em desenvolver software personalizado para empresas de gestão financeira. Essa fábrica de software tem 12 membros da equipe, entre seus desenvolvedores, técnicos de infraestrutura, e gestores.

A equipe decide implementar a metodologia Scrum e recebe seu primeiro projeto de uma empresa de gerenciamento financeiro de vale refeições. Um dos desenvolvedores exerce o papel de Scrum Master durante meio período e roda os relatórios uma vez por semana.

Esse cliente precisa de um sistema que valide automaticamente os pedidos de recarga submetidos pelos clientes / empresas que fornecem vale refeição a seus funcionários. Também precisa que a aplicação gerencie o uso do cartão em qualquer mercado ou restaurante que esteja cadastrado no sistema. A empresa recebeu várias regularizações do governo que são regidas por leis que devem ser seguidas à risca.

A Equipe optou por usar a versão Team Foundation Server Express e o Visual Studio 2015 Community Edition. Também optou por usar Git como seu repositório de código fonte.

Em sua equipe de infraestrutura, estão 4 técnicos que decidem se dividir em operações e suporte interno. O gestor tomou a função de Product Owner e se encarrega de inserir o esforço e valor de cada item do Backlog.

Os diretores da Empresa insistem em participar de todas as reuniões, inclusive do Scrum Diário, para ter maior controle e transparência do progresso do projeto.

Após três Sprints, o cliente percebe que precisa apressar a entrega devido às exigências legislativas que definiram multa para atrasos na implementação, que deve ocorrer em menos de 6 meses.

A Diretoria decide contratar mais 6 desenvolvedores temporários para conseguir terminar o projeto em tempo. No entanto, resolvem renegociar o orçamento do projeto, que agora é muito maior do que o planejado inicialmente.

Essa negociação demora 3 semanas para ser finalizada.

Ao final da 4 Sprint, o cliente retorna com feedback de que a funcionalidade principal – a que gerencia os pedidos de recargas dos clientes – não está processando direito, adicionando R\$ 0,04 centavos a mais em cada operação. Essa funcionalidade foi lançada na primeira Sprint e a Equipe se mobiliza por completo para encontrar o problema e mitigar a situação.

A equipe decide que a melhor opção é criar um ambiente de testes igual ao servidor onde o cliente está hospedando sua solução. Entretanto, o analista que construiu o ambiente de produção do cliente está de férias e só volta em 2 semanas. Nada foi documentado sobre o ambiente.

Atividade 3 [15 minutos]

Discuta e descreva quais os pontos positivos das decisões da organização em sua implementação do Scrum

Atividade 4 [15 minutos]

Discuta e descreva quais os pontos negativos e aponte as justificativas para considerá-los como negativos

Atividade 5 [15 minutos]

Discuta e descreva quais seriam as inspeções e adaptações feitas na equipe e organização para que o projeto voltasse aos trilhos

Atividade 6 [15 minutos]

Discuta e escreva sobre as escolhas de ferramentas da organização, licenças de software e ambientes

Atividade 7 [15 minutos]

O que a equipe poderia ter feito diferente, e como que esse estudo de caso se aplica ao seu projeto.

Avaliação

Para as questões 1 a 4

Você foi empregado como desenvolvedor na empresa ABC.com e a organização possui somente um Domínio de Active Directory, chamado ABC.com e usam Microsoft Visual Studio 2015 para seu desenvolvimento.

Questão 1

Você foi designado como Product Owner da nova aplicação ABC.com e tem em suas mãos uma lista de itens de Backlog do Produto (PBI) que incluem valores de negócio já definidos para a distribuição inicial da aplicação.

Uma equipe de desenvolvimento bastante experiente e um Scrum Master foram designados ao seu projeto. Você já estabeleceu a duração pretendida das Sprints.

Agora você está no processo de usar o Backlog atual como base do lançamento planejado. Você incumbiu a equipe de desenvolvimento de preparar avaliações de alto nível para completar cada item de Backlog. Você, então, aloca os itens de Backlog em várias Sprints para definir o lançamento.

Para completar o processo, você agenda o lançamento.

Quais das seguintes opções você deve usar para agendar o lançamento?

- a) Storyboard do Microsoft PowerPoint.
- b) Microsoft Project.
- c) Team Foundation Server.
- d) Planilha de Excel.

Questão 2

Você foi designado a garantir que a equipe use o processo de Scrum. Após criar o Backlog do Produto, você distribui os itens de Backlog e identifica as atividades necessárias para concluir uma Sprint.

Você quer garantir que os itens são alocados aos membros da equipe de maneira correta. Quais das opções a seguir está correta? (Escolha todas que se apliquem)

- a) Qualquer um dos membros da equipe pode alocar tarefas
- b) O número de tarefas que cada membro pode ter alocadas para si é determinado pelo WIP
- c) É possível trabalhar em um código sem ter que bloquear toda a estrutura do repositório
- d) Todas as anteriores

Questão 3

Sua equipe está em dúvidas se a quantidade de trabalho que se comprometeram a entregar está acima de sua capacidade para a Sprint atual. No entanto, a equipe não está segura quanto ao seu desempenho até o momento. Você instrui o Scrum Master a gerar o relatório de:

- a) Velocidade
- b) Burndown do Backlog do Produto
- c) Burndown da Sprint
- d) Tendências de Bug

Questão 4

A equipe de desenvolvimento está estimando os itens do Backlog e encontra dificuldades em compreender os requisitos de um item em específico.

Eles o convidam a participar de sua reunião de planejamento da Sprint e pedem auxílio.

Quais ações você toma:

- a) Instrui a equipe a buscar a documentação
- b) Instrui a equipe a entrar em contato com o cliente
- c) Esclarece as dúvidas da equipe e insere os mesmos na estória do item.
- d) Esclarece que os requisitos são desconhecidos e insiste que a equipe estime com bases no que já sabem.

Questão 5

A fim de maximizar o desempenho da equipe e viabilizar o sucesso do projeto, deve-se aplicar:

- a) Metodologia em Cascata, utilizando o Microsoft Project

- b) Metodologia Ágil, utilizando o Microsoft Excel
- c) Metodologia Scrum, utilizando o Team Foundation Server

Questão 6

As melhores práticas em WIP indicam que quanto mais itens estiverem sendo trabalhados ao mesmo tempo, maior probabilidade de entregar um valor agregado maior ao cliente.

- a) Verdadeiro
- b) Falso

Questão 7

Dentro do ciclo de ALM, há vários pontos que podem causar desperdícios. Aponte quais itens abaixo podem ser considerados como indicadores de desperdício:

- a) Superprodução
- b) Processamento
- c) Transporte
- d) Movimentação excessiva
- e) Todas as anteriores

Questão 8

Para que um projeto de desenvolvimento seja capaz de estar focado em ciclos menores de entrega, quais das opções a seguir indica o processo que deve ser aplicado.

- a) Reciclagem de atividades
- b) Retrospectiva da Sprint

- c) Inspeção e Adaptação
- d) Alternativas b e c estão corretas

Questão 9

Sua equipe foi instruída a controlar a submissão de código de um provedor de serviços externo. Qual das seguintes ações é mais recomendada?

- a) Considere restringir o acesso a check-in
- b) Use arquivos de arquitetura DGML para controlar o código
- c) Crie processos de gated check-ins e de build
- d) Crie um diagrama que especifique o processo de check-in

Questão 10

Para incentivar o feedback do cliente, para promover lançamentos sem bugs e para utilizar as ferramentas mais modernas de gerenciamento de ciclo de vida, quais ações podem ser tomadas?

- a) Crie uma solução de gerenciamento de ciclo de vida (ALM)
- b) Use o processo da Microsoft Security Development Lifecycle
- c) Use máquinas virtuais que são gerenciadas por terceiros
- d) Considere contratar desenvolvedores outsourced na Índia.

Referencias

[a] http://link.springer.com/chapter/10.1007%2F978-3-642-04133-4_13#page-1

- [b]** Green Book ALM (*Application Lifecycle Management*)
- [c]** <http://www.significados.com.br/zona-de-conforto/>
- [d]** <http://blog.smartbear.com/code-review/defining-code-quality/>
<https://zenexmachina.wordpress.com/2012/07/19/waterfall-vs-agile-a-knowledge-problem-not-a-requirements-problem/>
- [e]** Polanyi, M. (1958) Personal Knowledge
- [f]** Nonaka, I. & Takeuchi, H. (1995). The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation
- [g]** Tuckman, B. W. Developmental sequence in small groups. Psychological bulletin, v. 63, n. 6, p. 384, 1965.
- [h]** <http://agilemanifesto.org/>
- [i]** <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Portuguese-Brazilian.pdf>
- [j]** https://pt.wikipedia.org/wiki/Modelo_de_Tuckman
- [k]** <http://www.scrumguides.org/history.html>
- [l]** <http://www.significados.com.br/zona-de-conforto/>
- [m]** <http://blog.acelerato.com/agile/utilizando-planning-poker-para-estimar-seu-backlog/>
- [n]** <https://www.amazon.com.br/Agile-Estimating-Planning-Mike-Cohn/dp/0131479415>
- [o]** <https://www.scrumalliance.org/community/articles/2014/october/product-backlog-refinement>
- [p]** Anexo 1 - Processo de Desenvolvimento - Ciclo Completo
- [q]** <http://www.lume.ufrgs.br/handle/10183/127440>
- [r]** <https://msdn.microsoft.com/pt-br/library/ee156630.aspx>