

Formação em Gerenciamento Ágil de Projetos

► Preparatório para o exame PMI Agile Certified Practitioner (PMI ACP)®

**Formação essencial
para gestores de projetos ágeis**

Curso atualizado de acordo
com o último syllabus do
exame PMI ACP

Todos os direitos de cópia reservados. Não é permitida a distribuição física ou eletrônica deste material sem a permissão expressa do autor.

Versão: 2.0 Liberação: 10/03/17

Aviso de marcas registradas e direitos autorais

- Todos os direitos reservados. Nenhuma parte deste material poderá ser reproduzida ou transmitida em qualquer ou por qualquer meio sem a permissão escrita da TIEXAMES Consultoria e Treinamento Ltda.
- A TIEXAMES não licencia o uso de seu material para outras empresas. Se você encontrar outra empresa utilizando este material ou parte dele em treinamentos, por favor, denuncie pelo e-mail contato@tiexames.com.br.
- Algumas marcas registradas podem aparecer no decorrer deste curso. O uso destas marcas e logotipos é apenas para fins editoriais, em benefício exclusivo do proprietário da marca registrada, sem intenção de infringir as regras de sua utilização.

Ti.exames

Módulo 8



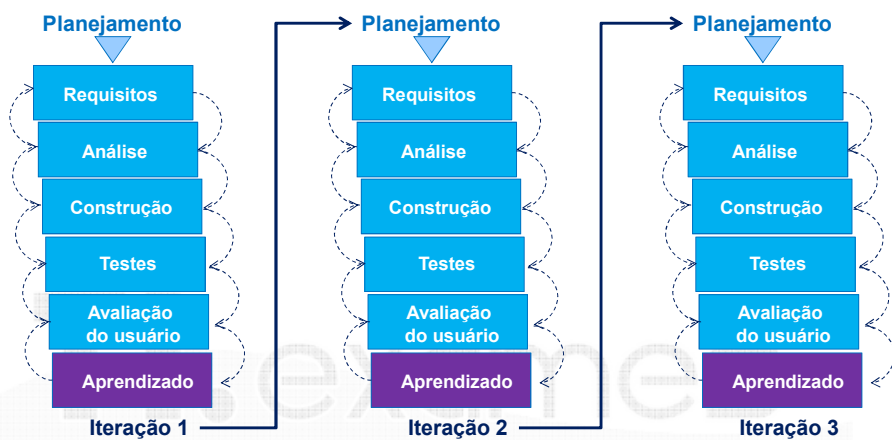
Práticas de melhoria contínua

Este módulo cobre:

- Conceitos iniciais sobre melhoria contínua
- Melhoria contínua de processo
- Melhoria contínua de produto
- Melhoria contínua de pessoas

Quando aplicar a melhoria contínua?

- É comum em projetos tradicionais a maioria das lições aprendidas serem capturadas somente no final do projeto para serem aplicadas em projetos futuros.
- Na abordagem ágil, procuramos aplicar os benefícios do aprendizado durante o projeto e o mais cedo possível conforme demonstrado no esquema abaixo.



Múltiplos níveis de melhoria

No Ágil, os esforços de melhoria contínua são apresentados em camadas como uma cebola – eles ocorrem em múltiplos níveis dentro do projeto.



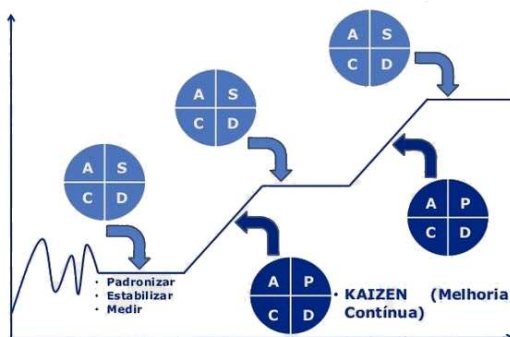
Kaizen e o PDCA

- Kaizen é a palavra japonesa para a melhoria contínua utilizando pequenas mudanças incrementais.

→ **KAI = Mudança / ZEN = Para Melhor**

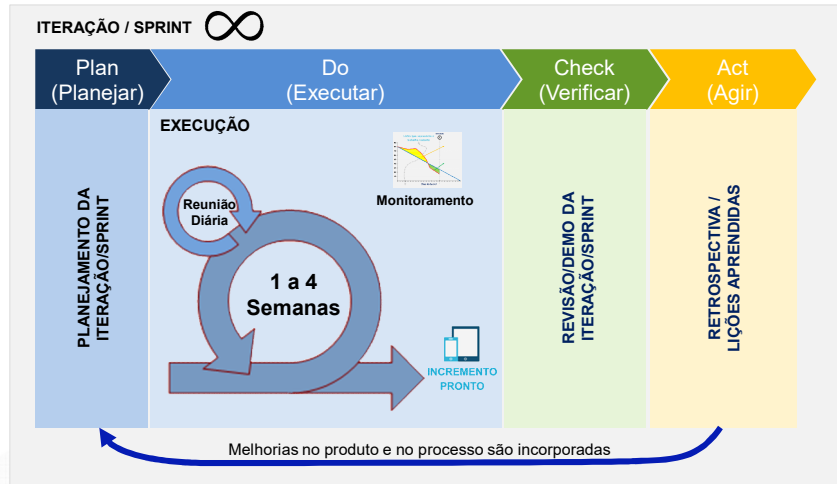
- É um processo diário de melhoria contínua, mas pode ser visto de forma genérica como uma “filosofia de melhorias”.
- O seu ciclo de atividades pode ser definido na sequência ao lado direito.

- O kaizen prega melhorias pequenas e incrementais em vez de mudanças radicais (conforme proposto pela reengenharia de processos).
- O ciclo PDCA é a principal ferramenta utilizada.



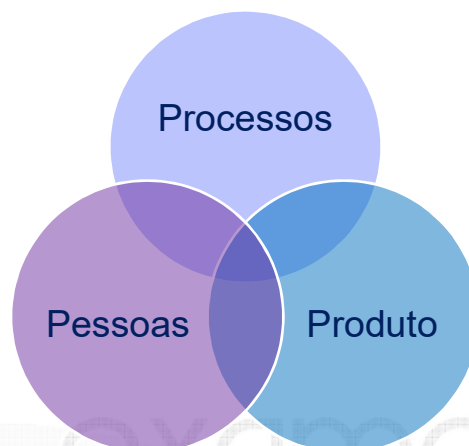
PDCA e o métodos ágeis

- A maioria dos frameworks ágeis, é então, um grande ciclo PDCA.



Onde aplicar melhoria contínua?

Em um ambiente ágil, as oportunidades de melhoria são observadas em três aspectos:



Módulo 8



Práticas de melhoria contínua

Este módulo cobre:

- Conceitos iniciais sobre melhoria contínua
- **Melhoria contínua de processo**
- Melhoria contínua de produto
- Melhoria contínua de pessoas

Ferramentas para melhoria de processo

Para avaliar um processo e identificar áreas de melhorias, nós podemos empregar ferramentas analíticas como:

Adaptação de processo (process tailoring)

Análise de processo

Mapeamento de fluxo de valor (value stream mapping)

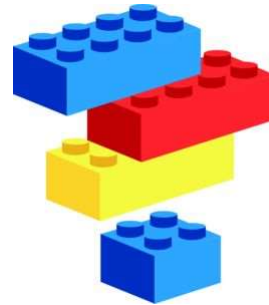
Pensamento sistêmico (System thinking)

Pre-mortems

Retrospectivas

Adaptação de processo (tailoring)

- Refere-se a adaptar a implantação do Ágil para melhorar seu funcionamento no seu ambiente de projeto.
- Entretanto, pode ser perigoso adaptar um framework ágil se você não tem entendimento dos motivos pelos quais as práticas foram propostas no framework.
- Recomenda-se para os times que são novos no Ágil que seja adotada uma metodologia na íntegra antes de tentar adaptá-la.
 - Isto porque os problemas que os novos times encontram com uma técnica ou prática padrão podem ser devido a sua falta de experiência no uso daquela técnica em vez de ser devido a questões relacionadas com a técnica em si.
 - Ao descartar uma prática antes da sua importância ser reconhecida, corre-se o risco de perder o benefício que a prática poderia trazer para o projeto.



Modelos híbridos

- Uma abordagem para personalizar um processo pode consistir em usar elementos de diferentes modelos, criando assim um modelo híbrido.
- Podemos fazer combinações com outros métodos ágeis ou até mesmo com uma abordagem tradicional. As principais combinações mais aplicadas são:

Híbrido Ágil – Ágil: Scrum - XP

- Estes dois modelos são praticamente complementares, visto que cada um tem foco em diferentes aspectos do projeto:
 - **XP**: fornece mais orientações técnicas do que gerenciais.
 - **Scrum**: fornece mais orientações gerenciais, mas não guia na forma de realizar o trabalho.

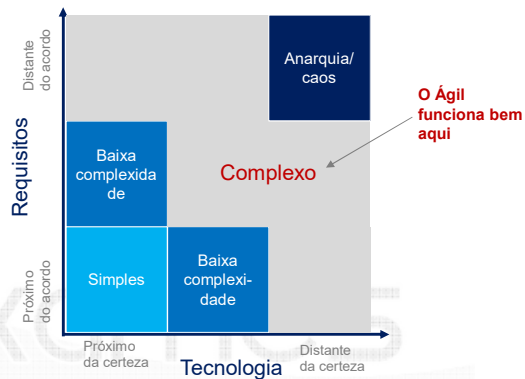


Híbrido Ágil – Tradicional

- Algumas vezes certas partes do projetos podem ser melhor executadas seguindo uma abordagem ágil e outras uma tradicional.
- Exemplo: Para validar o entendimento, achar a melhor solução, engajar o time, reduzir os riscos, é usada uma abordagem ágil. Para construir o produto, devido a não ver vantagem na forma iterativa e incremental, é usada uma abordagem tradicional.

Pensamento sistêmico (systems thinking)

- Quando o time está considerando mudar o seu processo, é útil entender o ambiente do projeto no nível de sistema. Este tipo de análise é chamado de **pensamento sistêmico**.
- Uma parte desta abordagem envolve classificar os projetos de acordo com a sua complexidade em duas áreas: **requisitos do projeto e abordagem tecnológica**.
- O Ágil funciona bem para projetos complexos que tem uma incerteza em relação aos requisitos e tecnologia, mas não é recomendado para aqueles que estão no classificação de caótico ou impossível de se ter controle.
- Nada impede de se obter benefícios com o Ágil também em projetos mais simples, mas usar práticas ágeis em projetos mais complexos é um grande desafio.



Análise de processo (process analysis)

- Envolve revisar e diagnosticar questões com os métodos ágeis do time.
- É a partir desta análise que é feita a decisão de adaptar ou não o processo.
- Algumas abordagens de análise de processo incluem:

Metodologia antipadrões

Metodologia padrões de sucesso



Metodologia antipadrões

Alistair Cockburn fornece uma lista de antipadrões (atributos ruins) para observar em nossos métodos:

Um tamanho para todos os projetos

É quase impossível criar uma metodologia para todos os tipos de projetos, todas as tecnologias e todos os tamanhos de time.

Intolerante

Uma metodologia pode ser muito rígida que não permita que o time tenha condições de fazer quaisquer adaptações que seja.

Pesado

Nem sempre uma metodologia pesada (com muitos artefatos, procedimentos, práticas) é mais segura. Uma metodologia pesada pode não dar a oportunidade de melhorar a chance de entregar o projeto com sucesso.

Enfeitado

Muitas vezes os times adicionam coisas aos seus métodos que eles pensam que estão fazendo, mas na prática não fazem e são coisas que tornam o processo mais caro ou leva a erros.

Não experimentado

Muitas metodologias criadas nunca foram experimentadas antes. Então, é melhor ver o que de fato funciona em um projeto e reusar do que criar algo totalmente novo acreditando que irá funcionar.

Usado uma vez

Uma metodologia que foi usada uma vez é melhor que aquela que nunca foi experimentada, mas ainda não é uma receita para o sucesso.

Metodologia padrões de sucesso

Além da sua metodologia antipadrões, Cockburn também lista sete padrões de metodologias de sucesso:

Comunicação cara a cara é o canal mais barato e rápido

Deveríamos tornar padrão este tipo de abordagem, estruturando o espaço do time e as reuniões do projeto para esta abordagem.

Excesso de peso na metodologia custa caro

A metodologia deve ser simples e leve para que em cima dela o time possa fazer melhorias e ter oportunidades de escolher outros caminhos que levam a eficiência.

Times maiores precisam de metodologias mais pesadas

Conforme o time cresce, a comunicação osmótica e conhecimento tácito se tornam mais difíceis de serem mantidos, exigindo assim a documentação escrita.

Projetos com criticidade maior requerem cerimônias maiores

Conforme a criticidade de um produto aumenta, o rigor associado com o desenvolvimento também deve aumentar. Exemplo: software para internet banking.

Feedback e comunicação reduzem a necessidade de entregas intermediárias

Em vez de escrever um documento para demonstrar entendimento dos requisitos, seria melhor mostrar imediatamente um protótipo.

Disciplina, habilidades e entendimento conta mais

Deveríamos considerar as pessoas espertas que podem realizar um trabalho exploratório e aplicar suas habilidades e entendimento que aquelas que seguem um processo formal ou criam documentação.

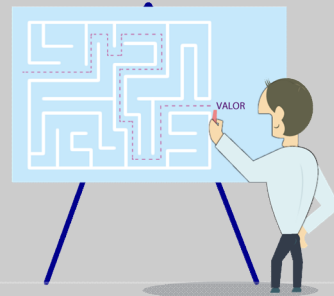
Eficiência é descartável em atividades sem gargalo

Não adianta melhorar o processo de coleta de requisitos se o gargalo está na codificação e testes. A eficiência tem que ser aplicada em todas as etapas.

Mapeamento do fluxo de valor (value stream)

- É uma técnica do *lean manufacturing* que tem sido adotada nos métodos ágeis.
- Sua proposta é otimizar o fluxo de informações e materiais requisitos para completar um processo, reduzindo assim o tempo que se leva para criar valor e eliminando desperdício ou qualquer trabalho desnecessário.
- O processo para aplicar esta técnica inclui os seguintes passos:

1. Identificar o produto ou serviço a ser analisado
2. Criar o mapa de fluxo de valor do processo AS-IS
3. Revisar o mapa e encontrar atrasos, desperdícios e restrições
4. Criar um novo mapa de fluxo de valor do processo TO-BE
5. Desenvolver um roteiro para criar o valor otimizado
6. Planejar visitar o processo no futuro para continuamente refiná-lo e otimizá-lo



Mapeamento do fluxo de valor – métricas

Para aplicar esta técnica é preciso entender as métricas a seguir:



Tempo de ciclo total

É a soma dos tempos de valor agregado e o de valor não agregado.



Tempo de valor agregado

É o tempo no qual as atividades estão gerando valor para o cliente.



Tempo de desperdício

É o tempo no qual as atividades não estão gerando valor para o cliente.



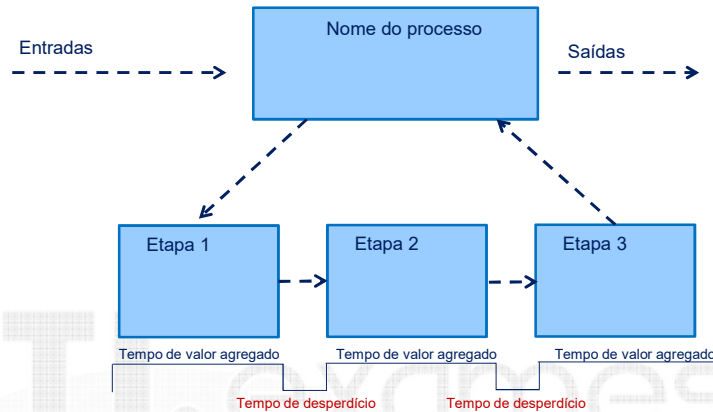
Eficiência do ciclo do processo

$$\text{Eficiência do ciclo do processo} = \frac{\text{Tempo de valor agregado}}{\text{Tempo de ciclo total}}$$

Podemos usar este indicador para comparar a eficiência entre processos.

Mapeamento do fluxo de valor - diagrama

O diagrama abaixo mostra como um processo pode ser mapeado aplicando esta técnica.

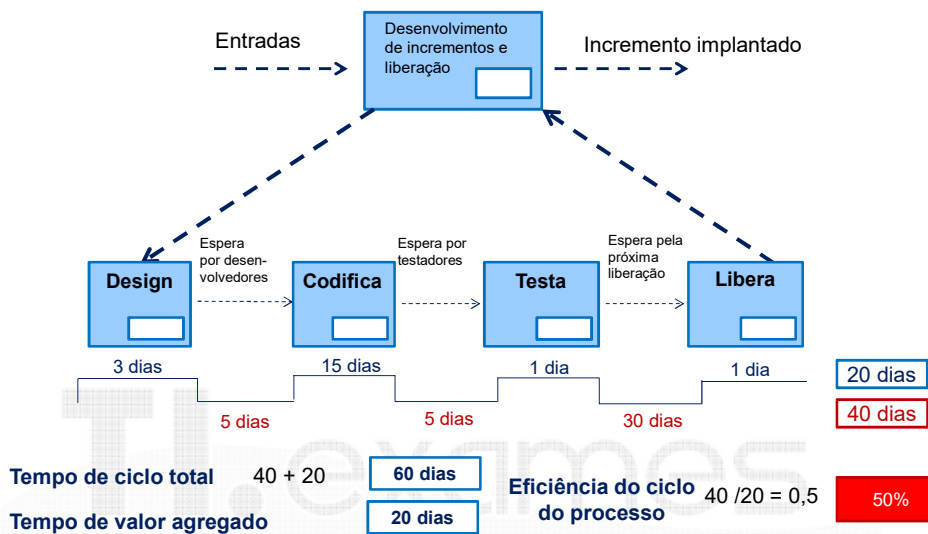


TI.exames © Todos os direitos reservados. Proibida a redistribuição deste material.

> Slide 19

Mapeamento do fluxo de valor - Exemplo

Unidade de trabalho: desenvolvimento e liberação de versão



TI.exames © Todos os direitos reservados. Proibida a redistribuição deste material.

> Slide 20

Mapeamento do fluxo de valor – Tipos de desperdícios

Abaixo são listados o 7 tipos de desperdícios mais comuns nos processos de desenvolvimento de software.

Desperdícios	Descrição	Exemplos
Trabalho parcialmente pronto	Trabalho iniciado, mas não finalizado	<ul style="list-style-type: none">▪ Código aguardando testes▪ Especificações aguardando desenvolvimento
Processos extras	Trabalho extra que não agrega valor	<ul style="list-style-type: none">▪ Documentação não utilizada▪ Aprovações desnecessárias
Funcionalidades extras	Funcionalidades que não são requeridas, embora sejam “legais”	<ul style="list-style-type: none">▪ Funcionalidades tecnológicas▪ “Folhear a ouro”
Multitarefas	Multitarefas entre diversos projetos diferentes	<ul style="list-style-type: none">▪ Pessoa em múltiplos projetos ao mesmo tempo
Espera	Esperas e atrasos para revisões e aprovações	<ul style="list-style-type: none">▪ Espera pela revisão do protótipo▪ Espera pela aprovação de documentos
Movimentação	Movimentações requeridas para comunicação/entrega de um grupo para outro	<ul style="list-style-type: none">▪ Times distribuídos▪ Times remotos
Defeitos	Produto defeituoso que necessita de correção	<ul style="list-style-type: none">▪ Bugs no software/documentação

Pre-mortem

- Consiste no time fazer um brainstorming para identificar possíveis pontos de falha em um projeto ou iteração **antes que eles aconteçam**, assim podendo evitar ou reduzir estes riscos.
- Esta técnica é especialmente valiosa em projetos ou iterações de longo prazo que tem mais chances de sofrer mudanças que projetos de curta duração.
- O exercício desta técnica é tipicamente empregado em quatro passos:



Imaginar a falha

O facilitador pede aos participantes para imaginar que eles estão no futuro após o projeto ter fracassado completamente.



Gerar razões para a falha

Os participantes, individualmente, por 3 a 5 minutos, desenvolvem uma lista de razões para o projeto ter fracassado.



Consolidar a lista

O facilitador pede para cada participante ler a sua lista e escreve os itens em um quadro.

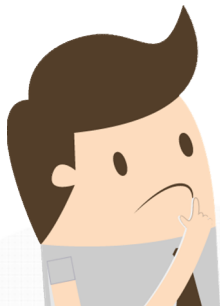


Revisitar o plano

O time revisa o seu plano para ver o que poderia evitar ou mitigar os riscos que eles levantaram.

Retrospectivas

- São eventos comuns para todos os métodos ágeis e servem para olhar para atrás e analisar uma situação para fazer melhoria.
- É uma oportunidade para inspecionar e melhorar seus métodos e seu de trabalho em equipe.
- É uma reunião realizada após uma release ou ainda após o projeto inteiro. Entretanto, o termo é mais utilizado para referenciar a reunião realizada a cada iteração, logo após a revisão da iteração (ou sprint).
- Durante esta reunião, basicamente três questões são levantadas:



O que está indo bem?

Quais áreas poderíamos aplicar melhorias?

O que deveríamos fazer de forma diferente?

Benefícios das retrospectivas

As retrospectivas oferecem vários benefícios:

Melhoram a produtividade

O time pode conseguir produzir mais trabalho "pronto" com as lições aprendidas e redução de retrabalho.

Melhoram as habilidades da equipe

As retrospectiva são um veículo para compartilhar conhecimento com as pessoas no time e com isto elas poderão realizar suas atividades com este conhecimento.

Melhoram a qualidade do produto

Podemos melhorar qualidade em nossos projetos achando circunstâncias que levam a defeitos e removendo as causas.

Melhoram a capacidade de trabalho

As retrospectivas focam em achar melhorias de eficiências nos processo, as quais podem melhorar a capacidade do time em fazer o seu trabalho.

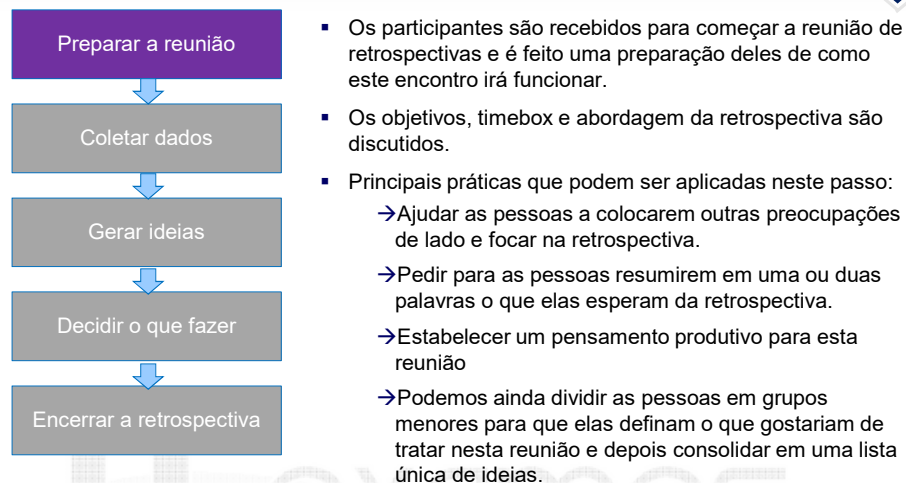
O processo da retrospectiva

O processo de retrospectiva pode ser realizado através de 5 passos:

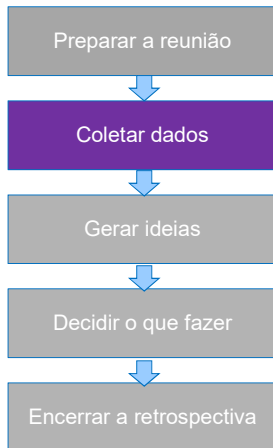
Tempos típicos para uma retrospectiva de 2 horas ao final de uma iteração:



O processo da retrospectiva – Passo 1: Preparar a reunião

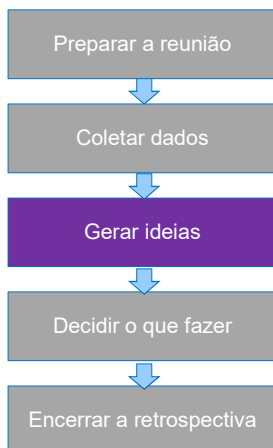


O processo da retrospectiva – Passo 2: Coletar os dados



- Aqui o time examina métricas, gráficos, histórias, reuniões, decisões, quantidade de código refatorado, entre outras coisas relacionadas a iteração (ou release, projeto, dependendo do foco da retrospectiva)
- Algumas práticas que podem facilitar o time a fornecer dados:
 - **Criar uma linha de tempo** e pedir aos participantes identificar com post-its os eventos bons, significativos e ruins que ocorreram.
 - Pedir aos participantes para escrever em post-its **coisas que funcionarem bem, coisas que não funcionaram tão bem e outras que somente desperdiçaram tempo**.
 - Também podem ser levantados dados em relação a **valores que apoiam o funcionamento do time** (respeito, comunicação, coragem, etc).
 - Usar **gráficos de controle de limites** para identificar se há problemas com a velocidade do time.

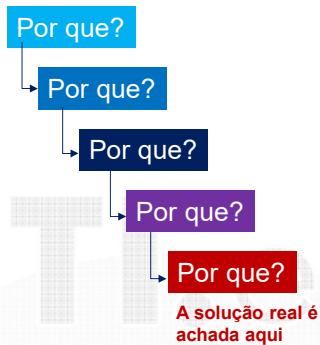
O processo da retrospectiva – Passo 3: Gerar ideias



- Aqui o time avalia os dados que foram coletados no passo anterior e faz uma interpretação deles para obter *insights*.
- O objetivo de gerar ideias ou insights neste passo é ajudar os membros do time a entender as implicações de suas descobertas e discussões.
- Espera-se ao final deste passo identificar as possíveis causas raízes para os problemas, o que será útil para tomar decisões no passo seguinte.
- Algumas práticas que podem ajudar neste passo:
 1. **Realizar brainstormings**
 2. **Aplicar a técnica 5 porquês**
 3. **Desenhar um diagrama de espinha de peixe**

A análise dos 5 porquês

Esta análise tem origem dentro da Toyota e do lean manufacturing e é usada também para achar a causa raiz de um problema através da identificação de um sintoma e então repetindo a questão "Por quê?" cinco vezes.



Exemplo de aplicação

O que aconteceu?

- Fomos bloqueados por causa de um time dependente que não completou sua parte do trabalho

Por que isso aconteceu?

- Bem, eles receberam o pedido muito tarde e não tiveram tempo para parar o seu outro trabalho e completá-lo

Por que eles receberam o pedido tão tarde?

- Nós não sabíamos sobre a dependência até que começamos a trabalhar na história

Por que não souberam sobre a dependência?

- Nós não tínhamos refinado a história antes de repassar

Por que a história não foi refinada antes de ser repassada?

- O PO não tinha a história preparada naquele momento

Por que o PO não tinha a história preparada?

- Não tínhamos visibilidade sobre o plano para a liberação

Por que não tivemos visibilidade sobre esse plano?

- Não fizemos nenhum planejamento de release

Diagrama de espinha de peixe (fishbone)

- Também é conhecido como diagrama de causa-e-efeito ou diagrama de Ishikawa.
- É usado para identificar fatores que contribuem para variações no desempenho de um processo ou outro tipo de problema em que se quer analisar possíveis causas.
- O problema/efeito é representado na cabeça do peixe e as possíveis causas são listadas em pequenas espinhas sob várias categorias.

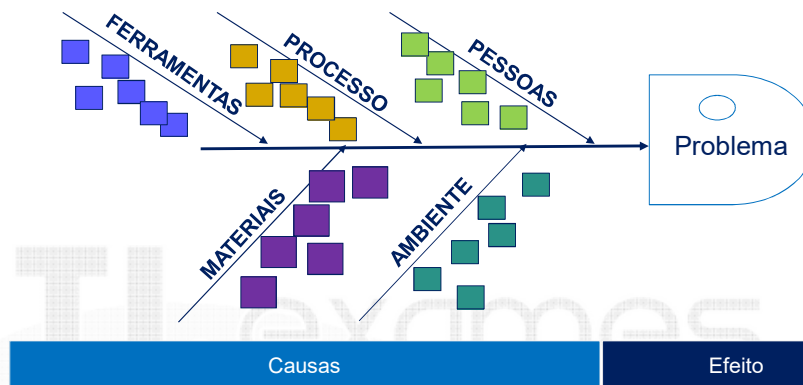


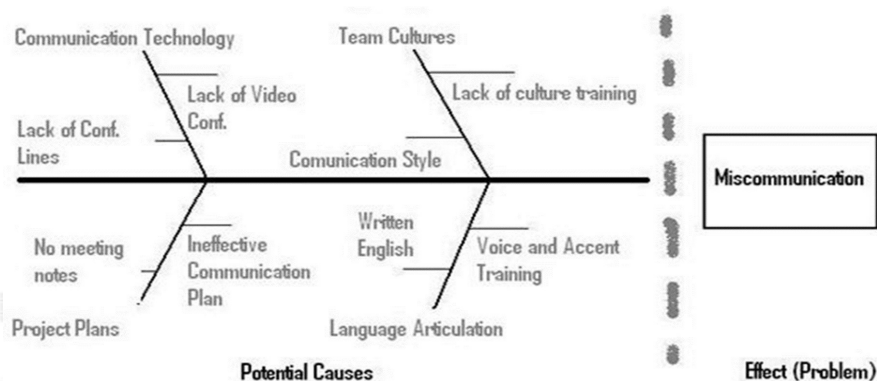
Diagrama de espinha de peixe (fishbone)

Procedimentos para usar o diagrama de espinha de peixe:

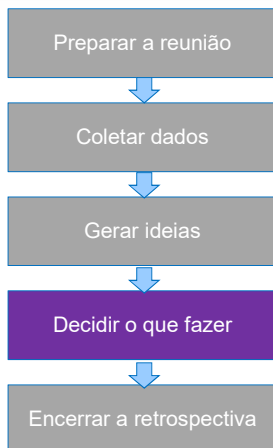
1. Acordar sobre a declaração de problema (também referenciada como efeito). Escrevê-la no centro de um flipchart ou quadro branco.
2. Fazer um brainstorm sobre principais categorias de causas do problemas. Se estas categorias forem difíceis de se achar, podem ser utilizadas as categorias genéricas:
 1. **Métodos**
 2. **Máquinas (ou equipamentos)**
 3. **Pessoas**
 4. **Materiais**
 5. **Mensuração**
 6. **Ambiente**
3. Escrever as categorias das causas como espinhas a partir da seta principal.
4. Fazer um brainstorm de todas as possíveis causas do problema. Pode ser utilizada a pergunta “por que isto acontece?” várias vezes.

Diagrama de espinha de peixe (fishbone)

- Problemas de comunicação são algo frequente em projetos de TI, especialmente quando os membros do time estão dispersos geograficamente. O exemplo abaixo lustra as possíveis causas das falhas de comunicação.



O processo da retrospectiva – Passo 4: Decidir o que fazer



- Aqui decidimos o que vamos fazer com os problemas identificados, ou seja, como resolvê-los.
- O resultado será um novo acordo de trabalho para a próxima iteração.
- É importante que as abordagens para solução sejam validadas e sejam criados objetivos mensuráveis para acompanhar o progresso e resolução do problema.
- Existem algumas práticas que podem ajudar a criar um plano de ação:
 1. **Estabelecer assuntos curtos usando uma estrela do mar**
 2. **Definir objetivos SMART**

Objetivos SMART

Ao estabelecer itens de ações é importante seguir estes critérios:

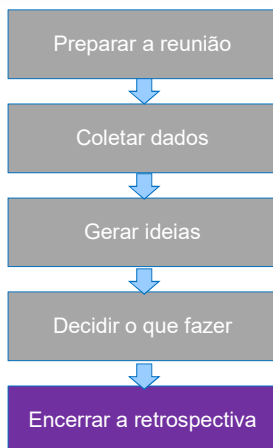
S	PECÍFICO Específica	Promessas vagas geralmente não geram resultados. Em vez de dizer “vamos tentar colocar as histórias para teste mais cedo”, poderíamos dizer “os desenvolvedores irão entregar pelo menos uma história a cada 2 a 3 dias para o QA”.
M	MEASURABLE Mensurável	É importante estabelecer uma forma de medição para verificar se um objetivo foi alcançado. Por exemplo: quantas histórias precisam ser completadas dentro de 2 a 3 dias?
A	ATTAINABLE Alcançável	É importante que o objetivo possa ser alcançado pelo time. Muitas vezes é mais apropriado utilizar metas de curto prazo que possam ser medidas durante o percurso que metas de longo prazo.
R	RELEVANT Relevante	Muitos objetivos podem dar a sensação de sucesso, mas podem não gerar nenhum benefício relevante. Por exemplo, qual seria o benefício de reduzir a duração da reunião diária de 15 para 5 minutos?
T	TIME BOUND Temporal	É importante que o objetivo tenha uma data limite para ser alcançado. Caso contrário, o time não irá se esforçar para que este seja alcançado.

Usando uma estrela do mar para representar as decisões

- Os diagramas são sempre úteis para ajudar nas discussões e podemos utilizar uma estrela do mar para representar as decisões com assuntos curtos.



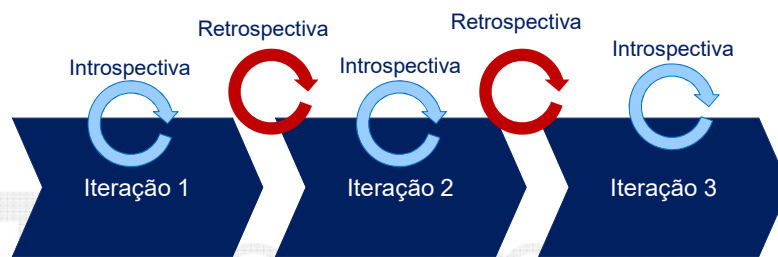
O processo da retrospectiva – Passo 5: encerrar a retrospectiva



- Aqui temos a oportunidade de refletir sobre o processo de retrospectiva e expressar nosso sentimento com cada um.
- Pode ser feito um resumo do que decidimos manter ou mudar, o que nós estamos satisfeitos e onde nós podemos fazer melhor o uso do nosso tempo nas próximas retrospectivas.
- Algumas práticas geralmente aplicadas aqui:
 - O time pode registrar em duas colunas: o que eles querem fazer mais e o que eles querem fazer menos.
 - Os membros fornecem feedback sobre a própria retrospectiva: o que ajudou, o que atrapalhou e como melhorar as retrospectivas futuras.
 - Os participantes expressam seu sentimento com cada um pelos esforços específicos durante a iteração.

Introspectiva

- As retrospectivas são muito úteis para o time identificar melhorias ao final de uma iteração para que na próxima iteração estas melhorias sejam implementadas.
- Entretanto, quando a duração da iteração é muito longa, nada impede o time de fazer reuniões curtas para tratar de questões pontuais durante a iteração ou ainda para propor melhorias a serem implementadas já na iteração em curso. Estas reuniões são chamadas de **introspectivas**.



Módulo 8



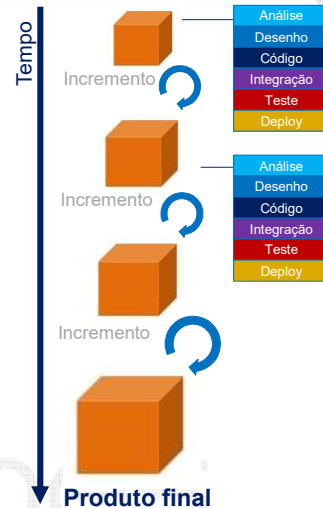
Práticas de melhoria contínua

Este módulo cobre:

- Conceitos iniciais sobre melhoria contínua
- Melhoria contínua de processo
- Melhoria contínua de produto
- Melhoria contínua de pessoas

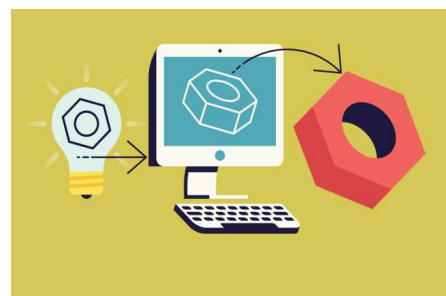
Melhoria contínua de produto

- Da mesma forma que nós refinamos os nossos processos, nós precisamos continuamente melhorar o produto.
- O desenvolvimento iterativo e incremental é uma forma de desenvolvimento contínuo, com o feedback do cliente nos dirigindo ao final.
- Quando o time constrói incrementos pequenos e coleta o feedback mais cedo, o produto evolui em direção aos verdadeiros requisitos de negócio. E algumas vezes os verdadeiros requisitos podem ser bem diferentes daqueles levantados no início.
- Por meio deste ciclo de desenvolvimento em pequenos incrementos, revisando, discutindo como melhorar, e então fazendo mais desenvolvimento e talvez enaltecendo algumas coisas, o produto ou serviço são construídos incrementalmente através de um processo de melhoria contínua.



Métodos de feedback

- São métodos de feedback: protótipos, simulações e demonstrações de funcionalidades.
- Na maioria das empresas é raro um produto ser construído duas vezes, portanto, para os usuários o produto que será desenvolvido geralmente será algo totalmente novo.
- Por isto, é importante que o cliente seja capaz de olhar algo antes de confirmar que o produto realmente possui as funcionalidades que ele precisa.



Módulo 8



Práticas de melhoria contínua

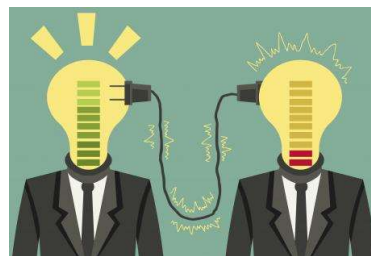
Este módulo cobre:

- Conceitos iniciais sobre melhoria contínua
- Melhoria contínua de processo
- Melhoria contínua de produto
- **Melhoria contínua de pessoas**

Compartilhamento de conhecimento

O compartilhamento de conhecimento acontece em vários níveis:

- Demonstração de produto é um exemplo óbvio, pois o seu propósito não é apenas apresentar o produto, mas também permitir o time saber o que funciona muito bem e o que não funciona. Nas demos pode ser possibilitado o seguinte tipo de diálogo:
 - **Time para o cliente**
 - **Cliente para o time**
- Time colocalizado é outro exemplo de compartilhamento de informação.
- As retrospectivas também são veículos para compartilhar o conhecimento.



Autoavaliação do time

- Também deve ser levada em consideração a melhoria contínua no aspecto humano do projeto.
- James Shore propõe uma técnica de autorreconhecimento que consiste na autoavaliação dos membros do time com relação aos seguintes tópicos:

Pensamento

- “Minha forma de pensar está sendo benéfica para o projeto?”

Colaboração

- “Estou trabalhando de forma colaborativa?”

Entrega

- “Estou comprometido com as entregas do projeto?”

Planejamento

- “Estou me planejando adequadamente para realizar os trabalhos do projeto?”

Desenvolvimento

- “Estou me desenvolvendo pessoal e profissionalmente no decorrer do projeto?”

Modelo de Jean Tabaka

Outro modelo para avaliação de time de alto desempenho é oferecido pelo “Jean Tabaka”. Este modelo investiga as seguintes áreas:

Auto-organização

O time é auto-organizado ou funciona em uma organização top-down baseada no comando e controle?

Empoderamento para tomar decisões

O time é empoderado para discutir, avaliar e tomar decisões ou ainda recebe decisões de uma autoridade externa?

Crença na visão e no sucesso

Os membros do time acreditam nos objetivos do projeto podendo inclusive resolver quaisquer problemas para alcançar estes objetivos?

Time comprometido

Os membros do time estão comprometidos com o sucesso do time em vez do seu sucesso individual?

Confiança mútua

O time tem confiança para trabalhar continuamente na melhoria de sua habilidade para agir sem medo ou bullying?

Tomada de decisão participativa

O time está engajado em decisões participativas em vez de depender de decisões autoritárias?

Orientação a consenso

As decisões do time são baseadas em consenso e não em um líder?

Desacordo construtivo

O time é capaz de negociar uma variedade de alternativas e impactos relacionados a uma decisão e, então, fornecer o melhor resultado?



Obrigado!