

**TI.exames**

**EDUCAÇÃO E TREINAMENTO**  
PARA ALAVANCAR A SUA CARREIRA

## Agile Scrum Foundation

Gerenciamento ágil de projetos com Scrum e outros frameworks ágeis

Preparatório para o exame EXIN Agile Scrum Foundation (ASF)

### Formação essencial para times ágeis

Product backlog → Sprint planning → Sprint backlog  
Daily Scrum (1-4 week sprint) → Box

**Curso atualizado de acordo com o Scrum Guide 2017**

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

Versão: 1.2 Liberação: 15/11/17

### 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](mailto: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**

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

Slide 2

## Módulo 2



# Mentalidade Ágil

**Este módulo cobre:**

- Desenvolvimento de software
- Modelo tradicional de desenvolvimento de software
- Lidando com problemas complexos
- Processos empíricos
- Agilidade
- Manifesto Ágil e seus 12 princípios
- Iterativo e incremental

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

Slide 3

### Módulo 2: Mentalidade ágil

## Desenvolvimento de software

- Todo desenvolvimento de software passa tipicamente pelas seguintes etapas:
  - Análise
  - Desenho/design
  - Desenvolvimento
  - Testes
  - Entrega
- Existem diferentes processos formalizados para o desenvolvimento de software, mas todos eles contêm tipicamente etapas semelhantes às listadas acima.

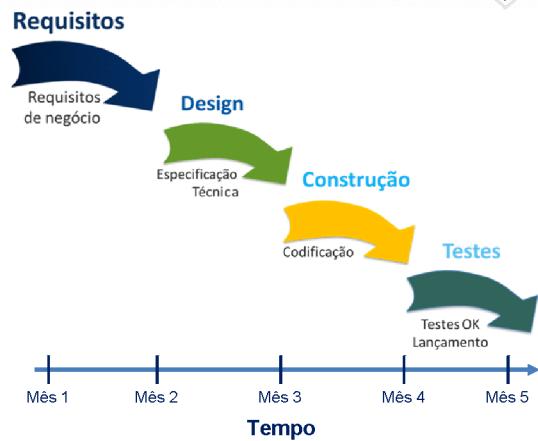
- O ponto principal aqui é a forma **como organizar estas etapas**, o que cria diferentes ciclos de vida.
- Alguns ciclos de vida são ágeis e alguns não.

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

Slide 4

## Modelo tradicional ou cascata (preditivo)

- O modelo em cascata (waterfall em inglês) é o mais tradicional.
- Surgiu na década de 1960.
- Neste modelo, o software é construído por fases.
- A principal característica deste ciclo de vida é que tudo é definido antes do início do desenvolvimento e há uma clara separação entre os processos.
- Esta característica de definir tudo antecipadamente e tentar cumprir é chamado de **preditivo** (predictive).
- Um software funcional somente é uma saída no final da cascata.



## Desenvolvendo software com métodos tradicionais

- Muitas organizações utilizam uma metodologia chamada cascata (waterfall) em seus projetos de desenvolvimento de software ou produtos.
- Dois tipos de pessoas são necessários:



### Pessoas de negócio

Definem o que o produto deve fazer. Para definir as necessidades de produtos, eles fazem previsões a longo prazo sobre o futuro:

- Qual será o valor do produto?
- Como ele será recebido pelo mercado?
- Quando o produto será liberado?



### Gerentes de projeto/ planejadores

- Eles planejam como construir o produto.
- Com base nas necessidades de produto que foram deduzidas a partir das previsões de negócios, eles planejam uma sequência de atividades como análise, design, construção, testes e assim por diante.
- Estimativas também são elaboradas para determinar o custo e o prazo do projeto.

Módulo 2: Mentalidade ágil

## Planos são os direcionadores

- O resultado do planejamento de projeto que usa métodos tradicionais é um **plano detalhado**.

```

graph LR
    A[Planejamento tradicional] --> B[Planejam o projeto com base em diversas premissas e projeções.  
Determinam o custo e o prazo do projeto.]
    B --> C[Planos de projeto detalhados]
    C --> D[Consideram a adequação ao plano como sucesso do projeto.  
Alterações no plano são vistas como resultado de mau planejamento.  
Mudanças são desencorajadas – longo processo de controle de mudanças.]
    D --> E[Sucesso é o cumprimento do plano!]
  
```

**Planejamento tradicional**

**Planos de projeto detalhados**

**Sucesso é o cumprimento do plano!**

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

Slide 7

Módulo 2: Mentalidade ágil

## Modelo tradicional ou cascata

- Um plano de projeto baseado no modelo cascata espera que a sequência de atividades seja executada de forma contínua sem a necessidade de mudanças.
- Há resistências às mudanças quando elas ocorrem. Não há o planejamento para retorno de atividades.

```

graph TD
    A[Necessidades do produto] --> B[Desenvolvimento no modelo tradicional]
    B --> C[Produto]
    subgraph B [Desenvolvimento no modelo tradicional]
        B1[Especificar] --> B2[Desenvolver]
        B2 --> B3[Testar]
        B3 --> B4[Liberar]
    end
    C --> D[Negócio]
    C --> E[Projeto]
    D --> F[O time de desenvolvimento executa as atividades em sequência, conforme o plano. Não tratam as mudanças de forma proativa, pois invalidariam o plano.]
  
```

**Necessidades do produto**

**Desenvolvimento no modelo tradicional**

**Especificar → Desenvolver → Testar → Liberar**

**Negócio**

**Projeto**

**O time de desenvolvimento executa as atividades em sequência, conforme o plano. Não tratam as mudanças de forma proativa, pois invalidariam o plano.**

**Produto**

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

Slide 8

Módulo 2: Mentalidade ágil

### Problema: feedback tardio

- Após a fase de Requisitos, o feedback é liberado.
- Se o feedback não é feito, os requisitos de negócios são liberados. Mas este feedback é importante.
- Feedback tardio é comum em projetos tradicionais.
- Produz um produto obsoleto.

**Requisitos**

Requisitos de negócios

**Design**

Especificação Técnica

**Construção**

**Feedback tardio**

**Detalhamento antecipado**

**Tempo**

Mês 1      Mês 2      Mês 3      Mês 4      Mês 5

Testes OK  
Lançamento

Produção

Ícones: Documento, Usuário, Caixa aberta.

**TI.exames** ® Todos os direitos reservados. Proibida a redistribuição deste material. > Slide 9

Módulo 2: Mentalidade ágil

### Detalhamento antecipado

- O modelo cascata segue o conceito **Big Design Up Front (BDUF)**.
- Os defensores do BDUF acreditam que o tempo gasto revisando exaustivamente a especificação garante a ausência de mudanças críticas na fase de execução.
- O modelo BDUF então faz com que o cliente precise definir tudo o que quer de forma antecipada.
- Como o cliente não terá "chance" de pedir novamente, a tendência é que este peça tudo, até algumas funcionalidades raramente utilizadas.

Diga-me tudo o que você precisa, agora.  
Vamos montar um projeto e ele depende do escopo.

Eu preciso de...

**TI.exames** ® Todos os direitos reservados. Proibida a redistribuição deste material. > Slide 10

Módulo 2: Mentalidade ágil

## Detalhamento antecipado

- Há diversos fatores que tornam questionável o uso de BDUF para projetos de software:
  - Requisitos não são completamente compreendidos no início do projeto
  - Usuários só sabem exatamente o que querem após verem uma versão inicial do software
  - Requisitos mudam durante o processo de desenvolvimento
  - Novas ferramentas e tecnologias tornam a estratégia de desenvolvimento imprevisível.
- Isso se aplica a diversos outros tipos de projetos.
- O BDUF é considerado adequado apenas para projetos estáveis, com pouca ou nenhuma mudança.

**TI.exames** © Todos os direitos reservados. Proibida a redistribuição deste material. > Slide 11

Como o conteúdo é modulado no desenvolvimento do software?

- Interativo
- Simples
- Incremental
- Escalão (waterfall)

### Quiz

Quiz - 6 questions

Last Modified: nov 20, 2017 at 11:12 PM

#### PROPERTIES

On passing, 'Finish' button: [Goes to Next Slide](#)

On failing, 'Finish' button: [Goes to Next Slide](#)

Allow user to leave quiz: [At any time](#)

User may view slides after quiz: [At any time](#)

Show in menu as: [Multiple items](#)

[!\[\]\(4fc2f7cc81639d6310529267a2ee602c\_img.jpg\) Edit in Quizmaker](#) [!\[\]\(714d70875eaf0e2f34d0a261eaf96dad\_img.jpg\) Edit Properties](#)

## Módulo 2 – Mentalidade Ágil

### ➤ Lidando com problemas complexos

TI.exames

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

➤ Slide 13

#### Módulo 2: Mentalidade ágil

### Desenvolvimento de software é complexo

- Desenvolvimento de software é um problema complexo. Tipicamente envolve muitas pessoas, tecnologia, conhecimento, regras de negócio complexas e exige diversas integrações em diferentes ambientes.
- Esta complexidade é ainda multiplicada pela presença de vários outros fatores:
  - **Evolução dos requisitos** – um componente que foi desenvolvido “hoje” para poucos usuários pode não funcionar quando possuir muitos usuários “amanhã”
  - **Desenvolvedores** que trabalham no projeto podem **não estar disponíveis** no futuro e a produtividade pode variar
  - A **tecnologia** utilizada hoje pode estar **obsoleta** amanhã
  - O mercado e conceitos de negócio evoluem com muita rapidez exigindo **alterações constantes no software**
- Tais problemas complexos dependentes do tempo são também chamados de problemas adaptativos complexos.
- A quantidade de itens “desconhecidos” no início de um projeto é realmente grande!

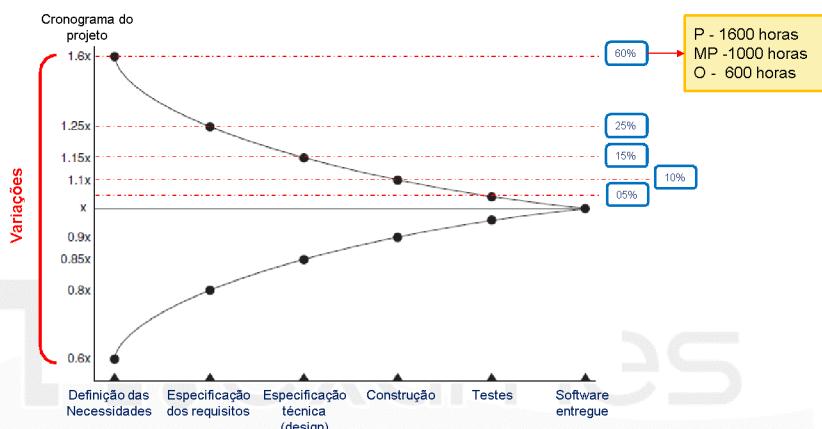
TI.exames

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

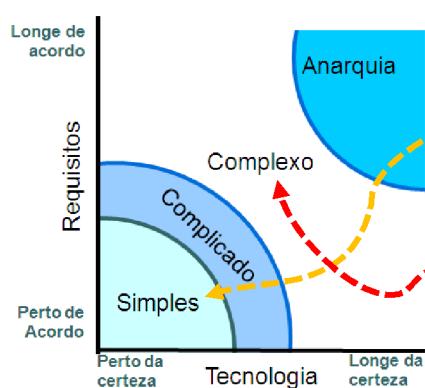
➤ Slide 14

## Cone da incerteza

- O planejamento tradicional usa técnicas científicas de planejamento para determinar prazo e custo logo no início do projeto.
- O problema é que em um sistema adaptativo complexo, quanto mais cedo, maiores são as incertezas, tornando o plano pouco assertivo.



## Processo empírico



### Processos definidos

São processos em que se conhecem todas as variáveis, há poucas ou nenhuma mudança e são repetitivos e previsíveis. Geralmente existe uma documentação aplicada na execução do processo.

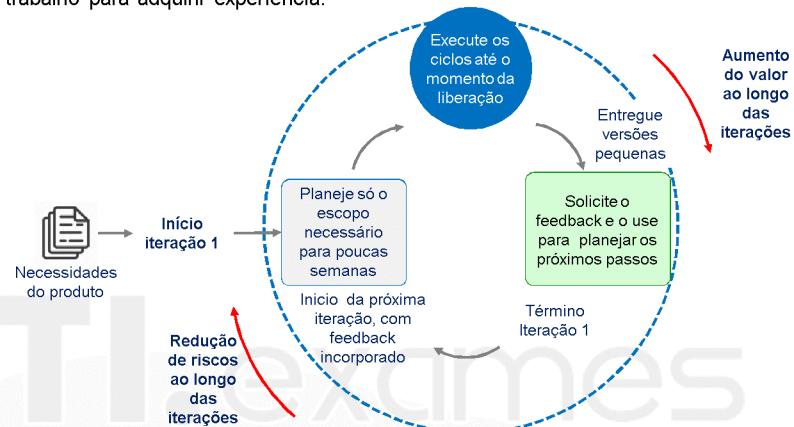
### Processos empíricos

São aqueles em que não se conhecem todas as variáveis, há mudanças e não são repetitivos nem previsíveis. Geralmente são baseados na experiência e no conhecimento de quem executa o processo.

O empirismo afirma que o conhecimento vem da experiência e devemos tomar decisões com base no que se conhece.

## Processo empírico

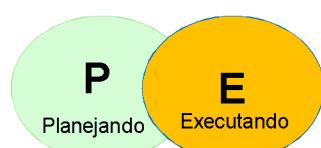
- A teoria do empirismo é baseada no conceito que problemas complexos são difíceis de serem previstos. O empirismo ajuda as pessoas a tratarem problemas complexos.
- Isso requer dar um passo de cada vez, com a realização de uma pequena quantidade de trabalho para adquirir experiência.



## Processo empírico planeja frequentemente

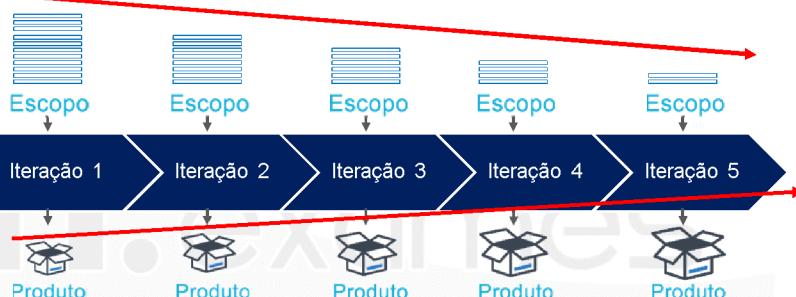
### Preditivo

Todo o planejamento é feito no início do projeto.



### Empírico (Adaptativo)

Planejamento é "just-in-time" e há replanejamento baseado em inspeção frequente dos resultados.



**Quiz**

Qual das seguintes declinações a FAÇA em relação ao conceito da inovação?

- As inovações são o resultado de uma estratégia de inovação.
- As inovações são o resultado de uma estratégia de inovação.
- As inovações são o resultado de uma estratégia de inovação.
- As inovações são o resultado de uma estratégia de inovação.
- As inovações são o resultado de uma estratégia de inovação.

**Quiz**

*Quiz - 2 questions*

Last Modified: nov 20, 2017 at 11:13 PM

**PROPERTIES**

On passing, 'Finish' button: [Goes to Next Slide](#)

On failing, 'Finish' button: [Goes to Next Slide](#)

Allow user to leave quiz: [At any time](#)

User may view slides after quiz: [At any time](#)

Show in menu as: [Multiple items](#)

 [Edit in Quizmaker](#)

 [Edit Properties](#)

19

## Módulo 2 – Mentalidade Ágil

# ➤ Agilidade

# TI.exames

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

➤ Slide 20

## Conceito de agilidade

- A palavra "Ágil" é um adjetivo, que significa:

“

Para mover-se rapidamente e facilmente, ou de pensar rapidamente e de uma forma inteligente.

”

- Ter rapidez e desembaraço.
- Entregar aquilo que gera o maior valor para o cliente, de forma cada vez mais rápida e com qualidade.
- Fazer as coisas complexas de forma simples.
- Ter a equipe toda comprometida com os objetivos do projeto.

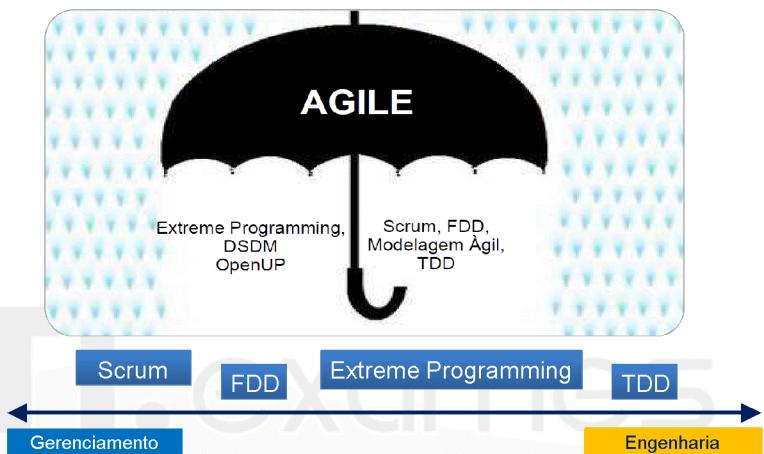
**Ter capacidade de responder rapidamente a mudanças.**

## Agilidade

Agilidade = <sup>ser</sup>ágil  
+  
 <sup>ser</sup>adaptativo  
+  
usar um **framework**

## O desenvolvimento ágil compreende vários frameworks

- O desenvolvimento ágil compreende vários frameworks e metodologias de desenvolvimento de software. É como se fosse um guarda-chuva que define características gerais que os modelos ágeis devem conter.



## O que precisamos fazer?

Precisamos:

- Estar mais próximos do negócio.
- Ter a consciência de que os requisitos mudam ao longo do tempo e que os usuários só sabem o que realmente querem quando usam o software.
- Demonstrar o software constantemente aos usuários e obter feedbacks de forma contínua.
- Aprender progressivamente sobre o produto (software).
- Ajustar os itens a serem desenvolvidos conforme o entendimento acontecer.
- Conversar mais e escrever menos.
  - ✓ Escrever somente aquilo que realmente é necessário.
- Estar preparados e aceitar as mudanças de forma natural.
  - ✓ Ações preventivas contra mudanças não devem ser realizadas.





**Quiz**

*Quiz - 2 questions*

Last Modified: nov 20, 2017 at 11:15 PM

**PROPERTIES**

On passing, 'Finish' button: [Goes to Next Slide](#)

On failing, 'Finish' button: [Goes to Next Slide](#)

Allow user to leave quiz: [At any time](#)

User may view slides after quiz: [At any time](#)

Show in menu as: [Multiple items](#)

 [Edit in Quizmaker](#)

 [Edit Properties](#)

**Módulo 2 – Mentalidade Ágil**

**► Manifesto Ágil**

**TI.exames**

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

► Slide 26

## Origem das abordagens ágeis

- Apesar de algumas metodologias ágeis existirem há mais de 20 anos, foi no ano de 2001 que o pontapé foi dado para a grande disseminação das abordagens ágeis.
- Nesta época os principais responsáveis pela teoria e pela aplicação de diversos métodos se reuniram para discutir sobre as melhores maneiras de gerenciar e desenvolver softwares. Entre os participantes deste grupo podemos destacar nomes como Kent Beck, Ken Schwaber e Jeff Sutherland.
- Destes encontros acabaria por sair o Manifesto Ágil e a formação da Aliança Ágil, uma organização sem fins lucrativos destinada a promover a adoção dos métodos ágeis.
- O Manifesto Ágil documentou os princípios orientadores para o desenvolvimento ágil e definiu uma filosofia em torno de um conjunto de metodologias existentes.



## Manifesto Ágil

### Manifesto para o desenvolvimento ágil de software

Estamos descobrindo maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Através deste trabalho, passamos a valorizar:

- Indivíduos e interação entre eles mais que processos e ferramentas
- Software em funcionamento mais que documentação abrangente
- Colaboração com o cliente mais que negociação de contratos
- Responder a mudanças mais que seguir um plano

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.

### 12 Princípios

Módulo 2: Mentalidade ágil

## Manifesto Ágil

**Indivíduos e interação entre eles** **mais que processos e ferramentas**

Sem um time forte e relacionamentos saudáveis, projetos de desenvolvimento tendem a falhar

- A maioria das metodologias tradicionais estão focadas em processos e ferramentas.
- Não podemos realizar projetos como se pudessem ser realizados por uma máquina, por causa de todas as incertezas envolvidas e a criatividade necessária.
- A melhor maneira de melhorar o desempenho é concentrar-se sobre os aspectos humanos.

**TI.exames** ® Todos os direitos reservados. Proibida a redistribuição deste material. > Slide 29

Módulo 2: Mentalidade ágil

## Manifesto Ágil

**Software em funcionamento** **mais que documentação abrangente**

Um documento expressa o que é pretendido mas nada é mais importante que código pronto

- Por que criar uma documentação completa para começar um projeto?
- Para nos comunicarmos com o cliente e para criarmos uma base para a solução que supostamente deve atender as necessidades do cliente.
- No entanto, o fato é que usuários não sabem o que querem, até que vejam o software funcionando, e nenhuma documentação pode substituir software funcionando para este fim.

**TI.exames** ® Todos os direitos reservados. Proibida a redistribuição deste material. > Slide 30

Módulo 2: Mentalidade ágil

## Manifesto Ágil

The diagram shows a balance scale. On the left side, there is a green upward-pointing arrow above the text "Colaboração com o cliente". On the right side, there is a yellow downward-pointing arrow above the text "mais que negociação de contratos". The center of the scale is a dark blue triangle pointing downwards. Below the scale, a yellow box contains the text: "É melhor trabalhar em uma boa solução com o cliente, do que entrar em conflito".

- Em um ambiente adaptativo não tem sentido sem adaptação, e a base para a adaptação é a colaboração com o cliente.
- O modelo ágil não se baseia no detalhamento inicial total.
- Alterações, ainda que no final de um projeto, não criam grandes quantidades de retrabalho, porque não há um projeto inicial detalhado que necessita ser revisado a cada alteração.

**TI.exames** ® Todos os direitos reservados. Proibida a redistribuição deste material. > Slide 31

Módulo 2: Mentalidade ágil

## Manifesto Ágil

The diagram shows a balance scale. On the left side, there is a green upward-pointing arrow above the text "Responder a mudanças". On the right side, there is a yellow downward-pointing arrow above the text "mais que seguir um plano". The center of the scale is a dark blue triangle pointing downwards. Below the scale, a yellow box contains the text: "As mudanças fazem parte da natureza de um projeto de software".

- Não planejamos todo o projeto de forma antecipada, porque sabemos que clientes e usuários não sabem o que eles querem de forma total e detalhada logo no inicio do projeto.
- Utiliza-se um ciclo de vida adaptativo, que se baseia na resposta às mudanças.
- Plano é uma foto, planejamento é um ato contínuo.

**TI.exames** ® Todos os direitos reservados. Proibida a redistribuição deste material. > Slide 32

Módulo 2: Mentalidade ágil

## Manifesto Ágil

**Manifesto para o desenvolvimento ágil de software**

Estamos descobrindo maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Através deste trabalho, passamos a valorizar:

- Indivíduos e interação entre eles mais que processos e ferramentas
- Software em funcionamento mais que documentação abrangente
- Colaboração com o cliente mais que negociação de contratos
- Responder a mudanças mais que seguir um plano

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.

**12 Princípios**

Fonte: <http://manifestoagil.com.br/>

**TI.exames** ® Todos os direitos reservados. Proibida a redistribuição deste material. > Slide 33

Módulo 2: Mentalidade ágil

## Princípio 01: Satisfazer o cliente

**01**

Nossa maior prioridade é satisfazer o cliente por meio da entrega adiantada e contínua de software de valor.

- O objetivo é ter clientes satisfeitos, porque eles vão voltar e eles também nos recomendarão a outros clientes potenciais.
- Deve-se entregar aquilo é mais importante para o cliente primeiro, não necessariamente o que é mais fácil.
- Entregar de forma rápida, em pequenas quantidades e de forma contínua.
- Obter feedback e melhorar as próximas entregas (ciclo PDCA).



**TI.exames** ® Todos os direitos reservados. Proibida a redistribuição deste material. > Slide 34

## Princípio 02: Aceitar mudanças

02

Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se ajustam às mudanças para que o cliente possa obter vantagens competitivas.

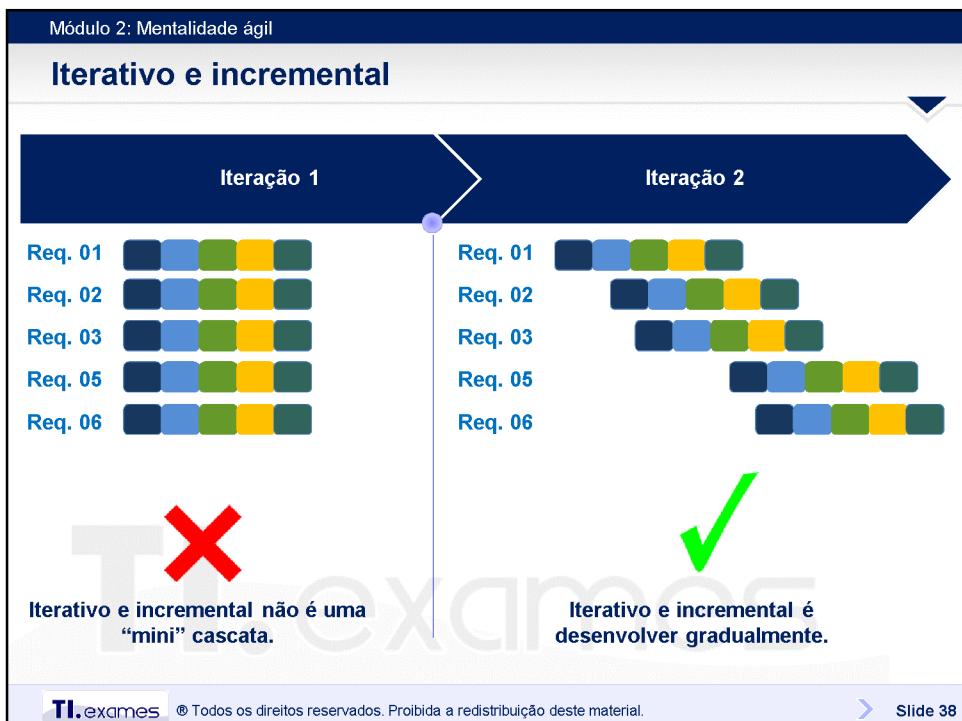
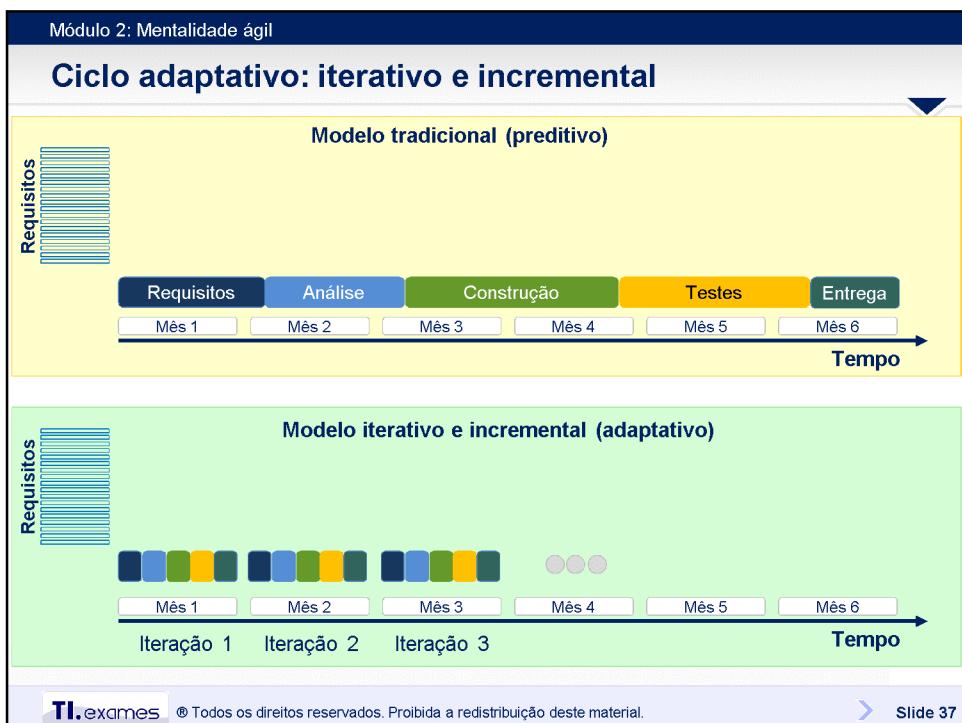
- Usando um ciclo de vida adaptativo, pode-se aceitar mudanças mais facilmente, porque não há nenhum grande detalhamento antecipado que dificulte isso.
- Cada solicitação de mudança é um passo em direção ao que o cliente realmente quer.
- É dada vantagem competitiva ao cliente ao permitir uma resposta rápida às mais recentes mudanças no ambiente externo aproveitando as oportunidades emergentes.



## Ciclo adaptativo: iterativo e incremental

- Para facilitar as mudanças, no desenvolvimento ágil utilizam-se ciclos adaptativos. No nosso contexto isso significa dizer que desenvolvimentos softwares de forma iterativa e incremental.
- O software é construído por partes (incrementos) e para cada parte executa-se um ciclo (iteração), que inclui todas as atividades necessárias para completar as funcionalidades acordadas para o incremento.





## Princípio 03: Entregar software frequentemente

03

Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.

- O cliente terá uma melhor compreensão do que ele quer quando ele vê software funcionando.
- O feedback real recebido deve ser usado para melhorar as próximas entregas.
- Quanto menor o tempo entre os feedbacks, melhor.
- Entregas frequentes fornecem valor de forma antecipada à medida que clientes já podem se beneficiar de funcionalidades importantes.



## Princípio 04: Trabalhar em conjunto

04

Pessoas relacionadas a negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.

- Os princípios ágeis incluem requisitos de manutenção e documentação leve, e reconhecem que a mudança é uma realidade normal e aceitável no desenvolvimento de software.
- Isso faz com que seja necessária uma colaboração estreita entre pessoas de negócio e técnicas.
- É particularmente importante para esclarecer os requisitos *just-in-time* e para manter todos os membros do time na mesma página ao longo do desenvolvimento.



## Princípio 05: Manter times motivados

05

Construir projetos ao redor de indivíduos motivados, dando a eles ambiente e suporte necessário, e confiar que farão seu trabalho.

- Um ambiente Ágil é baseado em um time multifuncional e auto-organizado que se administra e encontra o seu caminho em vez de receber ordens.
- Microgestão e uma abordagem top-down para a gestão são evitadas, pois limitam a velocidade da adaptação necessária.
- Esta é uma grande responsabilidade para os desenvolvedores e nem todos os desenvolvedores são capazes de trabalhar desta forma.
- Quando temos os membros do time preparados, devemos confiar neles, motivá-los e capacitá-los para permitir agilidade.

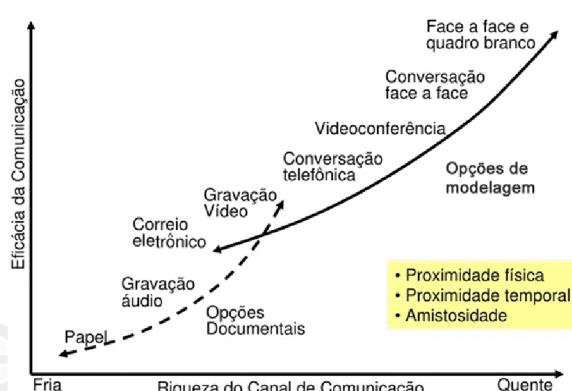


## Princípio 06: Comunicar-se adequadamente

06

O método mais eficiente e eficaz de transmitir informações para e dentro de um time de desenvolvimento é através de conversas cara a cara.

- Nada pode substituir conversas “cara a cara” – são mais rápidas, esclarecedoras e eficazes.
- É uma grande vantagem ter times colocalizados, mas isso não significa que não se pode ter um projeto ágil com um time distribuído.
- Utilizar recursos de videoconferência para minimizar os efeitos



## Princípio 07: Software rodando é a medida de progresso

07

Software funcional é a medida primária de progresso.

- O progresso de 99% ainda é 0% pronto!
- Em desenvolvimento de software está pronto (100%) ou não está (0%).
- Para possibilitar esta forma de gerenciamento, as atividades e entregas devem ser pequenas o suficiente para poderem ser reportadas como “pronta” ou “não pronta”.
- O valor agregado entregue é a medida de progresso.



## Princípio 08: Manter um ambiente sustentável

08

Processos ágeis promovem um ambiente sustentável. Patrocinadores, desenvolvedores e usuários devem ser capazes de manter indefinidamente passos constantes.

- Metodologias ágeis buscam o equilíbrio entre vida pessoal e profissional.
- O objetivo é entregar valor (software) e não “horas”.
- Fazer horas extras pode parecer que faz as coisas andarem mais depressa, mas na realidade isso diminui a produtividade e aumenta os defeitos.
- Fazer horas extras não é proibido, mas altamente desencorajado.



## Princípio 09: Entregar com excelência

09

Continua atenção à excelência técnica e bom design aumenta a agilidade.

- Deve-se dar a atenção à excelência técnica e ao bom design para evitar problemas.
- Mau design reduz a produtividade ao longo do tempo, aumentando o custo para manter o software e reduzindo o valor entregue.
- Não se deve esquecer que a meta é uma solução "suficientemente boa" em vez de perfeita.



## Princípio 10: Pensar e agir de forma simples e priorizada

10

Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.

- O princípio de Pareto (regra 80/20) é aplicado neste momento.
- A ideia é focar nos 20% que trarão 80% dos resultados.
- Focar no que é essencial para a entrega de valor para clientes e usuários, e não em coisas que não agregam valor como componentes e processos, entre outros.
- Entregar uma solução boa é melhor do que não entregar uma solução ótima – “o ótimo é inimigo do bom”.
- Pensar de forma simples, resolvendo o problema de hoje, sem tentar prever o futuro.



## Princípio 11: Requisitos e arquiteturas emergem

11

Os melhores requisitos, arquiteturas e designs emergem de times auto-organizáveis.

- As pessoas geralmente trabalham melhor quando elas são respeitadas e têm a autoridade para decidir como trabalham.
- O desenvolvimento de software é um trabalho realizado em conjunto, no qual a soma dos conhecimentos gera produtos melhores.
- Times auto-organizáveis conseguem unir seus esforços e tirar proveito do conhecimento coletivo – “duas cabeças pensam melhor do que uma”.



## Princípio 12: Melhorar continuamente

12

Em intervalos regulares, o time reflete em como ficar mais eficaz e então se ajusta e otimiza seu comportamento de acordo.

- O modelo Ágil acredita que sempre há espaço para melhorias, não importa o quanto bem as coisas estão sendo feitas.
- Após curtos espaços de tempo, o time reflete sobre como melhorar seu desempenho.
- O objetivo é melhorar a cada iteração, pelo menos um pouco.



Este princípio é chave para o bom funcionamento de um processo empírico.



**Quiz**

*Quiz - 7 questions*

Last Modified: nov 22, 2017 at 07:24 PM

**PROPERTIES**

On passing, 'Finish' button: [Goes to Next Slide](#)

On failing, 'Finish' button: [Goes to Next Slide](#)

Allow user to leave quiz: [At any time](#)

User may view slides after quiz: [At any time](#)

Show in menu as: [Multiple items](#)

 Edit in Quizmaker     Edit Properties

Módulo 2: Mentalidade ágil

## Abordagem tradicional x agilidade: principais diferenças

Abordagem tradicional	Abordagem ágil
O foco do gerente de projetos está nos recursos e no gerenciamento do projeto.	Times são auto-organizados.
Requisitos são “escritos na pedra” no início do projeto.	Requisitos mudam constantemente.
Mudanças no projeto são prevenidas.	Mudanças no projeto são bem-vindas e aceitas.
Requisitos são priorizados pela dependência entre os mesmos.	Requisitos são priorizados pelo valor que trazem ao negócio.
O status do projeto é comumente medido pelo % de conclusão das fases ou marcos do projeto.	O projeto é medido em termos de funcionalidades que já foram liberadas.
Riscos normalmente não são gerenciados diariamente.	Riscos são gerenciados diariamente e por todo o time.
O cliente precisa esperar muito tempo para ver o software funcionando.	Logo no início do projeto o cliente já pode ver partes do software funcionando.

**TI.exames** ® Todos os direitos reservados. Proibida a redistribuição deste material. Slide 50

## Quiz final

Recomendamos agora completar o quiz final deste módulo. Clique no botão abaixo para abrir a página do quiz.

**CLIQUE PARA REALIZAR O QUIZ FINAL**

O botão acima não abriu a página do quiz?

Caso o botão acima não abra a página do quiz no seu navegador, então acesse o link "Realizar quiz" disponível na lista de módulos gravados dentro da página do curso no seu ambiente de ensino.

 [Realizar quiz](#)  [Baixar Pdf](#)

**TI.exames** ® Todos os direitos reservados. Proibida a redistribuição deste material.  Slide 51

## Fim do módulo

Pronto, você finalizou este módulo. Leia as instruções abaixo:

- Recomendamos neste momento fazer uma revisão dos slides para confirmar o entendimento de tudo o que foi apresentado neste módulo.
- Você pode clicar no (X) da janela para fechar este módulo.
- A sua nota obtida no teste do slide anterior será exibida na lista dos módulos.

 [Última nota quiz](#)  [Ver aula](#)  [Baixar Pdf](#)

**TI.exames** ® Todos os direitos reservados. Proibida a redistribuição deste material.  Slide 52