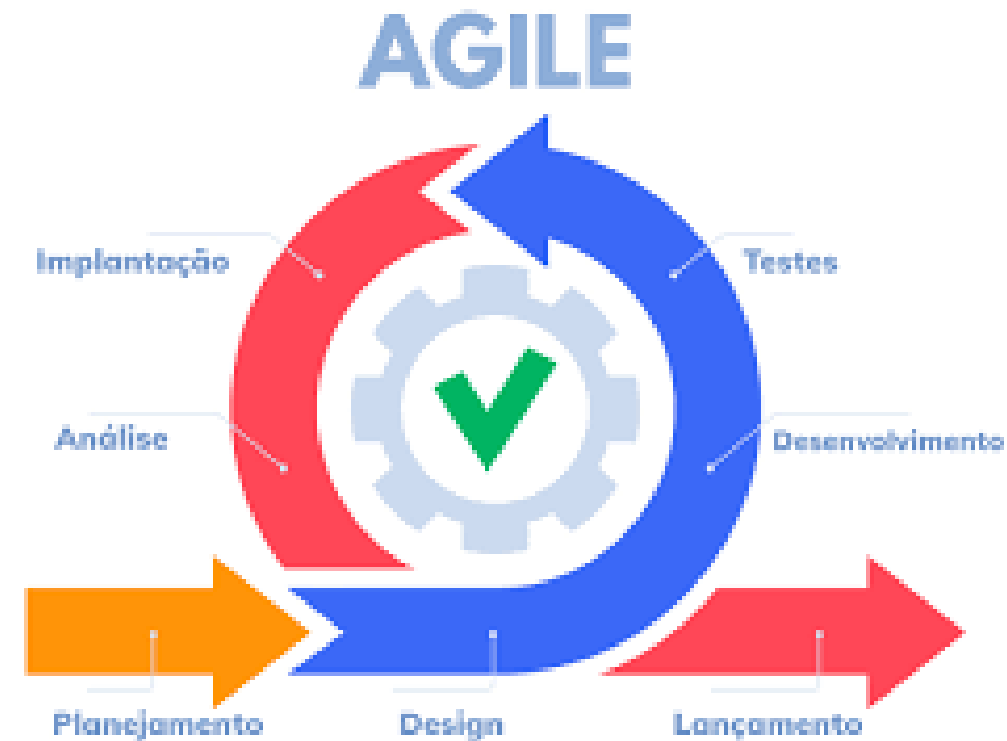


# Engenharia de Software I

Prof. Agnaldo Cieslak, MsC

# Desenvolvimento Ágil de Software



# Uma Introdução a Métodos Ágeis

---

- Atualmente as empresas operam em um ambiente global, com **mudanças rápidas**.
- Desta forma precisam **responder** a novas **oportunidades** e **novos mercados**, a **mudanças** nas condições econômicas e ao **surgimento** de **produtos** e **serviços concorrentes**.
- Softwares fazem parte de quase todas as operações de negócios, assim, **novos softwares** precisam **desenvolvidos rapidamente**
- E, por **operarem** em um **ambiente** de **mudanças rápidas**, muitas vezes é praticamente **impossível** obter um conjunto **completo de requisitos** de software estável.

➤ **Processos tradicionais** de desenvolvimento de software que planejam especificar completamente os requisitos e, em seguida, projetar, construir e testar o sistema **não estão adaptados ao desenvolvimento rápido de software**.

# Uma Introdução a Métodos Ágeis

---

- Os processos de **desenvolvimento rápido de software** são concebidos para **produzir, rapidamente, softwares** úteis.

## Características importantes

- Os processos de especificação, projeto e implementação são intercalados.
- O sistema é desenvolvido em uma série de versões.
- Interfaces de usuário do sistema são geralmente desenvolvidas com um sistema interativo de desenvolvimento que permite a criação rápida do projeto de interface por meio de desenho e posicionamento de ícones na interface.

# Manifesto Ágil

---

- A filosofia por trás dos métodos ágeis é refletida no **manifesto ágil**, que foi acordado por muitos dos principais desenvolvedores desses métodos. Esse manifesto afirma:

*Estamos descobrindo melhores maneiras de desenvolver softwares, fazendo-o e ajudando outros a fazê-lo. Através desse trabalho, valorizamos mais:*

- *Indivíduos e interações* do que *processos e ferramentas*
- *Software em funcionamento* do que *documentação abrangente*
- *Colaboração do cliente* do que *negociação de contrato*
- *Respostas a mudanças* do que *seguir um plano*

*Ou seja, embora itens à **em azul** sejam importantes, valorizamos mais os que os itens em **vermelho**.*

# Manifesto Ágil

## Os 12 Princípios

---

### 1. Geração de Valor

- Nossa maior **prioridade é satisfazer ao cliente** com **entrega contínua e adiantada** de software com valor agregado.

### 2. Flexibilidade:

- **Mudanças de requisitos são bem-vindas**, mesmo tardiamente no desenvolvimento. Os processos ágeis tiram vantagem das mudanças, visando à **vantagem competitiva** para o cliente.

### 3. Frequência

- **Entregar software funcionando frequentemente**, de poucas semanas a poucos meses, com preferência para a escala menor de tempo.

### 4. União:

- **Pessoa de negócios e desenvolvedores** devem **trabalhar** diariamente em **conjunto** por todo o projeto.

### 5. Motivação

- Construir projetos em torno de **indivíduos motivados**, dando a eles o **ambiente** e o **suporte** necessário e **confiando** neles para realizar o trabalho.

# Manifesto Ágil

## Os 12 Princípios

---

### 6. Comunicação

- O **método** mais eficiente e eficaz de **transmitir informações** para a equipe e entre a equipe de desenvolvimento é a **conversa frente a frente**.

### 7. Funcionalidade

- **Software funcional** é a **medida primária** de progresso

### 8. Sustentabilidade

- Processos ágeis **promovem um desenvolvimento sustentável**. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.

### 9. Revisão

- Contínua atenção à **excelência técnica** e bom projeto **aumenta a agilidade**.

### 10. Simplicidade

- a arte de maximizar a quantidade de trabalho não realizado – é essencial.

### 11. Organização

- As **melhores arquiteturas**, os **melhores requisitos** e **projetos** emergem de **times auto-organizáveis**.

### 12. Autoavaliação

- Em intervalos regulares, o **time reflete** sobre como pode **ser mais eficaz**, então **refina e ajusta seu comportamento** de acordo.

# Métodos Ágeis

	TRADICIONAL	METODOLOGIAS ÁGEIS
Pressupostos fundamentais	Sistemas totalmente especificáveis, previsíveis; desenvolvidos a partir de um planejamento extensivo e meticuloso	Software adaptativo e de alta qualidade; pode ser desenvolvido por equipes pequenas utilizando os princípios da melhoria contínua do projeto e testes orientados a rápida resposta a mudanças
Controle	Orientado a processos	Orientado a pessoas
Estilo de gerenciamento	Comandar e controlar	Liderar e colaborar
Gestão do conhecimento	Explícito	Tácito
Atribuição de papéis	Individual – favorece a especialização	Times auto-organizáveis – favorece a troca de papéis
Comunicação	Formal	Informal
Ciclo do projeto	Guiado por tarefas ou atividades	Guiado por funcionalidades do produto
Modelo de desenvolvimento	Modelo de ciclo de vida (Cascata, Espiral, ou alguma variação)	Modelo iterativo e incremental de entregas
Forma/estrutura organizacional desejada	Mecânica (burocrática com muita formalização)	Orgânica (flexível e com incentivos a participação e cooperação social)



# Métodos Ágeis

---

- **Algumas dúvidas e considerações:**

1. É possível fazer **manutenção** dos sistemas desenvolvidos em uma **abordagem ágil**, dada a ênfase do processo de desenvolvimento em **minimização** da **documentação** formal?
  - Supostamente, a **documentação formal deve descrever o sistema** e, assim, tornar sua compreensão mais fácil para as pessoas que fazem as mudanças.
  - Porém, **na prática**, a documentação formal **nem sempre é atualizada**, e, portanto, **não reflete exatamente o código do programa**.

Práticas ágeis, portanto, enfatizam a importância de se escrever **códigos bem-estruturados** e investir na **melhoria do código**.

# Métodos Ágeis

---

- **Algumas dúvidas e considerações:**

2. Os métodos ágeis podem, efetivamente, ser usados para a **evolução** de um sistema em resposta às **solicitações de mudança do cliente**?
  - Envolvimento de um representante em tempo integral é menos provável durante a manutenção

O documento-chave é o documento de **requisitos do sistema**, que informa ao analista de sistemas ou engenheiro de software o que o sistema deve fazer. Sem esse conhecimento, é difícil **avaliar o impacto das mudanças** propostas

➤ A principal **dificuldade** após a entrega do software é **manter o envolvimento dos clientes** no processo