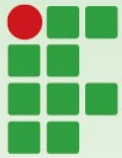


**INSTITUTO FEDERAL**  
Santa Catarina



# Desenvolvimento Ágil de Software

Prof. Edjandir Corrêa Costa  
[edjandir.costa@ifsc.edu.br](mailto:edjandir.costa@ifsc.edu.br)

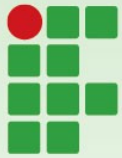


**INSTITUTO FEDERAL**  
Santa Catarina



# Métodos Ágeis – História

- Na início da década de 90 havia uma visão de que a melhor maneira para se criar software era por meio de um planejamento cuidadoso, formalidades e projeto apoiado por CASE;
- Essa visão veio da comunidade da Engenharia de Software, responsável por sistemas de softwares grandes, como sistemas aeroespaciais e do governo.

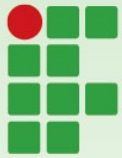


**INSTITUTO FEDERAL**  
Santa Catarina



# Métodos Ágeis - História

- O *overhead* de planejamento estava se tornando tão pesado que estava dominando o processo de desenvolvimento. Gastava-se mais tempo em análises do que na construção do programa e testes.
- A insatisfação levou ao surgimento, no final da década de 90, dos novos 'métodos ágeis'.



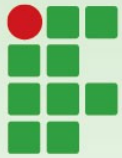
**INSTITUTO FEDERAL**  
Santa Catarina



# Métodos Ágeis - Manifesto

“Estamos descobrindo melhores maneiras de desenvolver softwares, fazendo-o e ajudando outros a fazê-lo. Através deste 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. “



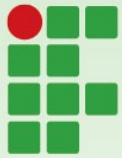
**INSTITUTO FEDERAL**  
Santa Catarina



# Métodos Ágeis – Mais conhecidos

- XP – Extreme Programming
- SCRUM
- Crystal
- DSDM
- FDD



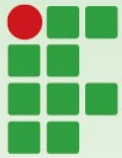


**INSTITUTO FEDERAL**  
Santa Catarina



# Métodos Ágeis – Princípios

- Envolvimento do cliente
- Entrega Incremental
- Pessoas, não processos
- Aceitar as mudanças
- Manter a simplicidade

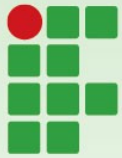


**INSTITUTO FEDERAL**  
Santa Catarina



# Métodos Ágeis – Indicados

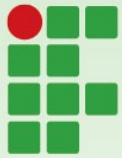
- Desenvolvimento de produtos pequenos ou médios;
- Sistemas personalizados dentro de uma organização , em que existe um compromisso claro do cliente de se envolver, com poucas regras ou regulamentos externos que afetam o software.



# Métodos Ágeis – Dificuldades

- Sucesso depende de um cliente disposto e capaz de passar o tempo com a equipe de desenvolvimento;
- Alguns membros podem ter dificuldades em interagir com outros da equipe;
- Priorizar mudanças é difícil quando existem muitos *stakeholders*;
- Manter a simplicidade exige trabalho extra;
- Migrar de processos bem definidos para um processo mais informal pode ser difícil.



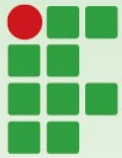


**INSTITUTO FEDERAL**  
Santa Catarina



# Métodos Ágeis – Quando usar

- Na verdade, a maioria dos projetos de software inclui práticas das abordagens dirigidas a planos e ágeis:
  - Se é importante ter um manter uma especificação: usar abordagem dirigida a planos;
  - Se o feedback do cliente for rápido: métodos ágeis;
  - O uso de métodos ágeis demanda boas ferramentas, bons projetistas e programadores.

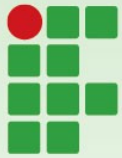


**INSTITUTO FEDERAL**  
Santa Catarina



# XP – Extreme Programming

- Criado em 2000 por Kent Beck, em XP os requisitos são expressos por histórias de usuário. Quando o código novo é integrado ao sistema, todos os testes devem ser executados com sucesso.

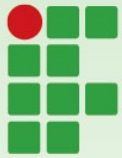


INSTITUTO FEDERAL  
Santa Catarina



# XP – Extreme Programming

- Princípios:
  - O desenvolvimento incremental é sustentado por meio de pequenos e frequentes releases;
  - Engajamento contínuo do cliente;
  - Pessoas e não processos sustentado pela programação em pares;
  - Mudanças são aceitas por meios de *releases* contínuos para clientes, *test-first* e integração contínua;
  - Simplicidade é feita por meio da refatoração constante.

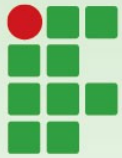


INSTITUTO FEDERAL  
Santa Catarina



# XP – Extreme Programming

- Práticas:
  - Planejamento incremental;
  - Pequenos *releases*;
  - Projeto simples;
  - Test-first;
  - Refatoração;
  - Programação em pares;



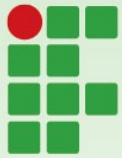
**INSTITUTO FEDERAL**  
Santa Catarina



# XP – Extreme Programming

- Práticas (continuação):
  - Propriedade coletiva;
  - Integração contínua;
  - Ritmo sustentável;
  - Cliente no local;



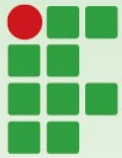


**INSTITUTO FEDERAL**  
Santa Catarina



# XP – Extreme Programming

- Em XP os requisitos são expressos em histórias de usuário. Cada história é escrita pelos desenvolvedores e cliente em um cartão. Ela deve prover o entendimento do que deve ser feito além de facilitar a estimativa de velocidade da equipe. A história deve ser focada nas necessidades dos usuários.



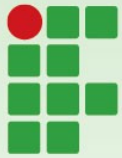
**INSTITUTO FEDERAL**  
Santa Catarina



# XP – Extreme Programming

- Estória de usuário:

Kate é uma médica que deseja prescrever medicamentos para um paciente de uma clínica. O prontuário do paciente já está sendo exibido em seu computador, assim, ela clica no campo “medicação” e pode selecionar “medicação atual”, “nova medicação” ou “formulário”. Se ela selecionar “medicação atual”, o sistema pede que ela verifique a dose. Se ela quiser mudar a dose, ela altera esta e em seguida, confirma a prescrição. Se ela escolher “nova medicação”, o sistema assume que ela sabe qual medicação receitar. Ela digita as primeiras letras do nome do medicamento. O sistema exibe uma lista de possíveis fármacos que começam com essas letras. Ela escolhe a medicação requerida e o sistema responde, pedindo-lhe para verificar se o medicamento selecionado está correto. Ela insere a dose e, em seguida, confirma a prescrição.

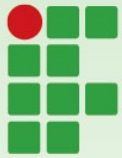


**INSTITUTO FEDERAL**  
Santa Catarina



# XP – Extreme Programming

- Os cartões são as principais entradas para o processo de planejamento no XP ou o “jogo de planejamento”. Uma vez que tenham sido desenvolvidos, a equipe os divide em tarefas e estima o esforço e os recursos necessários para a realização de cada tarefa.

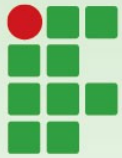


**INSTITUTO FEDERAL**  
Santa Catarina



# XP – Extreme Programming

- Tarefa 1: Alterar dose de medicamentos prescritos
- Tarefa 2: Seleção de formulário
- Tarefa 3: Verificação de dose
- .....



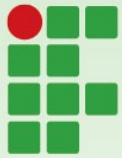
**INSTITUTO FEDERAL**  
Santa Catarina



# XP – Extreme Programming

- O Cliente é quem prioriza as histórias para implementação. Assim como os requisitos mudam, as histórias podem mudar ou até mesmo serem descartadas.



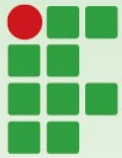


**INSTITUTO FEDERAL**  
Santa Catarina



# XP – Extreme Programming

- A abordagem XP enfatiza a importância dos testes do programa. Extreme Programming inclui uma abordagem de testes que reduz as chances de erros desconhecidos na versão atual do sistemas

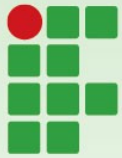


**INSTITUTO FEDERAL**  
Santa Catarina



# XP – Extreme Programming

- As principais características dos testes em XP são:
  - Desenvolvimento test-first;
  - Desenvolvimento de teste incremental a partir de cenários;
  - Envolvimento dos usuários no desenvolvimento de testes e validação;
  - Uso de frameworks de testes automatizados.

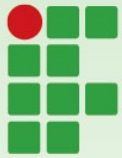


**INSTITUTO FEDERAL**  
Santa Catarina



# XP – Extreme Programming

- O desenvolvimento test-first ou TDD é uma das principais inovações do XP. Em vez de escrever algum código e, em seguida escrever os testes, você escreve os testes antes de escrever o código. Você pode rodar os testes sempre que incrementar o código da classe.

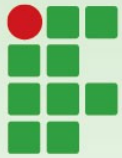


**INSTITUTO FEDERAL**  
Santa Catarina



# XP – Extreme Programming

- Programação em pares, outra técnica inovadora do XP, consiste em dois programadores escreverem juntos uma mesma funcionalidade, lado a lado, na mesma estação de trabalho. Os pares são criados de forma dinâmica, de modo que todos os membros da equipe trabalhem uns com os outros.



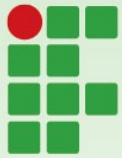
**INSTITUTO FEDERAL**  
Santa Catarina



# XP – Extreme Programming

- Vantagens da programação em pares:
  - Dá suporte à idéia de propriedade e responsabilidade coletiva para o sistema;
  - Atua como um processo de revisão informal, pois cada linha de código é observada por, pelo menos, duas pessoas.
  - Dá suporte à refatoração, que é um processo de melhoria de software.



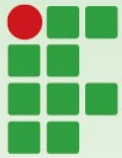


**INSTITUTO FEDERAL**  
Santa Catarina



# Gerenciamento ágil de projetos

- A principal responsabilidade dos gerentes de projeto de software é gerenciar o projeto para que o software seja entregue no prazo e dentro do orçamento. A abordagem padrão para gerenciamento de projetos é dirigida a planos. O desenvolvimento ágil requer uma abordagem diferente, adaptada para o desenvolvimento incremental e para o melhor aproveitamento da equipe.

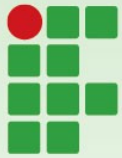


**INSTITUTO FEDERAL**  
Santa Catarina



# SCRUM

- O método SCRUM é um método ágil geral, mas o seu foco é no gerenciamento do desenvolvimento iterativo. Pode ser usado também em combinação com as práticas do XP.

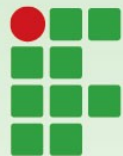


**INSTITUTO FEDERAL**  
Santa Catarina



# SCRUM

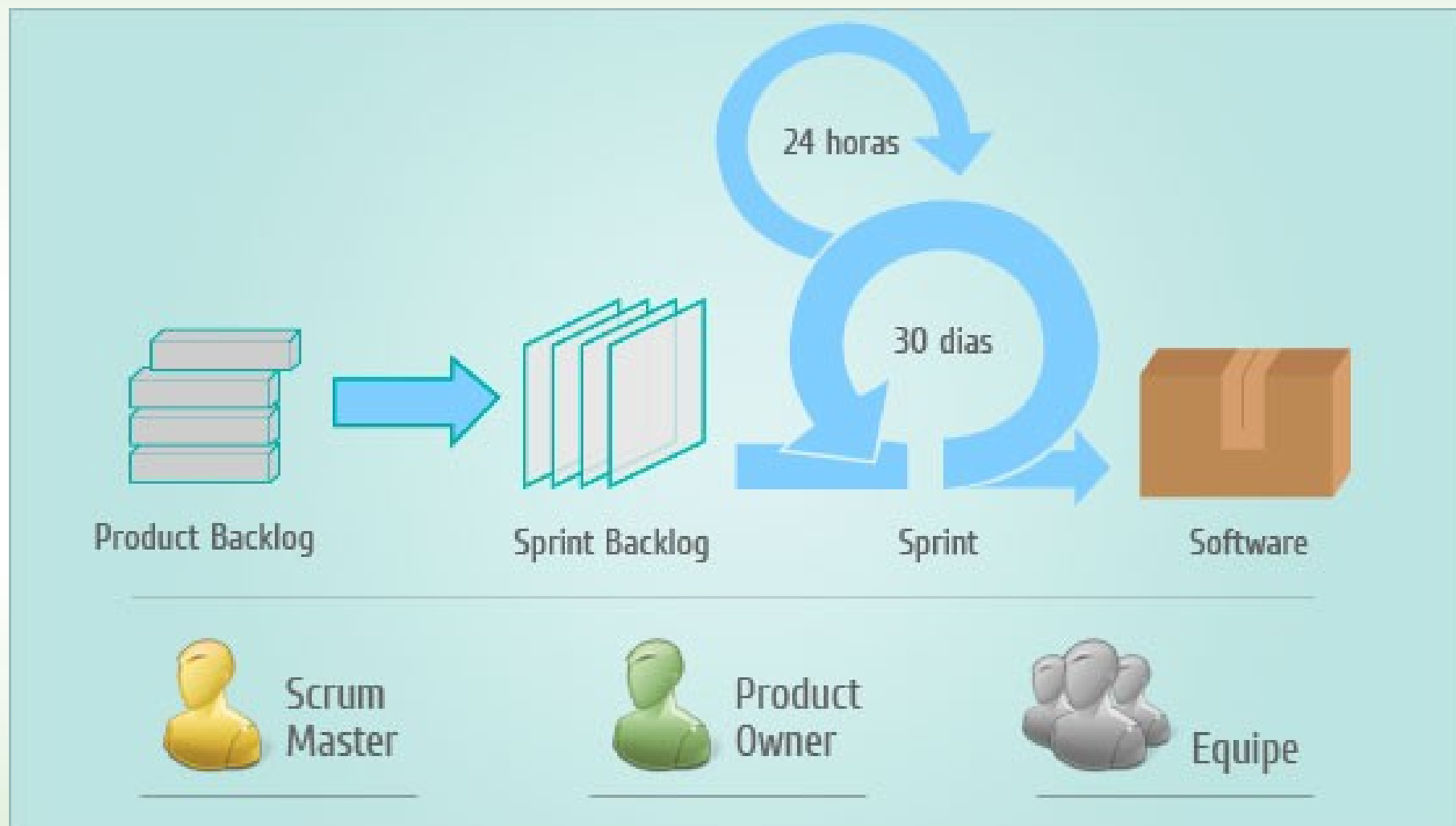
- No SCRUM existem três fases:
  - A primeira é de um planejamento geral, onde se definem os objetivos gerais do projeto;
  - Em seguida, ocorre uma série de ciclos de *sprint*, sendo que cada ciclo desenvolve um incremento do sistema;
  - A última fase encerra o projeto, completa documentação exigida, manuais de ajuda e avalia as lições aprendidas

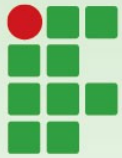


**INSTITUTO FEDERAL**  
Santa Catarina



# SCRUM





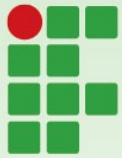
INSTITUTO FEDERAL  
Santa Catarina



# SCRUM

- Na fase central do SCRUM:
  - Sprints representam os ciclos que tem como resultado um incremento do sistema. Em média duram de 2 a 4 semanas;
  - O ponto de partida é o *backlog* do produto que, a medida que as sprints vão acontecendo, vai diminuindo;
  - A fase de seleção envolve todos da equipe do projeto;
  - Uma vez que todos estejam de acordo a equipe se organiza para desenvolver o ciclo.





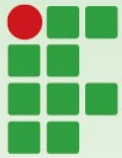
**INSTITUTO FEDERAL**  
Santa Catarina



# SCRUM

- Papéis do SCRUM:

- SCRUM Master: tem o papel de proteger a equipe de distrações externas, além de atuar com um facilitador e apoiador na execução das rotinas do SCRUM;
- Product Owner: representa o cliente na equipe. É o responsável por garantir que o resultado das sprints atendam as necessidades do cliente;
- Equipe: “auto-gerenciável”, todos têm a visão do que é importante para o cliente e o comprometimento para atendê-lo.

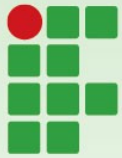


**INSTITUTO FEDERAL**  
Santa Catarina



# SCRUM

- **Eventos do SCRUM:**
  - Planejamento da Sprint: são definidos as histórias que serão atendidas pela sprint;
  - Reunião diária: acontece todos os dias e tem a duração média de 15 minutos. Participam todos os integrantes da equipe, product owner e scrum master. Em cada sessão apresenta-se o resultado do trabalho do dia anterior, os problemas que aconteceram e o que será realizado no dia;
  - Revisão de sprint: se apresenta o resultado da sprint e se discute o que deve ser melhorado para a execução da próxima sprint



**INSTITUTO FEDERAL**  
Santa Catarina



# Bibliografia

**SOMMERVILLE, I.; Engenharia de Software. 9<sup>a</sup>.  
Edição. São Paulo: Pearson, 2011.**