

Modelos Ágeis:

XP

SCRUM

Engenharia de Software I - Profa. Priscila Faccioli

Tópicos

- 1- O que é um Processo Ágil?
- 2- XP – Extreme Programming;
- 3- SCRUM;
- 4- FDD – Feature Driven Development;
- 5- ASD- Adaptive Software Development;
- 6- DSDM- Dynamic Systems Development;
- 7- Crystal
- 8- Modelagem Ágil

O que é um Processo Ágil?

- **As metodologias ágeis são adequadas para situações em que a mudança de requisitos é freqüente. Para ser realmente considerada ágil, a metodologia deve aceitar a mudança em vez de tentar prever o futuro.**

Manifesto Ágil

- Reunião entre 17 gurus da comunidade de desenvolvimento;
- Realizada entre os dias 11 e 13 de fevereiro de 2001;
- Estação de esqui nas montanhas de Utah, Estados Unidos.

Manifesto Ágil- Princípios

- **Indivíduos e interações em vez de processos e ferramentas;**
- **Software executável em vez de documentação;**
- **Colaboração do cliente ao invés de negociação de contratos;**
- **Respostas rápidas a mudanças em vez de seguir planos.**

Manifesto Ágil- Princípios

O Manifesto Ágil não rejeita processos e ferramentas, documentação, negociação de contratos nem planejamento, mas simplesmente mostra que estes têm importância secundária quando comparados com os indivíduos, com o software executável, com a colaboração dos clientes e as respostas rápidas às mudanças. Esses conceitos aproximam-se melhor da forma como as pequenas e médias empresas trabalham e respondem às mudanças.

XP – Extreme Programming

XP – Extreme Programming

O *Extreme Programming* é um modelo de desenvolvimento de software, criado em 1996, por Kent Beck, no Departamento de Computação da montadora de carros Daimler Chrysler, ele possui muitas diferenças em relação a outros modelos, podendo ser aplicado a projetos de alto risco e com requisitos dinâmicos.

XP – Valores

A XP enfatiza o desenvolvimento rápido do projeto e visa a garantir a satisfação do cliente, além de favorecer o cumprimento das estimativas. As práticas e valores da XP proporcionam um agradável ambiente de desenvolvimento de software para os seus seguidores, que são conduzidos por quatro valores: **Comunicação, Simplicidade, Feedback e Coragem.**

XP – Valores

Comunicação: Manter o melhor relacionamento possível entre clientes e desenvolvedores, preferindo conversas pessoais a outros meios de comunicação.

Simplicidade: Permitir a criação de código enxuto que não deve possuir funções desnecessárias. Também evita-se adicionar funcionalidades que talvez só venham a ser importantes no futuro.

XP – Valores

Feedback: O programador terá informações constantes sobre o código e o cliente. A informação do código é dada pelos testes constantes, que indicam os erros tanto individuais quanto do software integrado. Em relação ao cliente, o feedback constante significa que ele terá freqüentemente uma parte do software totalmente funcional para avaliar. Eventuais erros e não-conformidades serão rapidamente identificados e corrigidos nas versões seguintes. Desta forma, a tendência é que o produto final esteja de acordo com as expectativas reais do cliente.

XP – Valores

Coragem: É necessário coragem para implantar os três valores anteriores. Por exemplo, não são todas as pessoas que possuem facilidade de comunicação e têm bom relacionamento. A coragem também dá suporte à simplicidade, pois assim que a oportunidade de simplificar o software é percebida, a equipe pode experimentá-la. Além disso, é preciso coragem para obter feedback constante do cliente. Por exemplo, ao cliente testar uma parte executável constantemente, o cliente poderá querer aumentar as funcionalidades a serem implementadas, ocasionando alterações no escopo inicial do software.

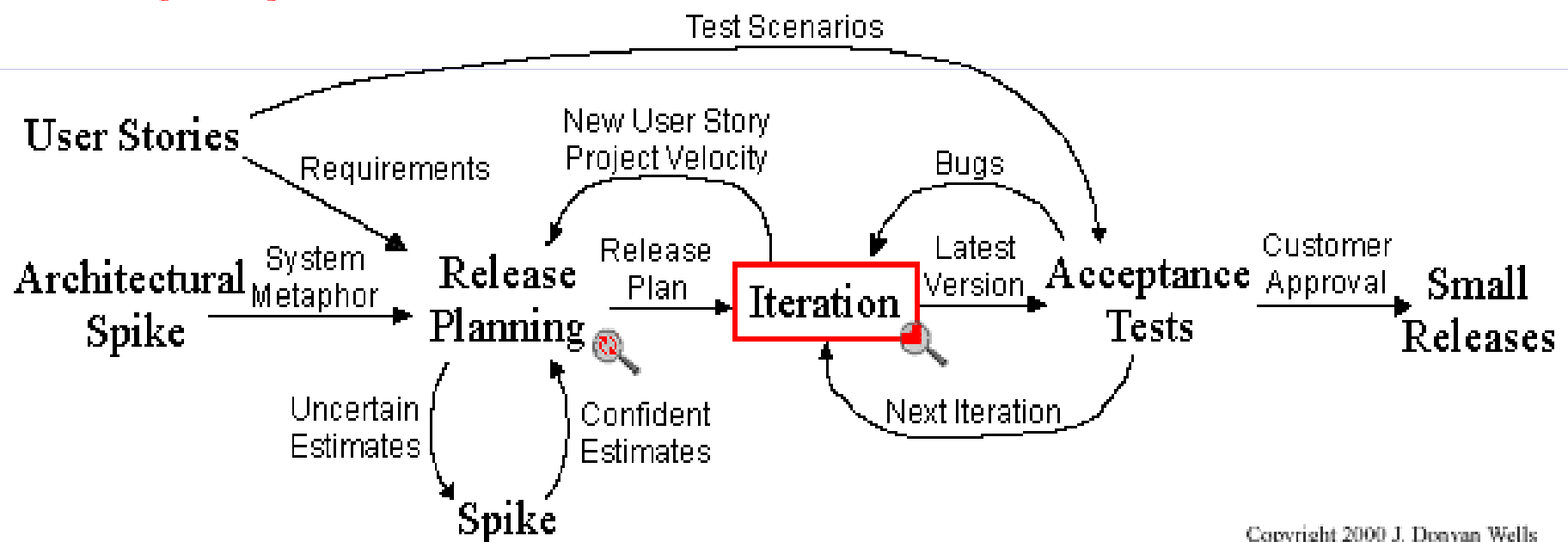
XP – Valores

Coragem: É necessário coragem para implantar os três valores anteriores. Por exemplo, não são todas as pessoas que possuem facilidade de comunicação e têm bom relacionamento. A coragem também dá suporte à simplicidade, pois assim que a oportunidade de simplificar o software é percebida, a equipe pode experimentá-la. Além disso, é preciso coragem para obter feedback constante do cliente. Por exemplo, ao cliente testar uma parte executável constantemente, o cliente poderá querer aumentar as funcionalidades a serem implementadas, ocasionando alterações no escopo inicial do software.

XP – Ciclo de Vida



Extreme Programming Project



XP – Práticas

- **1- Jogo de Planejamento:** O desenvolvimento é feito em iterações semanais. No início da semana, desenvolvedores e cliente reúnem-se para priorizar as funcionalidades. Essa reunião recebe o nome de Jogo do Planejamento. Nela, o cliente identifica prioridades e os desenvolvedores as estimam. Como o escopo é reavaliado semanalmente, o projeto é regido por um contrato de escopo negociável, que difere significativamente das formas tradicionais de contratação de projetos de software. Ao final, o cliente recebe novas funcionalidades prontas para serem colocadas em produção.

XP – Práticas

- **2- Pequenas Versões:** A liberação de pequenas versões funcionais do projeto auxilia muito no processo de aceitação por parte do cliente, que já pode testar uma parte do sistema que está comprando. As versões chegam a ser ainda menores que as produzidas por outras metodologias incrementais, como o RUP.
- **3- Metáfora:** Procurar facilitar a comunicação com o cliente, entendendo a realidade dele. O conceito de rápido para um cliente de um sistema jurídico é diferente para um programador experiente em controlar comunicação em sistemas de tempo real, como controle de tráfego aéreo. É preciso traduzir as palavras do cliente para o significado que ele espera dentro do projeto.

XP – Práticas

4- Projeto Simples: Projeto simples significa dizer que caso o cliente tenha pedido que na primeira versão apenas o usuário "teste" possa entrar no sistema com a senha "123" e assim ter acesso a todo o sistema, você vai fazer o código exato para que esta funcionalidade seja implementada, sem se preocupar com sistemas de autenticação e restrições de acesso.

5- Time Coeso (*Whole Team*): A equipe de desenvolvimento é formada pelo cliente e pela equipe de desenvolvimento (de 1 a 10 integrantes).

XP – Práticas

- **7- Testes de Aceitação:** São testes construídos pelo cliente em conjunto com analistas e testadores, para aceitar um determinado requisito do sistema.
- **8- Ritmo Sustentável:** Trabalhar com qualidade, buscando ter ritmo de trabalho saudável (40 horas/semana, 8 horas/dia), sem horas extras. Horas extras são permitidas quando trouxerem produtividade para a execução do projeto.
- **9- Reuniões em pé :** Reuniões em pé para não se perder o foco nos assuntos de modo a efetuar reuniões rápidas, apenas abordando tarefas realizadas e tarefas a realizar pela equipe.

XP – Práticas

- 10- Posse Coletiva: O objetivo com isto é fazer a equipe conhecer todas as partes do sistema.
- 11- Programação em Pares: é a programação em par/dupla num único computador. Geralmente a dupla é criada com alguém sendo iniciado na linguagem e a outra pessoa funcionando como um instrutor. Como é apenas um computador, o novato é que fica à frente fazendo a codificação, e o instrutor acompanha ajudando a desenvolver suas habilidades. Dessa forma o programa sempre é revisto por duas pessoas, evitando e diminuindo assim a possibilidade de erros (bugs). Com isto, procura-se sempre a evolução da equipe, melhorando a qualidade do código fonte gerado.

XP – Práticas

- **12- Padrões de Codificação** : A equipe de desenvolvimento precisa estabelecer regras para programar e todos devem seguir estas regras.
- **13- Desenvolvimento Orientado a Testes**: Primeiro crie os testes unitários e depois crie o código para que os testes funcionem. Esta abordagem é complexa no início, pois vai contra o processo de desenvolvimento de muitos anos. Só que os testes unitários são essenciais para que a qualidade do projeto seja mantida.

XP – Práticas


- **14- Refatoração:** A refatoração melhora a clareza (leitura) do código, dividido em módulos mais coesos e de maior reaproveitamento, evitando a duplicação de código-fonte.
- **15- Integração Contínua:** Sempre que realizar uma nova funcionalidade, nunca esperar uma semana para integrar na versão atual do sistema. Isto só aumenta a possibilidade de conflitos e a possibilidade de erros no código fonte. Integrar de forma contínua permite saber o status real da programação.

XP – Vantagens

- Torna o processo de desenvolvimento de SW mais ágil e flexível;
- Análise prévia de tudo que pode acontecer durante o desenvolvimento do projeto, oferecendo qualidade, confiança, data de entregas e custos promissores;
- O XP é ideal para ser usada em projetos em que o cliente não sabe exatamente o que deseja e pode mudar muito de opinião durante o desenvolvimento do projeto. Com feedback constante, é possível adaptar rapidamente eventuais mudanças nos requisitos.
- Entregas constantes de partes operacionais do software. Desta forma, o cliente não precisa esperar muito para ver o software funcionando, como nas metodologias tradicionais.

XP – Desvantagens

- Não existe uma avaliação de riscos, devendo, portanto implementar uma análise e estratégias que buscam diminuir os erros.
- A análise de requisitos é informal e com isso pode não ser bem vista pelos clientes, que poderão se sentir inseguros quanto ao bom funcionamento do sistema.
- Refatoração do código pode ser vista como irresponsabilidade e incompetência, pois, não existe uma preocupação formal na utilização do código.
- A falta de documentação é característica do processo XP, pois, o mesmo não dá muita ênfase em burocracias (documentos, formulários, processos, controles rígidos, etc.).



SCRUM

O QUE É O SCRUM?

- O Scrum não é considerado uma metodologia e sim um processo ou ainda um framework para gerenciamento de projetos ágeis.
- É baseado em ciclos de 30 dias, onde se trabalha para alcançar objetivos bem definidos. Estes objetivos são representados através de uma lista que é constantemente atualizada e definindo suas prioridades.

QUANDO SURTIU O SCRUM?

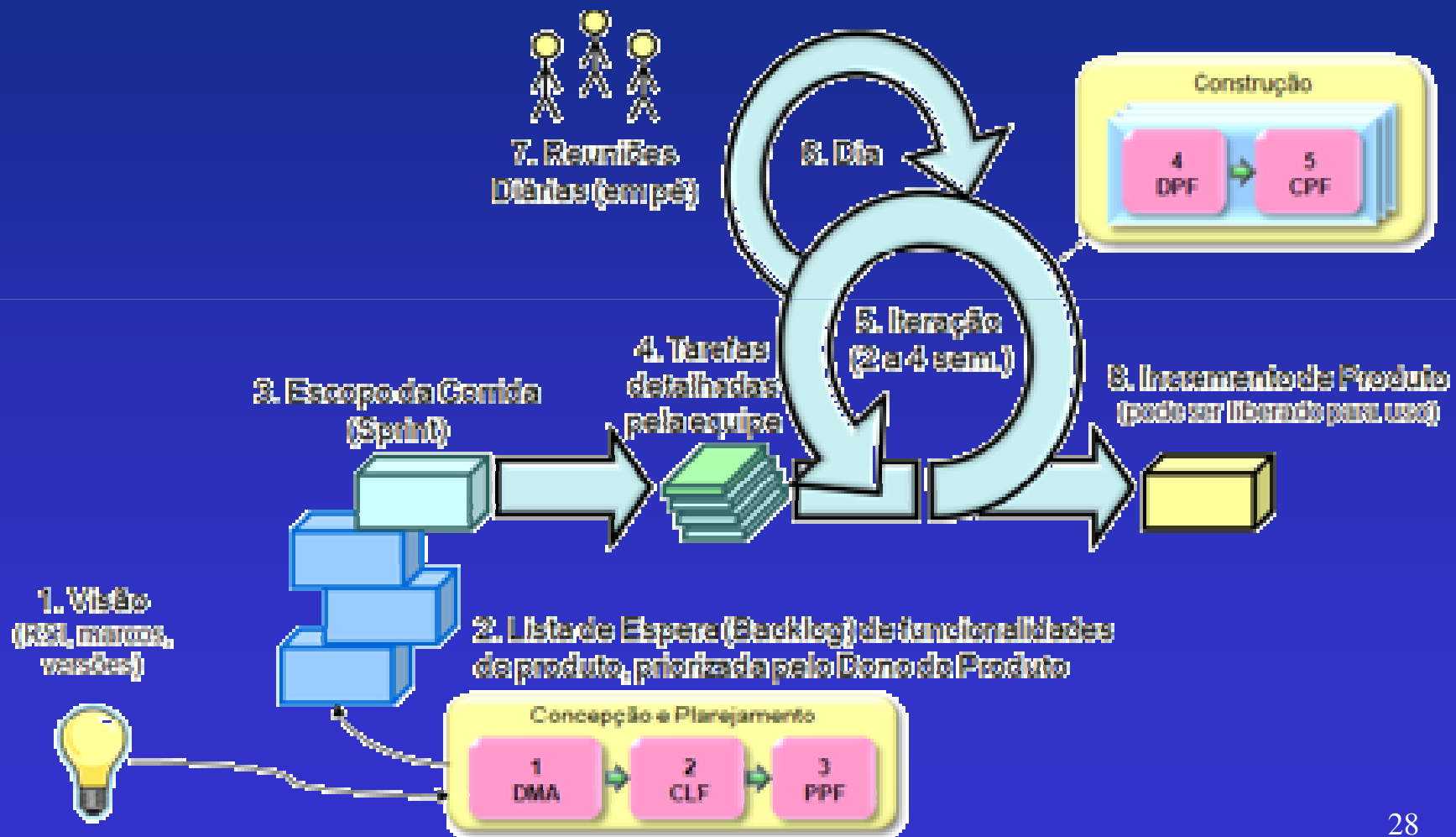
Em 1995, sua definição foi formalizada e implantada no desenvolvimento de softwares em todo o mundo.

O nome Scrum foi inspirado em um tipo de jogada que acontece no jogo de rugby para retornar a bola, onde é necessária a participação de todos os jogadores. Se jogador falhar, afeta toda equipe.

COMO USAR O SCRUM?

- **PRODUTO – SISTEMA E OU PARTES DE UM SISTEMA.**
- **TEAM ROLES – PAPÉIS DOS MEMBROS DO TIME.**
- **PRODUCT BACKLOG – LISTA DE REQUISITOS.**
- **RELEASE BACKLOG – PRIORIZAÇÃO E ESTIMATIVA DE TEMPO.**
- **BACKLOG – GRUPO REDUZIDO DE REQUISITOS PRIORIZADO.**
- **BURNDOWN CHARTS – MONITORA O PROGRESSO.**
- **DAILY SCRUM - REUNIÕES DIÁRIAS (OPCIONAL).**
- **SPRINTS - O CICLO – FLUXO DE TRABALHO.**

CICLO DE VIDA



PRODUTO – TIPO DE SISTEMA



Apesar do Scrum ter sido destinado para gerenciamento de projetos de software, ele pode ser utilizado em equipes de manutenção de software ou como uma abordagem geral de gerenciamento de projetos/programas.

TEAM ROLES – REGRAS DO TIME



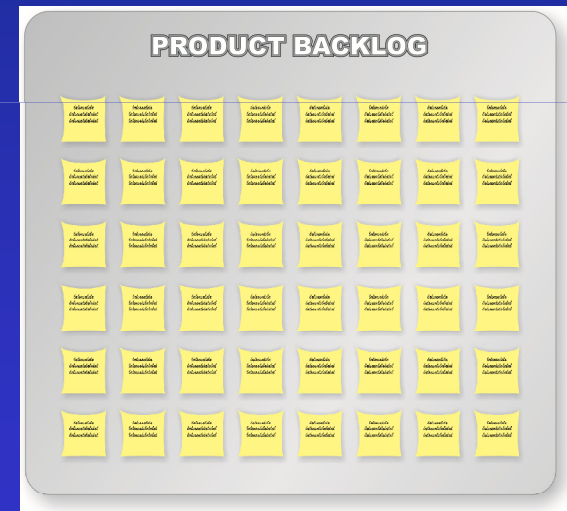
-Desenvolvedores

PRODUCT BACKLOG – LEVANTAMENTO DE REQUISITOS

O cliente e a equipe de desenvolvimento juntos fazem o **PRODUCT BACKLOG**, que nada mais é que o Levantamento de Requisitos.

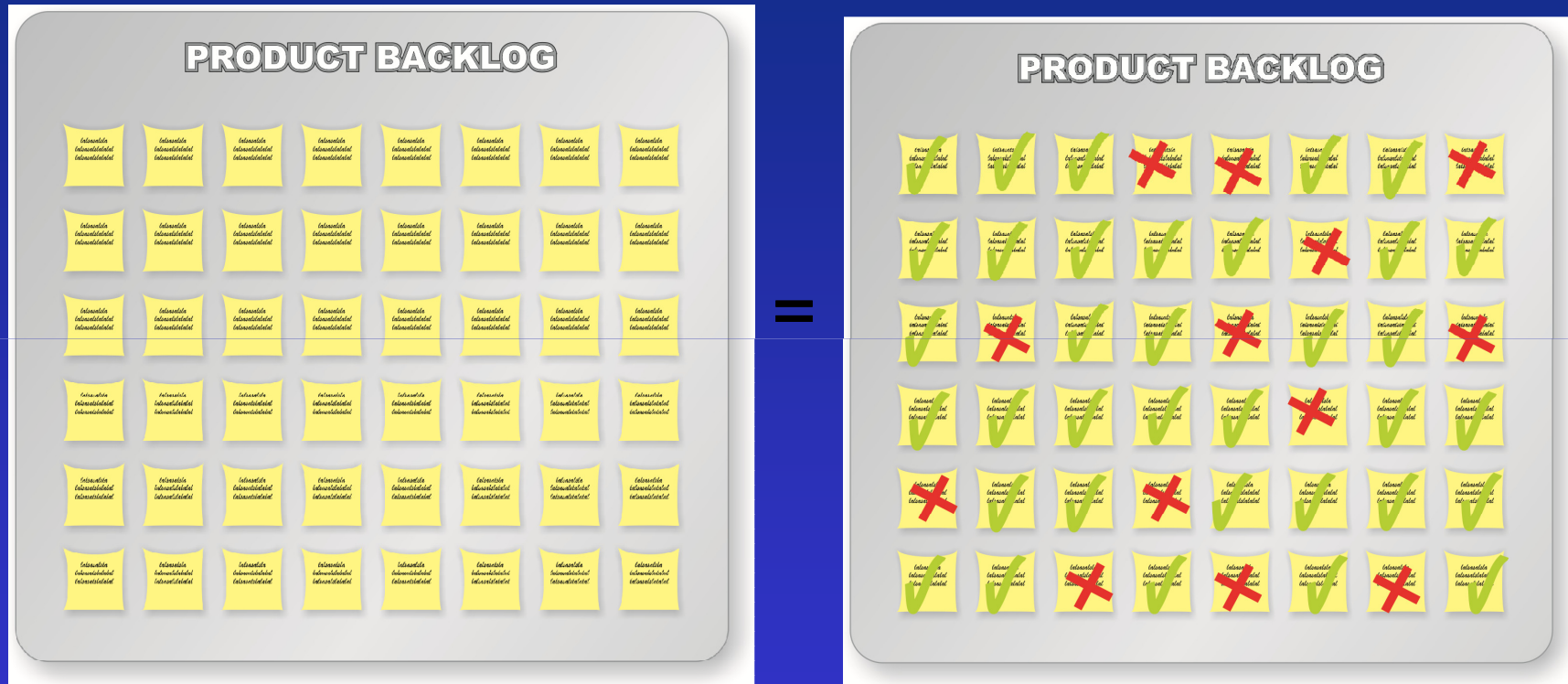


+ CLIENTE =



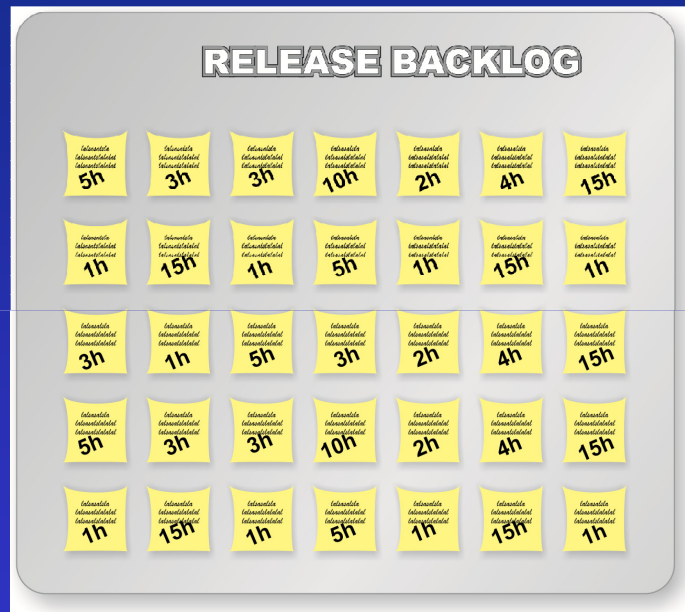
- Lista de tarefas para desenvolver o produto.

PRODUCT BACKLOG – LEVANTAMENTO DE REQUISITOS

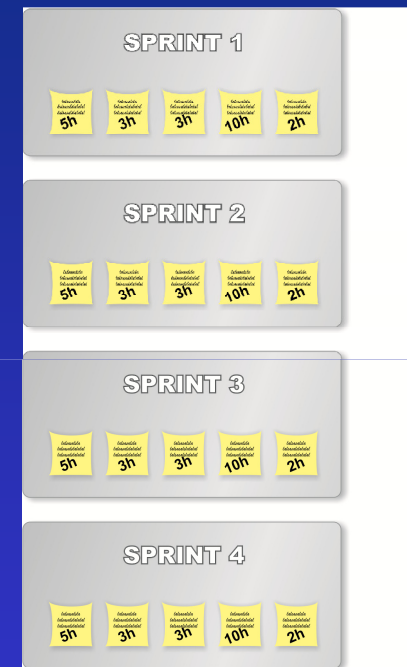


**AS PRINCIPAIS TAREFAS SÃO SELECIONADAS
SEPARADAS E COLOCADAS NO RELEASE
BACKLOG.**

SPRINTS – CICLOS DE DESENVOLVIMENTO

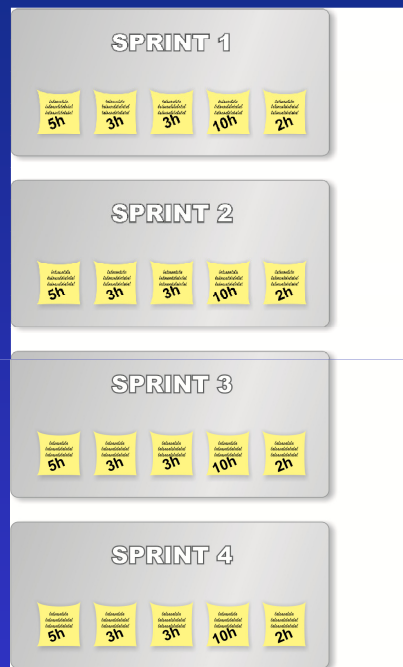


=



O RELEASE É DIVIDIDO EM DIVERSOS SPRINTS QUE SÃO DESENVOLVIDOS CONFORME A PRIORIDADE E OS PRAZOS ESTIMADOS.

• BURNDOWN CHARTS (MONITORA O PROGRESSO)



PRODUTIVIDADE DIA-A-DIA

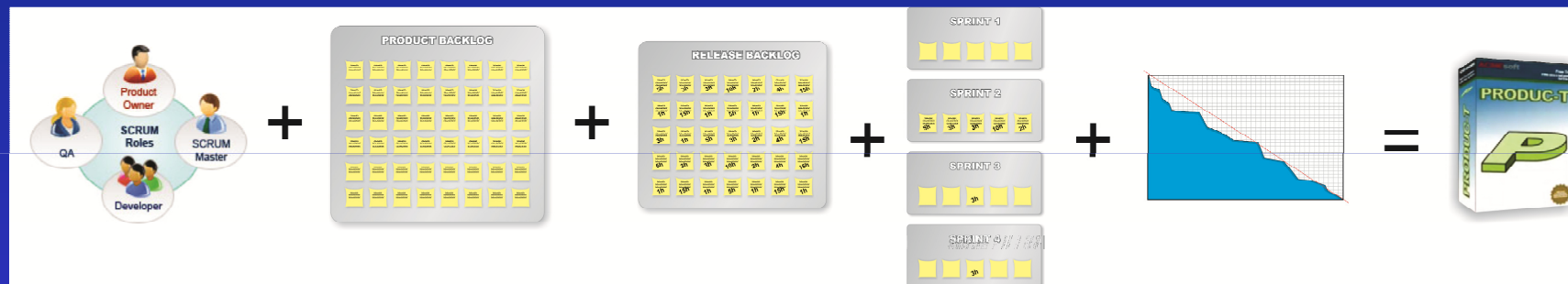
O BURNDOWN É A VISUALIZAÇÃO DA PROGRESSÃO DO PROJETO DIA-A-DIA, ATRAVÉS DELE É POSSÍVEL SABER SE O PROJETO SERÁ ENTREGUE NO PRAZO.

- **DAILY SCRUM
(REUNIÕES DIÁRIAS)**



**DEPENDENDO DA EXPERIÊNCIA DA EQUIPE PODEM SER
FEITAS REUNIÕES CURTAS DIÁRIAS, EM PÉ PARA NÃO
PERDER TEMPO. EVITAM ATRASOS NO SPRINTS.**

• CICLO SCRUM (FLUXO DE TRABALHO)



ACIMA TEMOS O CICLO DE DESENVOLVIMENTO SCRUM.

Vantagens:

- ✓ Os papéis são bem definidos, todos têm conhecimento sobre as suas responsabilidades;
- ✓ É um processo ágil e flexível, tornando melhor a reação as mudanças que ocorrem durante o projeto;
- ✓ É focado no controle e gerenciamento, buscando minimizar os riscos e maximizar a qualidade;
Os times são pequenos, a comunicação é mais eficiente;

Desvantagens:

- X** Ausência de práticas de Engenharia de Software, pois é voltada para o gerenciamento do projeto;
- X** Necessita a associação de uma outra metodologia de Engenharia de Software, por exemplo XP;
- X** É difícil de ser implementada, principalmente devido a resistência de mudanças culturais.