

Engenharia de Software

Professor:

Zady Castaneda Salazar



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Campinas

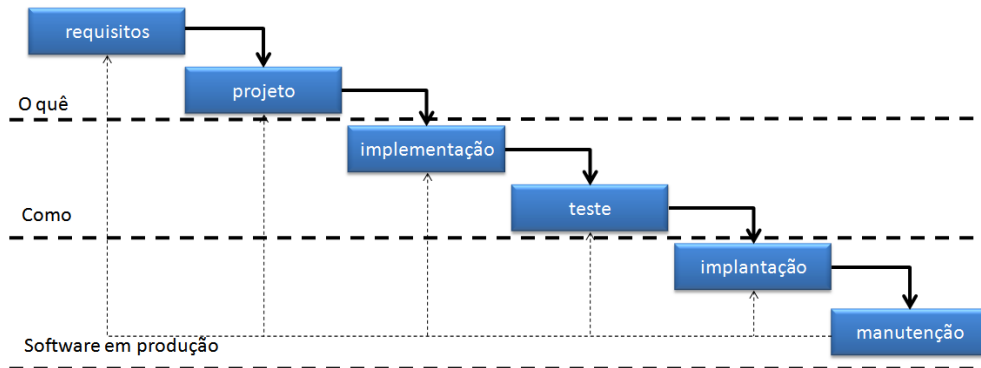
Aula 4

Processos, Métodos e Modelos da Engenharia de Software

Agenda:

- Modelo ágeis de processo de software
 - **Programação Extrema XP**
 - **Scrum**

Modelos tradicionais



Modelo cascata

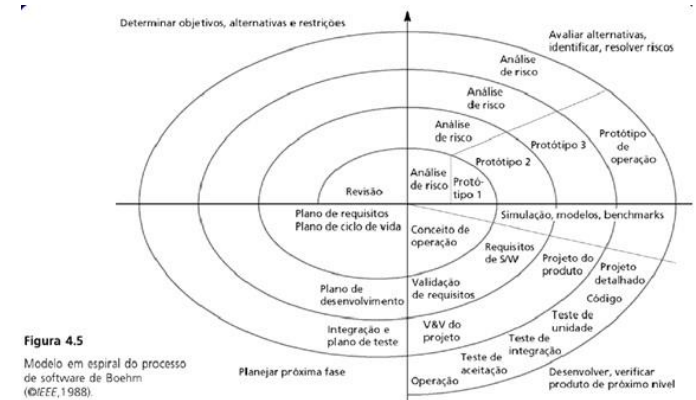


Figura 4.5
Modelo em espiral do processo
de software de Boehm
(QIEEE, 1988).

Modelo espiral

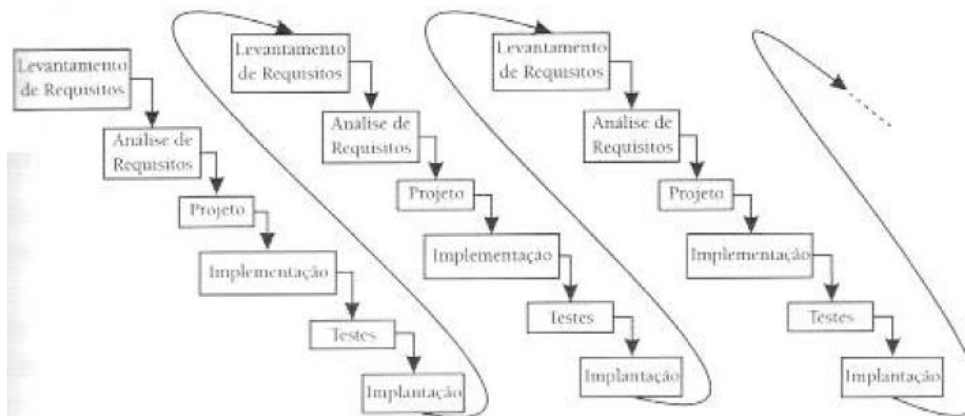
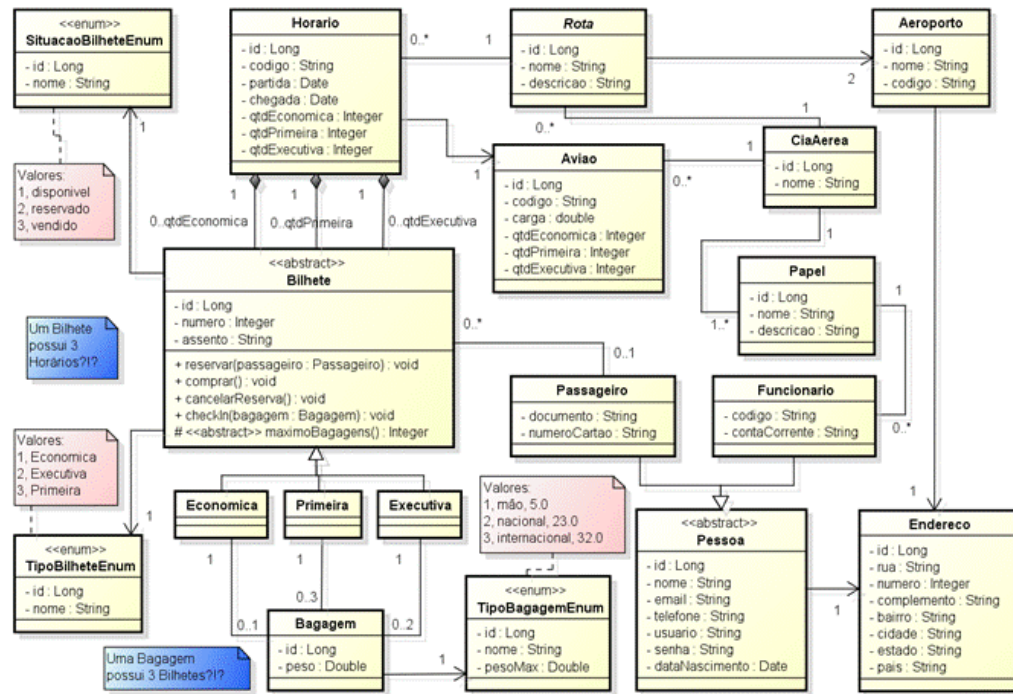


Figura 2-2 No processo incremental e iterativo, cada iteração é uma “minicascata”.

Modelo incremental

Engenharia de software : Modelos tradicionais

- Modelos tradicionais colocam grande ênfase em **projetar antes de construir**.



The screenshot shows the decolar.com website interface for booking flights. The main navigation bar includes links for Hotéis, Passagens, Pacotes, Carros, Ingressos, Disney, and Se. The search section is titled "Encontre seu Voo" and includes a green button "Economize até 35%!". The search criteria are set to "Ida e Volta" (one-way), "Só ida" (round-trip), and "Multidestino" (multi-stop). The origin is "São Paulo, Brasil" and the destination is "Aeroporto Internacional Jose Martin, H". The dates are "02 Jan 2018" and "16 Jan 2018", with a duration of "15 Dias". The number of passengers is "1". The search results show "Até 1 escala" (up to 1 stop) and "Sem preferência" (no preference) for the class of service. The search button is labeled "Procurar".

Engenharia de software : Modelos tradicionais

Em gestão de projetos as metodologias tradicionais funcionavam de maneira limitada porque:

- **Não conseguiam antecipar necessidades** que futuramente poderiam ser supridas;
- Perdiam **muito tempo no planejamento** e **demoravam mais para produzir**;
- Tanto a empresa quanto o cliente precisam **ter uma noção** muito específica do **tempo de desenvolvimento**;
- Qualquer alteração faz com que seja necessário **um grande retrabalho** e **gera atrasos**;
- Grande parte dos **erros no projeto passa despercebida** e são notados apenas pelo cliente final.

Engenharia de software : Modelos tradicionais

As empresas hoje em dia operam em um ambiente global em **rápida mudança**.

Precisam responder às **novas oportunidades e mercados**.

Então o novo software tem que **ser desenvolvido rapidamente**, para que seja possível **tirar vantagem das novas oportunidades** e **responder á pressão da concorrência**.

Engenharia de software : Modelos tradicionais

ENTREGA E DESENVOLVIMENTO RÁPIDO SÃO OS REQUISITOS MAIS IMPORTANTES DA MAIORIA DOS SISTEMAS DE NEGOCIOS.

ATUALMENTE AS EMPRESAS PODEM ESTAR **DISPOSTAS A NEGOCIAR A QUALIDADE E COMPROMETER REQUISITOS** SE PUDEREM IMPLANTAR UM NOVO SOFTWARE RAPIDAMENTE.

Engenharia de software : Modelos tradicionais

Para alguns tipos de software — como os sistemas de controle críticos em segurança, para os quais uma análise completa do sistema é essencial —, essa abordagem dirigida por plano é a mais indicada. No entanto, em um ambiente empresarial dinâmico, isso pode ser problemático. Quando o software finalmente estiver disponível para uso, o motivo original de sua aquisição pode ter mudado tão radicalmente que ele acaba sendo inútil. Portanto, especialmente no caso de sistemas de negócio, os processos de desenvolvimento e entrega rápidos são essenciais.

Hoje, muitas empresas abandonaram esses modelos em alguns projetos e buscam trabalhar com **metodologias ágeis**.



Métodos ágeis de desenvolvimento de software

- Movimento iniciado por programadores experientes e consultores em desenvolvimento de software.
- Questionam e se opõem a uma série de mitos/práticas adotadas em abordagens tradicionais de Engenharia de Software e Gerência de Projetos.
- Manifesto Ágil: Assinado por 17 desenvolvedores em Utah em fevereiro/2001.



Os valores preconizados pelo **Manifesto Ágil** são:

- **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.

Os **doze princípios Ágeis** foram cunhados a partir do **Manifesto Ágil** por parte de seus autores.

Princípios.

1. Garantia da satisfação do consumidor com **entrega rápida e contínua de softwares funcionais**.
2. **Mudanças de requisitos**, mesmo no fim do desenvolvimento, ainda são bem-vindas.
3. Frequentemente são entregues **softwares funcionais** (semanas, ao invés de meses).
4. Desenvolvedores e pessoas relacionadas aos negócios devem **trabalhar, em conjunto**, até o fim do projeto.

Princípios.

5. Construir projetos com **indivíduos motivados**, dar-lhes ambiente e suporte necessários e confiar que farão seu trabalho.

6. Uma **conversa cara a cara** é o método mais eficiente e efetivo de transmitir informações para e dentro de uma equipe de desenvolvimento.

7. **Software em funcionamento** é a principal medida de progresso.

8. **Desenvolvimento sustentável**, de modo a manter um ritmo constante indefinidamente.

Princípios.

9. **Atenção contínua** para com a excelência técnica e para com bons projetos aumenta a agilidade.

10. **Simplicidade** – a arte de maximizar a quantidade de trabalho não efetuado – é essencial.

11. As melhores arquiteturas, requisitos e projetos emergem de **equipes auto-organizáveis**.

12. Em intervalos regulares, a equipe deve **refletir sobre como se tornar mais eficiente**.

O QUE SÃO MÉTODOS ÁGEIS?

- Conjunto de metodologias.
- Desenvolvimento de softwares de 1 a 4 semanas.
- Iteração (Planejamento, Requisitos, Projeto, Codificação, Teste e Documentação).
- Ao término de cada iteração = versão do software para validação.

Os métodos ágeis têm sido particularmente úteis para dois tipos de desenvolvimento de sistemas:

1. O desenvolvimento de um produto pequeno ou médio, por uma empresa de software, para venda. Praticamente todos os produtos de software e aplicativos são desenvolvidos atualmente usando uma abordagem ágil.
2. O desenvolvimento de sistemas personalizados dentro de uma organização, em que há um compromisso claro por parte do cliente de se envolver no processo de desenvolvimento, e no qual há poucos *stakeholders* externos e normas que afetem o software.

Os métodos ágeis funcionam bem nessas situações porque possibilitam comunicação contínua entre o gerente de produto ou o cliente do sistema e o time de desenvolvimento. O software em si é um sistema *stand-alone*, em vez de estar intimamente integrado com outros sistemas que estejam sendo desenvolvidos simultaneamente. Consequentemente, não há necessidade de coordenar fluxos de desenvolvimento paralelos. Os sistemas pequenos e médios podem ser desenvolvidos por times situados no mesmo local, então a comunicação informal entre seus membros funciona bem.

EXEMPLO DE METODOLOGIAS ÁGIES



➤ Scrum



➤ XP (Extreme Programming)

➤ Open Up (Open Unified Process)

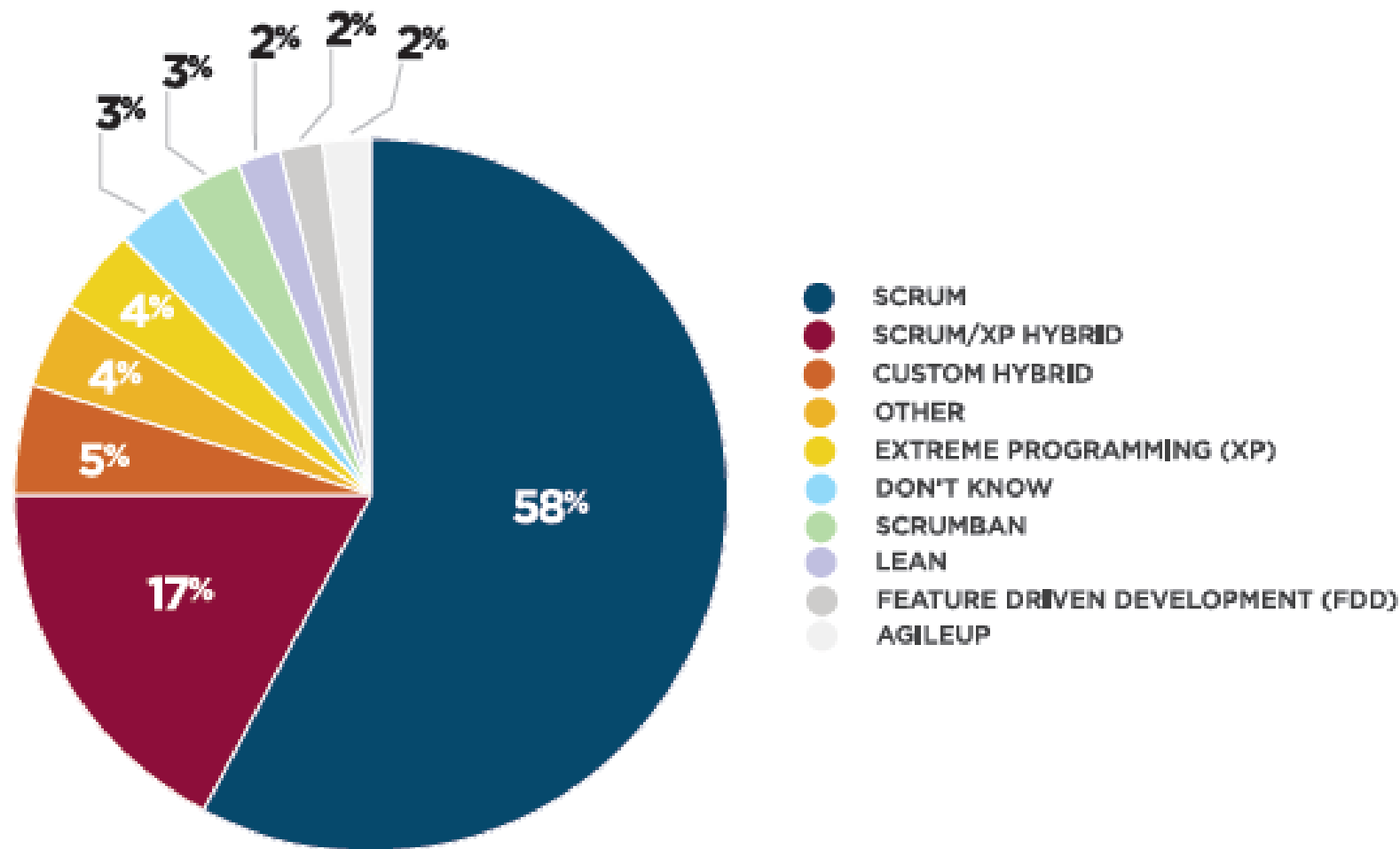
➤ AUP (Agile Unified Process)

➤ FDD (Feature Driven Development)

➤ TDD (Test Driven Development)

➤ DSDM (Dynamic Systems Development Method)

Metodologias ágeis.



Fonte: <http://www.devmedia.com.br/uso-de-metodologias-ageis-em-uma-organizacao-baseada-em-linha-de-produto-artigo-revista-engenharia-de-software-magazine-38/21662>

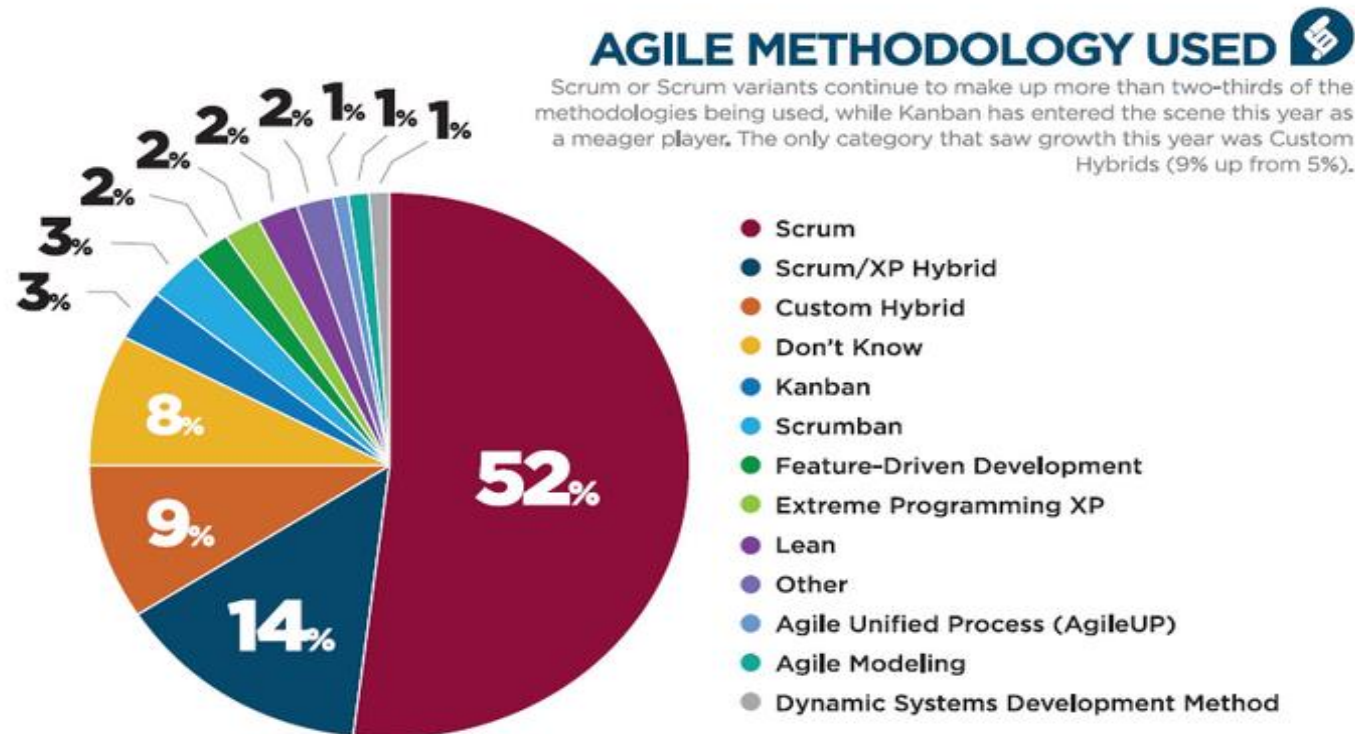
ano 2011

Metodologias ágeis.

No ano de 2012 foi elaborado um questionário Nacional sobre Métodos Ágeis .

A iniciativa partiu do departamento de Ciência e Computação IME-USP que contou com a participação de 466 participantes respondendo o questionário pela internet voluntariamente.

O questionário foi elaborado com base no trabalho da VersionOne, empresa que realiza essa pesquisa em nível global anualmente.

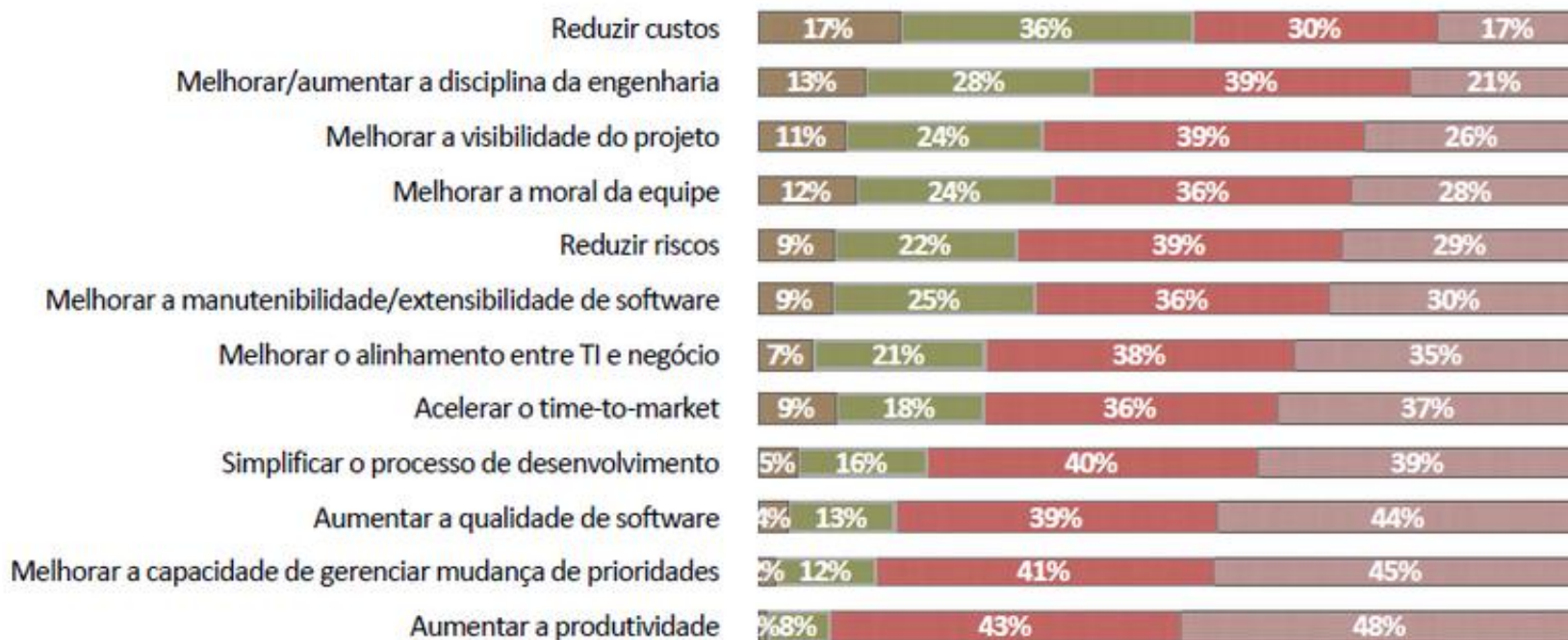


Fonte: <https://blog.myscrumhalf.com/o-estado-do-desenvolvimento-agil-no-brazil/ano-2012>

Metodologias ágeis.

RAZÃO MAIS IMPORTANTE PARA A ADOÇÃO DE MÉTODOS ÁGEIS NA EQUIPE/ORGANIZAÇÃO

■ Não importante ■ Um pouco importante ■ Muito importante ■ Maior importância



Fonte: <https://blog.myscrumhalf.com/o-estado-do-desenvolvimento-agil-no-brazil/>
ano 2012

Empresas que usam Metodologias Ágeis

- Google
- Microsoft
- Yahoo
- ThoughtWorks
- Siemens
- Nokia
- Philips
- Amazon.com
- HP
- Intel
- Xerox
- SEA Tecnologia
- OnCast
- Globo.com
- Abril
- UOL
- Improve It
- Caelum
- Teamware
- Objective Solutions
- LocaWeb



Programação Extrema XP

Programação Extrema XP

Uma das abordagens mais importantes para a mudança da cultura no desenvolvimento de software constitui o desenvolvimento da PROGRAMAÇÃO EXTREMA (XP).

A base de essa abordagem foi a ideia de **levar a níveis “extremos” boas praticas reconhecidas**, como por exemplo o desenvolvimento iterativo.

FIGURA 3.3 Ciclo de lançamento (*release*) da Programação Extrema.



Extreme Programming é um método que vai a **priorizar quais as partes mais importantes do Projeto**, e quem diz quais as partes importantes são os próprios **clientes**.



- O plano de iteração é a **lista de histórias selecionadas para serem implementadas dentro de uma iteração.**
- Um “laço” de iteração do XP geralmente é curto e tem uma duração de no **máximo duas semanas.**
- A definição de quais histórias entram num ciclo parte do **cliente que define de acordo com suas prioridades.**
- **Histórias muito complexas são retiradas do ciclo e repensadas para voltar posteriormente.**

Programação Extrema XP

Na XP, os requisitos são expressos em cenários (chamados de histórias do usuário) implementados diretamente como uma série de tarefas. Os programadores trabalham em pares e desenvolvem testes para cada tarefa antes de escreverem o código. Todos os testes devem ser executados com sucesso quando o novo código é integrado ao sistema, já que há um curto intervalo de tempo entre os lançamentos (*releases*) do sistema.



Na prática, a aplicação da Programação Extrema como proposta originalmente se provou mais difícil do que o previsto. Ela não pode ser integrada de imediato às práticas de gestão e à cultura da maioria das empresas; assim, as que adotam métodos ágeis selecionam as práticas de Programação Extrema mais adequadas ao seu modo de trabalho. Às vezes, essas técnicas são incorporadas aos processos de desenvolvimento das próprias empresas, mas, na maioria das vezes, são utilizadas em conjunto com um método ágil focado em gerenciamento, como o Scrum



Quando usar ?

A Extreme Programming (**XP**) é uma **Metodologia Ágil** para equipes pequenas e médias que desenvolvem software baseado em requisitos vagos e que se modificam rapidamente.



Metodologias ágeis.



SCRUM

- Metodologia cujas práticas **são aplicadas em um processo iterativo e incremental.**
- Indicado para **projetos complexos e imprevisíveis.**
- Modelado para se adaptar à **mudanças em qualquer fase do projeto.**
- A evolução do projeto é **acompanhada de perto por um representante do cliente.**

SCRUM

As necessidades do negócio vão determinar a prioridade do desenvolvimento.

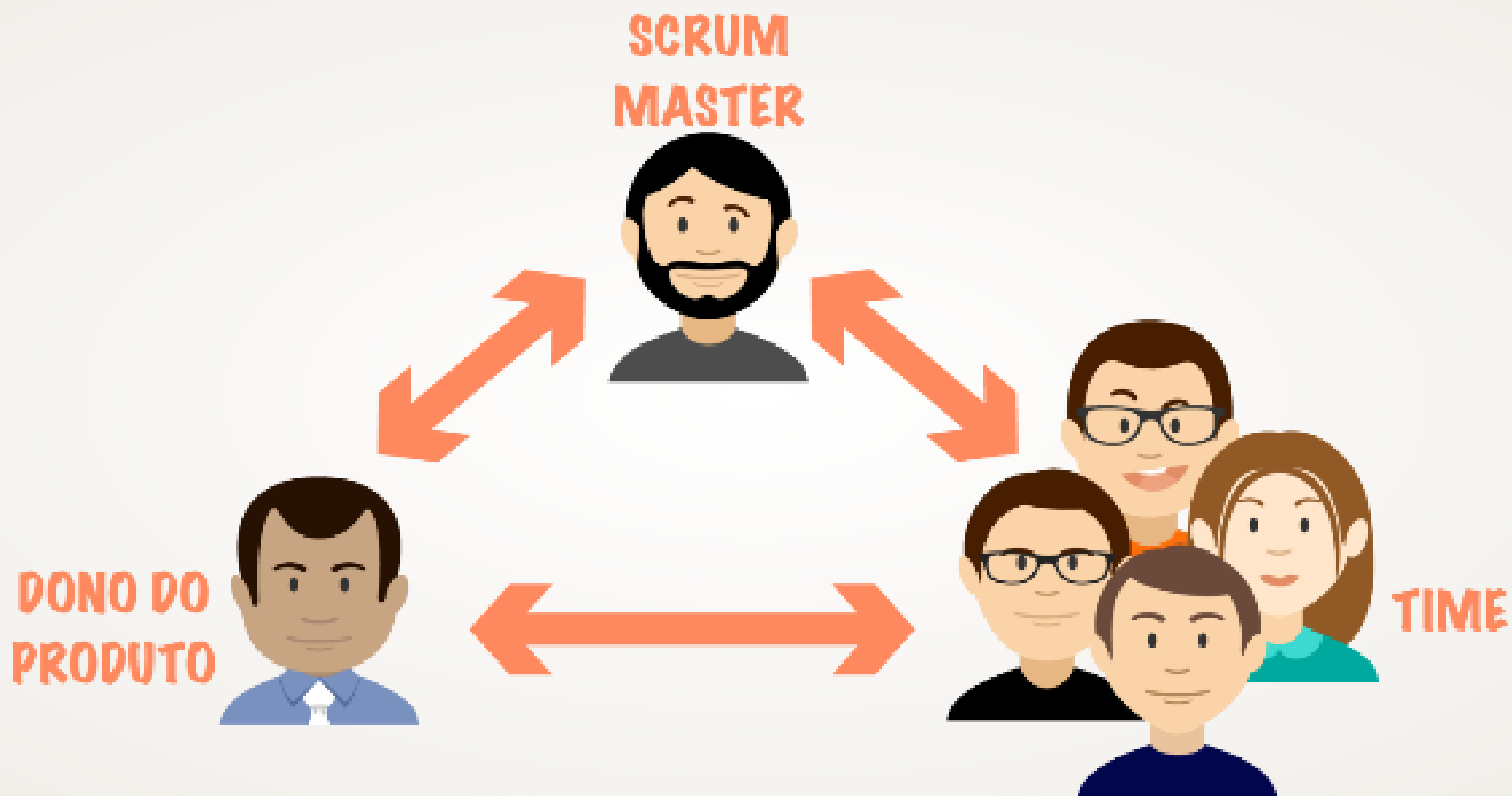
Ocorrem reuniões diárias para:

- **Planejar as iterações.**
- **Determinar as tarefas.**
- **Gerenciar os riscos.**
- **Definir prioridades.**



Metodologias ágeis.

Papéis: SCRUM.

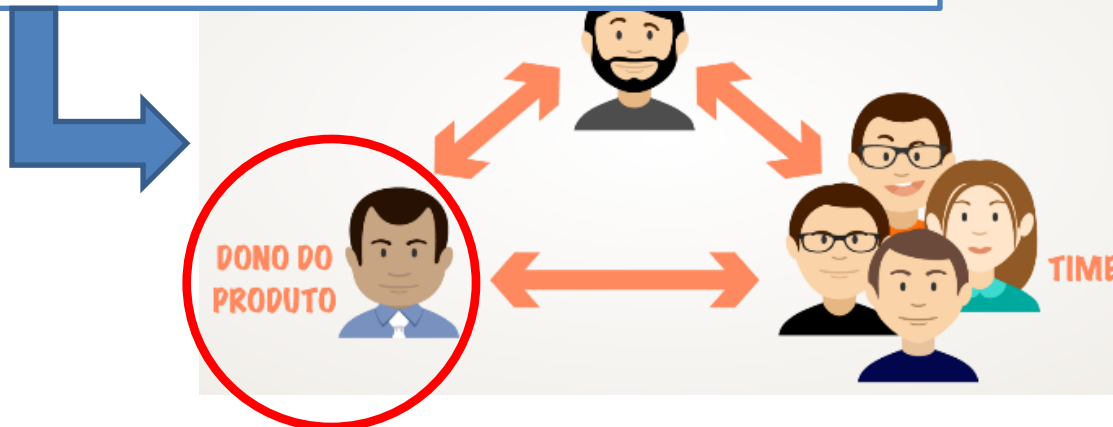


Metodologias ágeis.

Papéis: SCRUM.

- Chamado de P.O. (*Product Owner*), é a **pessoa que representa o cliente.**
- Para isso ele precisa ter uma **atitude ativa no projeto**, se comunicar bem com os *Stakeholders* para alinhar os objetivos do projeto, com os usuários para compreender as reais necessidades.
- Ele **organiza e prioriza o *Product Backlog*** (que é a lista de itens que devem ser desenvolvidos) de forma que o esforço do time de desenvolvimento possa trazer o maior retorno possível para o negócio.

Essa **atitude é peça chave** para que o P.O. possa conseguir maximizar o retorno sobre o investimento.

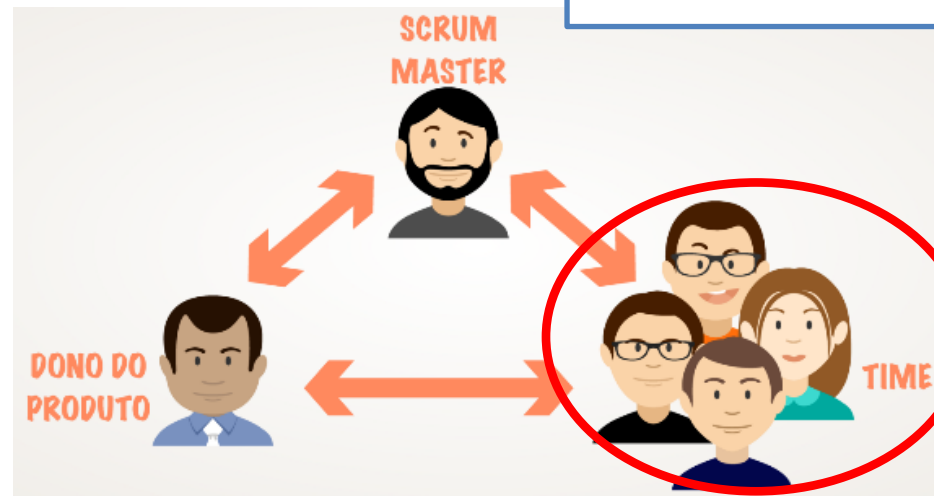


Metodologias ágeis.

Papéis: SCRUM.

Não há uma divisão em cargos ou sub-times, mesmo que haja membros do time com competências diferentes, o sucesso ou o fracasso do desenvolvimento são de responsabilidade do time todo.

- grupo pequeno, normalmente, entre 5 e 9 pessoas;
- deve ser **comprometido com o trabalho** a fim de atingir a meta de um Sprint;
- **desenvolve e produz** com qualidade;
- deve ser **cada vez mais auto-gerenciável** e multidisciplinar.

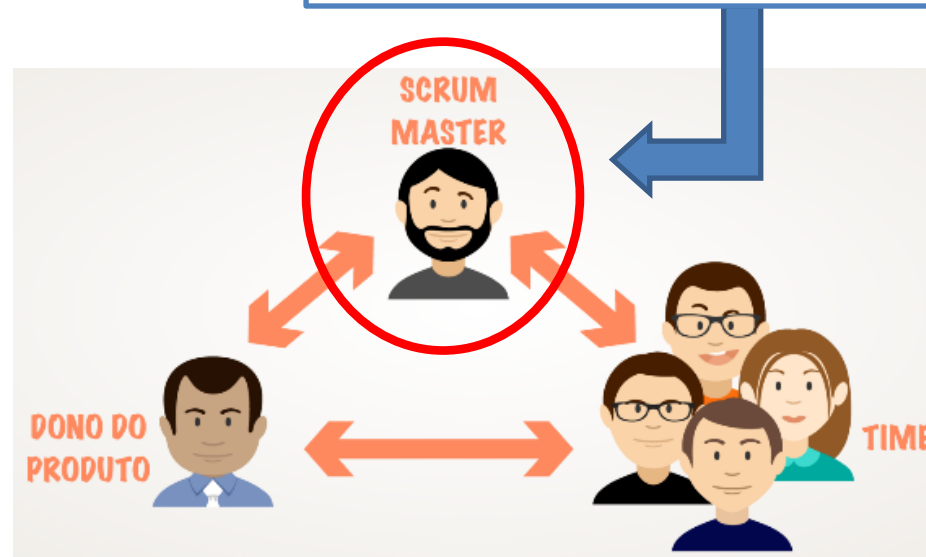


Metodologias ágeis.

Papéis: SCRUM.

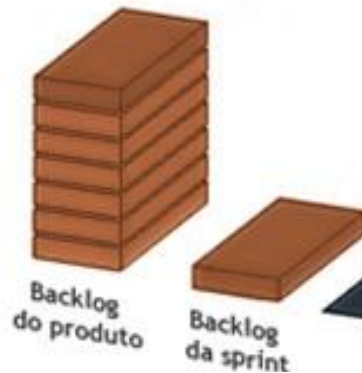
Ele não comanda o time de desenvolvimento, mas ele deve servir o time.

- deve ser **mediador e facilitador**.
- **remove impedimentos** da equipe assegurando que as práticas Scrum estão sendo executadas com eficiência.
- **deve dar suporte** ao amadurecimento do time, de modo que os pilares e as práticas sejam totalmente compreendidos. Em outras palavras, o objetivo do Scrum Master é tornar-se desnecessário.

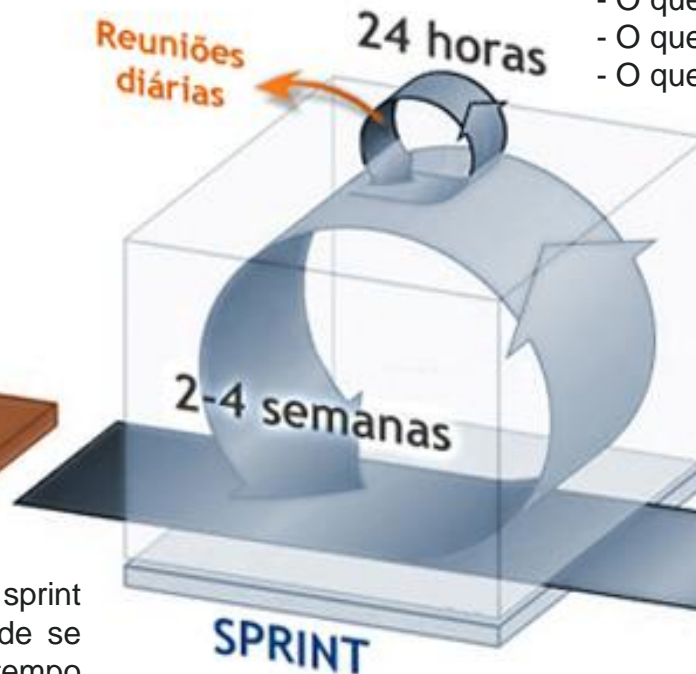


Ciclo do SCRUM.

Cria-se o **backlog** onde tem a lista de todas as funcionalidades que precisam ser desenvolvidas durante todo o projeto, junto com o stakeholder, onde precisa ser bem definido e detalhado no começo, listado e ordenado por prioridade do que é mais importante



Cria-se o sprint backlog, onde se decide o tempo necessário (dentro dos 30 dias) para criar a funcionalidade requisitada;



Depois de todo o planejamento, os itens do backlog são divididos nas equipes e entra no sprint que pode durar de 2 a 4 semanas.

A cada 24 horas tem uma reunião com os membros de equipes e questões devem ser respondidas como:

- O que você fez desde a última reunião?
- O que te impede de continuar?
- O que vai fazer até a próxima reunião?

Ao término do sprint as funcionalidades requisitadas são demonstradas.



Cerimônias do SCRUM

- Sprint Planning 1
- Sprint Planning 2
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Cada evento possui um **tempo máximo pré-determinado** para sua realização.

O **time-box** é o intervalo de tempo para realização de uma atividade. Deve-se ter disciplina e foco para fazer o trabalho determinado, sem desvios e devaneios, utilizando o tempo disponível do melhor modo possível.

Scrum Events	Timebox	
	2 weeks	1 month
Sprint Planning Meeting 1	2 hrs	4 hrs
Sprint Planning Meeting 2	2 hrs	4 hrs
Daily Scrum	15 min	15 min
Sprint Review	2 hrs	4 hrs
Sprint Retrospective	1,5 hrs	3 hrs

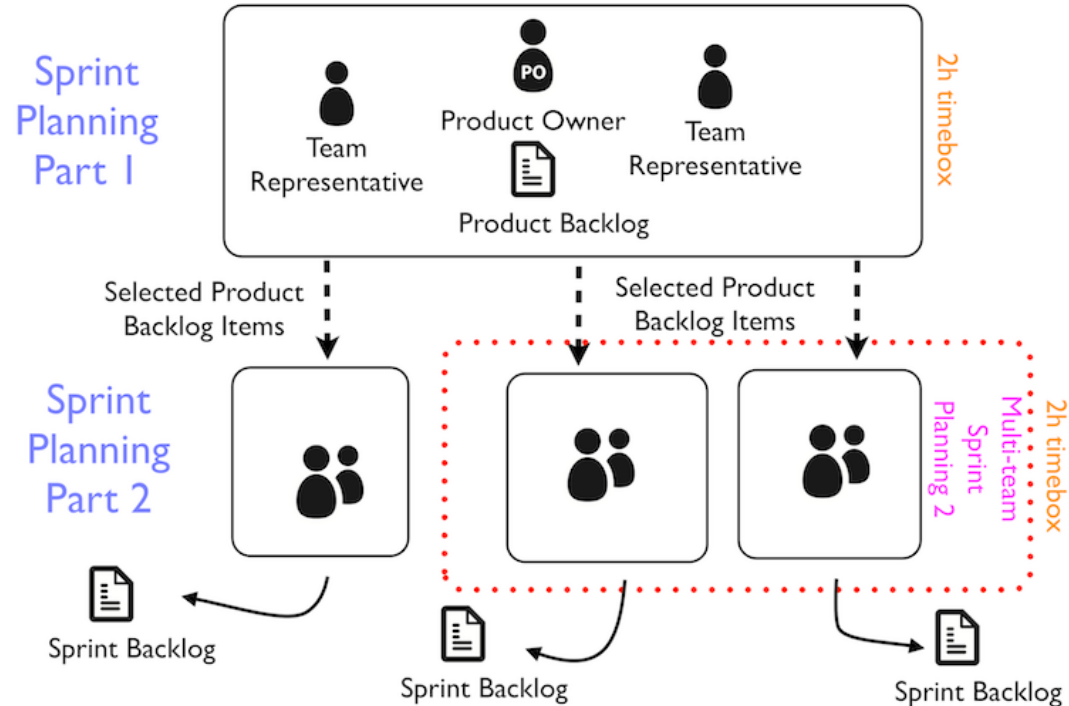
Cerimônias do SCRUM

Reunião de Estimativa :

- Preparação para o Sprint Planning (conjunto de atividades)
- Estimar baseado no tamanho, nunca em tempo
- Atualizar Product Backlog com as estimativas
- Importante para o **PO** criar o release plan

Cerimônias do SCRUM

• Sprint Planning 1



Product Backlog
Capacidade da equipe
Condições do Negócio

Revisa
Considera
Organiza

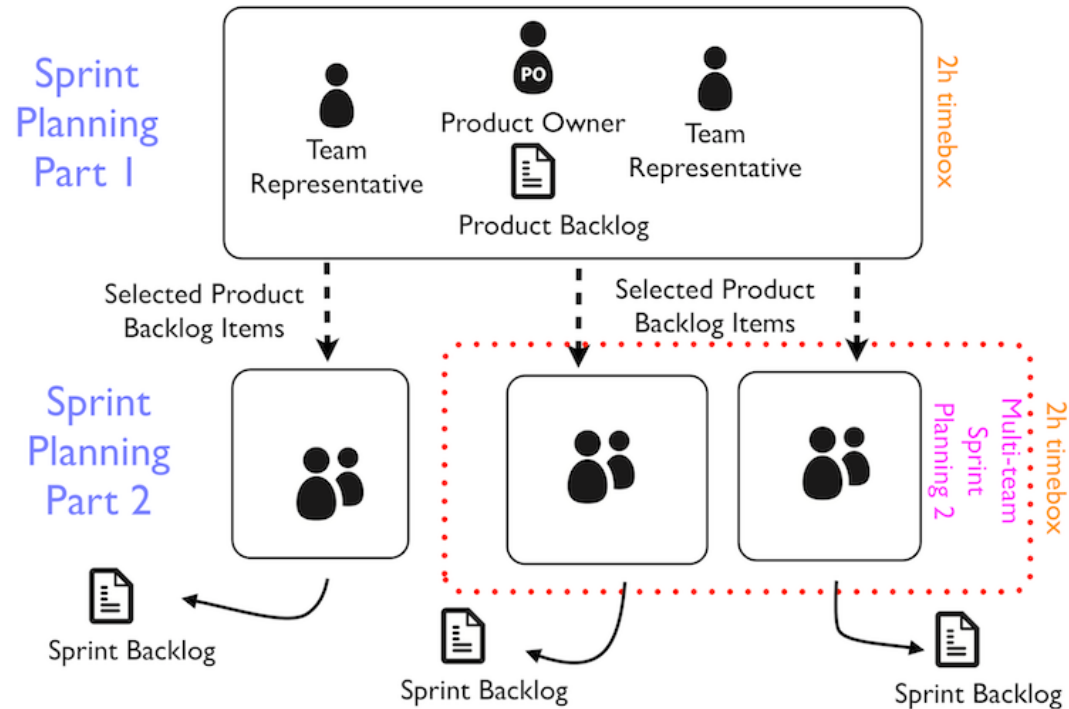
Objetivos da Sprint
Itens selecionados do backlog
Aceite do time

Cerimônias do SCRUM

• Sprint Planning 2

PO não precisa participar.

- É um planejamento tático da equipe.
- Os itens selecionados do Product Backlog são destrinchados em tarefas.
- Sprint Backlog.



Cerimônias do SCRUM

- **Daily Scrum** (diário): cada dia do Sprint a equipe faz uma reunião diária

Deve responder as três perguntas:

- O que fiz desde a ultima Daily Scrum?
- O que espero fazer até a próxima Daily Scrum?
- O que está impedindo o progresso?

Impedimentos são reportados.

Cerimônias do SCRUM

- **Sprint Review**

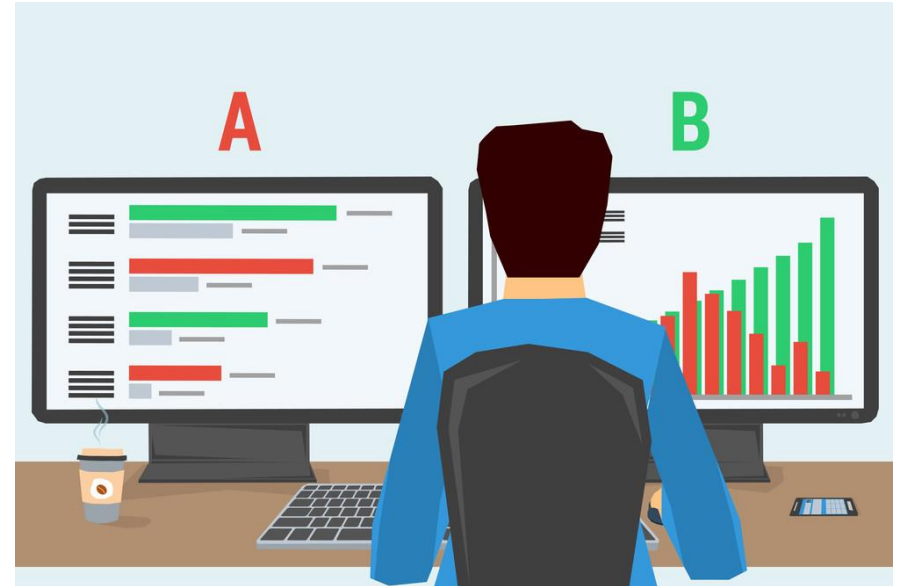
O que significa “pronto”?

- Team deve ter um **critério técnico** para indicar o que significa pronto!
- Incrementos funcionais são apresentados ao **PO** e interessados



Cerimônias do SCRUM

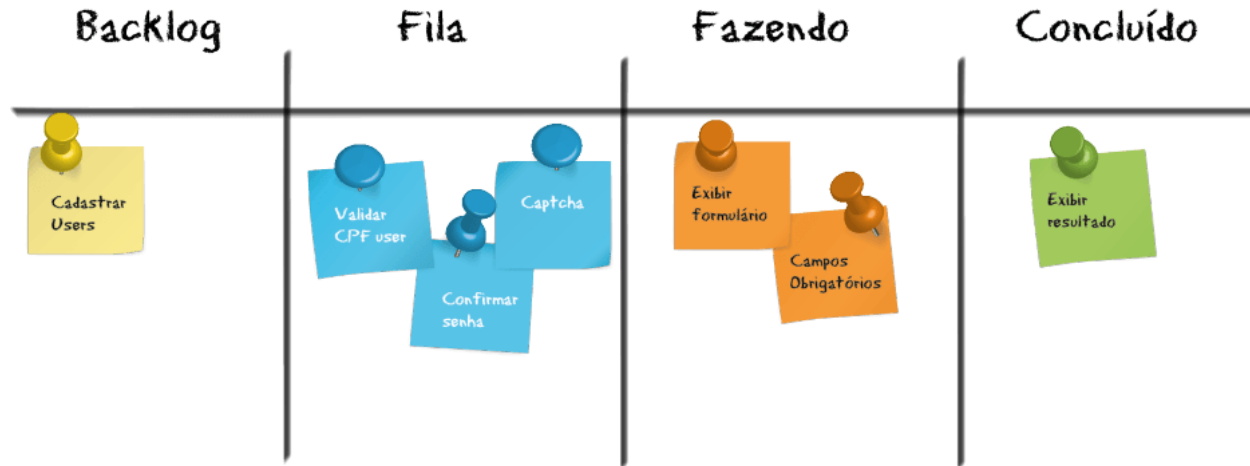
- **Sprint Retrospective**



- Team Backlog (para ajustar o processo)
- Backlog de impedimentos (mudanças na empresa)
- Os backlogs devem ser ordenados por importância

Metodologias ágeis.

Uma ferramenta importante no Scrum é o **quadro de trabalho**.



Backlog – lista com todos os itens a serem desenvolvidos.

Fila – o que vai entrar ou está na Sprint atual.

Fazendo – Coisas da Fila de desenvolvimento que já estão sendo feitas pelo time.

Concluído – Tarefas da Sprint atual que já foram finalizadas.

Ferramentas para Scrum: Como gerenciar seus projetos?

<https://www.siteware.com.br/projetos/ferramentas-para-scrum/>

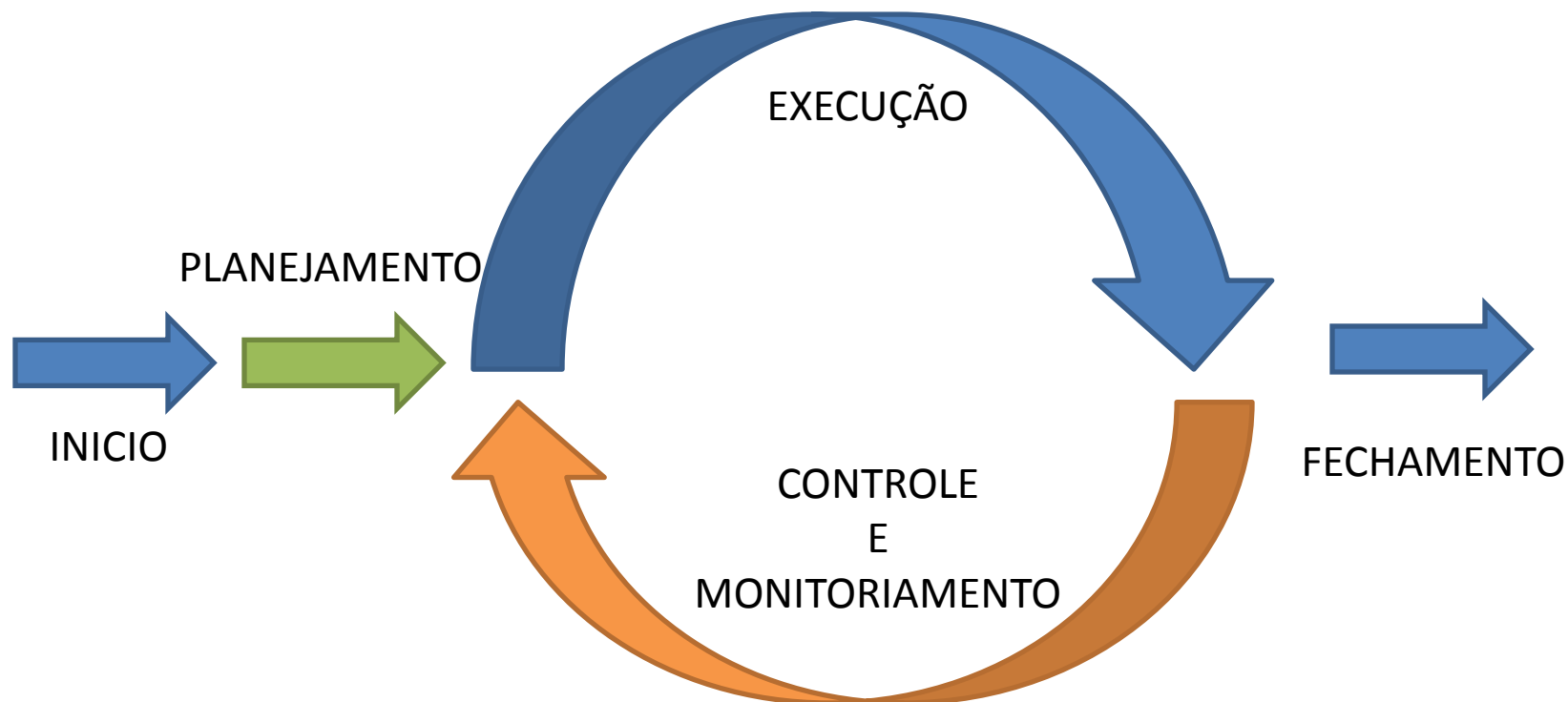
IceScrum

É uma aplicação web para acompanhamento do Scrum, seguindo suas regras/princípios.

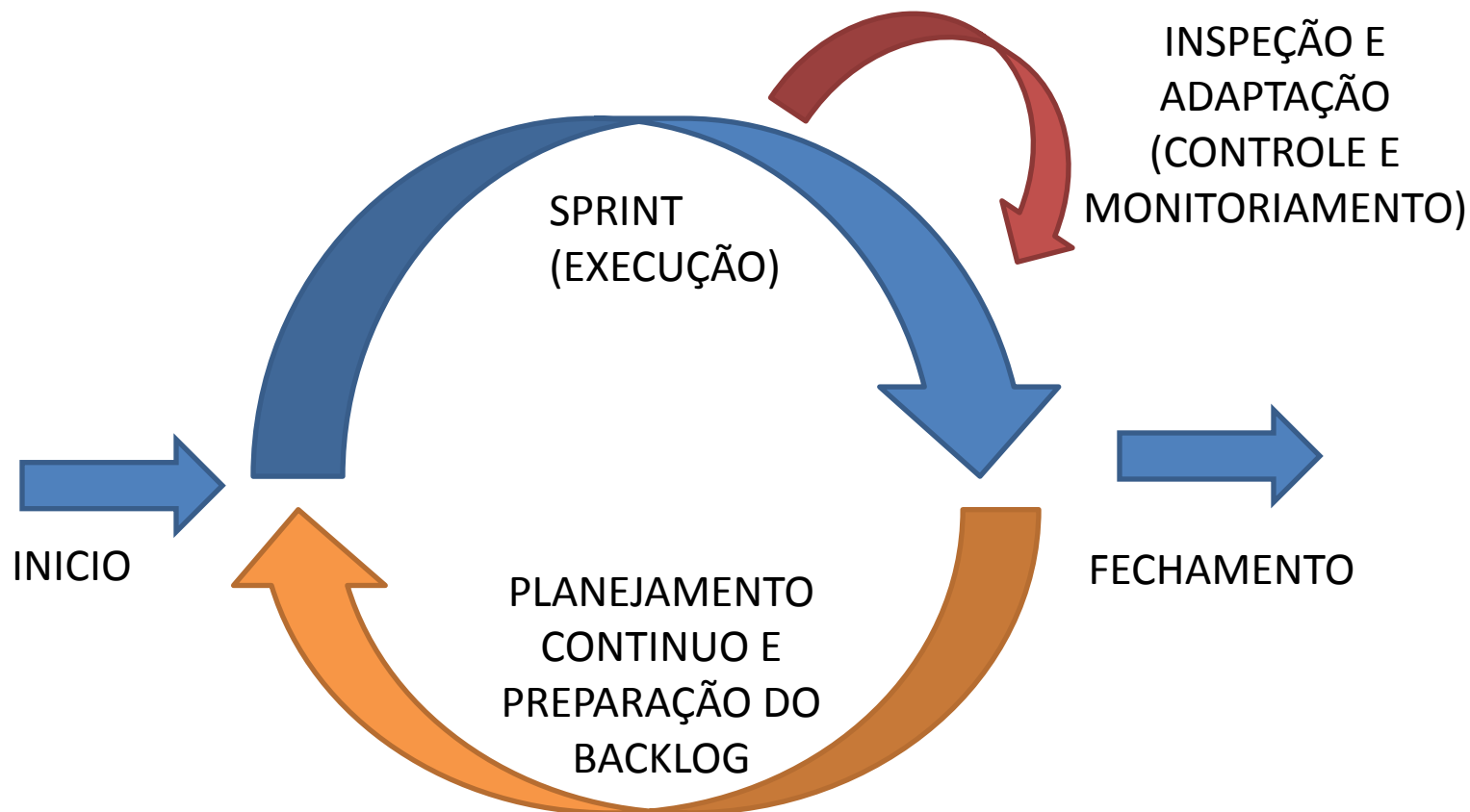
Nele existe as histórias de usuários, casos de aceitação para cada história, atores, sprint, gráfico de Burndown e Burnup (em tempo, pontos ou histórias), relatório de histórias de usuário, tarefas, etc.

Pode ser instalado ***gratuitamente***,

Processo de gestão de projeto tradicional



Processo de gestão de projeto usando srcum



Metodologias ágeis Vs tradicional

Isso não significa que o estilo tradicional não funciona mais em circunstância nenhuma; existe uma dose de rigor na abordagem tradicional da qual até mesmo a metodologia ágil pode beneficiar.

- Muitos profissionais ainda estão desacostumados a essa metodologia.
- Projeto tão complexo, a ponto de precisar de técnicas mais avançadas que reduzam a complexidade dele e consigam controlá-lo.
- As empresas não estão preparadas para o Scrum ou as empresas não sabem como usar segundo as restrições atuais da empresa.



- Bibliografia**



Capítulo 3	Desenvolvimento ágil de software	57
3.1	Métodos ágeis	59
3.2	Técnicas de desenvolvimento ágil	61
3.3	Gerenciamento ágil de projetos	69
3.4	Escalabilidade dos métodos ágeis	72

