

FIA/P GRADUAÇÃO

DISCIPLINA: PROJETO DE SISTEMAS APLICADO AS MELHORES PRÁTICAS EM QUALIDADE DE SOFTWARE E GOVERNANÇA DE TI

AULA:

5- Processo de Software

PROFESSOR:

RENATO JARDIM PARDUCCI

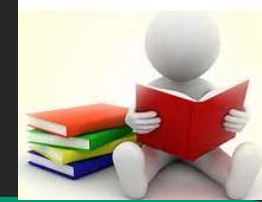
PROFRENATO.PARDUCCI@FIAP.COM.BR

[Renato Parducci - YouTube](#)

AGENDA DA AULA

- ✓ CMMi nível 2 de maturidade - PP,
- ✓ MS.br nível GPR
- ✓ CVS e Processo SW
- ✓ Escolha de um modelo de processo de software - Clássico x Ágil (CVS e processo)

ESTUDO DE CASO



Agora que está compreendido como funciona a evolução de um programa da qualidade, Consuelo vai iniciar um treinamento “on the job” com você e a equipe da GD da empresa de Dilan.

O primeiro passo é definir um processo de trabalho que seja seguido por todos e depois, recheá-lo com ferramentas de produtividade.

Consuelo preparou uma apresentação inicial sobre processos de software e você deverá entendê-los e escolher um para trabalhar nos projetos da empresa.

Atenda com atenção a esse treinamento da consultora!

DESENVOLVIMENTO DA CAPACIDADE E MATURIDADE NA QUALIDADE DE SOFTWARE

Vamos iniciar a jornada de aprendizado sobre como desenvolver práticas que possibilitem evoluir os níveis de Qualidade de uma empresa na produção de Software.

A partir de agora, serão percorridos os níveis do CMMi, com demonstração das principais técnicas e ferramentas que permitem cumprir com as determinações do Guia da Qualidade e alcançar certificações de reconhecimento do mérito.

**CMMi nível 2 de maturidade - PP,
MS.br nível GPR**

**Escolha de um Processo de
Software**

O que você vai procurar resolver agora:

Nível de qualidade (certificação)	Áreas de prática (áreas de assuntos cujo processos devem ser descritos e praticados na empresa)	
5	<i>Análise de Causas (CAR)</i>	
4	<i>Gerenciamento e medição do desempenho (MPM)</i>	
3	<ul style="list-style-type: none"> *<i>Gerenciamento e desenvolvimento de requisitos (RDM)</i> *<i>Verificação e validação (VV)</i> *<i>Revisão por pares (PR)</i> *<i>Solução técnica (TS)</i> *<i>Integração de produto (PI)</i> *<i>Gestão de risco e oportunidade (RSK)</i> 	<ul style="list-style-type: none"> *<i>Treinamento organizacional (OT)</i> *<i>Análise e tomada de decisão (DAR)</i> *<i>Implementação de infraestrutura (II)</i> *<i>Processo de desenvolvimento (PAD)</i> *<i>Gestão do processo de software (PCM)</i>
2	<ul style="list-style-type: none"> *<i>Garantia de qualidade de processo (PQA)</i> *<i>Gerenciamento do acordo de serviço com fornecedores (SAM)</i> *<i>Estimativa (EST)</i> *<i>Planejamento (PLAN)</i> *<i>Monitoração e controle (MC)</i> 	<ul style="list-style-type: none"> *<i>Gerenciamento de configuração (CM)</i> *<i>Governança (GOV)</i>

METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

Existem **metodologias de condução de projetos** de software que, seguindo um Ciclo de Vida (Cascata, Espiral, Incremental ou Prototipação), **explicam processos, tarefas, artefatos e responsabilidades** na execução de trabalhos de planejamento execução e controle da produção.

Modelos de processo de software



MODELO PREDITIVO

Processo preditivo, segundo o FLEKS é aquele quando há uma definição clara ou quase clara do que deve ser feito, e o produto ou serviço tem certo grau de estabilidade.

Nesse caso, tem-se projetos caracterizados por estarem organizados em fases bem definidas, sequenciais, as quais empregam um conjunto de recursos (pessoas e ferramentas) e conhecimentos para realizar entregas específicas que vão gerando gradativamente o produto final.

Em TI, esse modelo é conhecido também como ABORDAGEM CLÁSSICA de produção de software.

Modelos de processo de software



O **modelo preditivo** permite uma condução onde parte-se de um planejamento geral para depois executar e controlar o projeto, seguindo práticas propostas no PMBoK (Project Management Body of Knowledge) do PMI (Project Management Institute):

- **Escopo:** requisitos de produto e das atividades do projeto
- **Tempo:** prazo de execução de tarefas e para entregar partes do produto até a finalização do projeto
- **Custo:** valor financeiro empregado no projeto para pagar recursos e comprar materiais e software
- **Recursos:** cálculo de esforço previsto, alocação e gerenciamento do emprego de mão-de-obra no projeto
- **Qualidade:** padrões de trabalho e de inspeção, controle e garantia da qualidade do processo de trabalho e produto gerado

Modelos de processo de software



- **Risco:** identificação, análise de exposição, planejamento, execução e controle de ações de mitigação
- **Aquisições:** programação e controle de compras para atender o projeto e contratos de serviços terceirizados
- **Comunicação:** estratégias de comunicação permanente de projeto por público interessado, garantido transparência
- **Partes interessadas:** engajamento de interessados, stakeholders de projeto
- **Integração:** documentação e organização de ações que permitam a evolução de uma ou mais frentes de trabalho o do projeto em harmonia e sincronismo.



Modelos de processo de software

MODELO ADAPTATIVO



Processo adaptativo, segundo o FLEKS, é aquele que permite adequação de mudanças ao longo do projeto, não sendo necessário ter total clareza do produto completo a ser entregue, de antemão.

Esse modelo segue o manifesto ágil e aplica modelos gerenciais como o SCRUM (que estudaremos em detalhes mais adiante).



Modelos de processo de software

MODELO HÍBRIDO



Nesse modelo de processo, organiza-se o projeto misturando práticas mais burocráticas e rígidas do modelo Preditivo com práticas Adaptativas.

Na prática, a empresa cria um escritório de projeto com rigores de documentação de planos e acompanhamento e controles com métricas de sucesso (OKR-Objectives and Key Results), cuidando da comunicação com stakeholders, com a equipe e parceiros, administrando orçamentos e tempo, determinando a alocação das pessoas para realizar projetos, padronizando técnicas e ferramentas através de escritórios de projeto, deixando que cada projeto específico possa ser conduzido de forma adaptativa no seu dia a dia, na forma como arquiteta e adapta a solução e interage com donos de produtos e ecossistemas de negócio.

Modelos de processo de software

MODELO HÍBRIDO



O modelo híbrido ainda pode envolver a realização de parte do projeto de forma Clássica e parte de forma Ágil: a produção daquilo cujo escopo está bem definido e será imutável pode ser conduzido no modelo Preditivo; enquanto aquilo sobre o qual ainda se tem dúvidas de escopo ou sobre a forma de produzir, pode ser realizado sob modelo Adaptativo!

Modelos de processo de software

AS VERDADES DO PROJETO



Segundo o FLEKS, é necessário garantir em todos projetos o princípio #BEFLEX:



METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

Resumindo...

Na **linha Clássica ou Preditiva**, existe uma **preocupação grande com a formalização** de atividades e as etapas do projeto são bem definidas e **pouco flexíveis**. Em geral são **empregadas em situações de projetos com escopo fechado** ou situações de implantação de software que envolvem praticamente parametrizações (configurações) de instalação, **com baixa probabilidade de mudança** em escopo, prazo e custo inicialmente planejados.

Na **linha Ágil ou Adaptativa**, o **foco está na geração de um produto útil com muita rapidez**. Os **controles sobre o projeto são simplificados** para que o foco não se perca. **Aceita-se mudanças ao longo do projeto**, as quais envolverão **renegociação e replanejamento rápido**. **O escopo não precisa estar completamente confirmado para iniciar**.

METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

Métodos **Clássicos** em geral seguem os modelos de Ciclo de Vida **Cascata ou Incremental**.

Métodos **Ágeis** geralmente seguem o **Ciclo de Vida Espiral ou Prototipação**.

METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

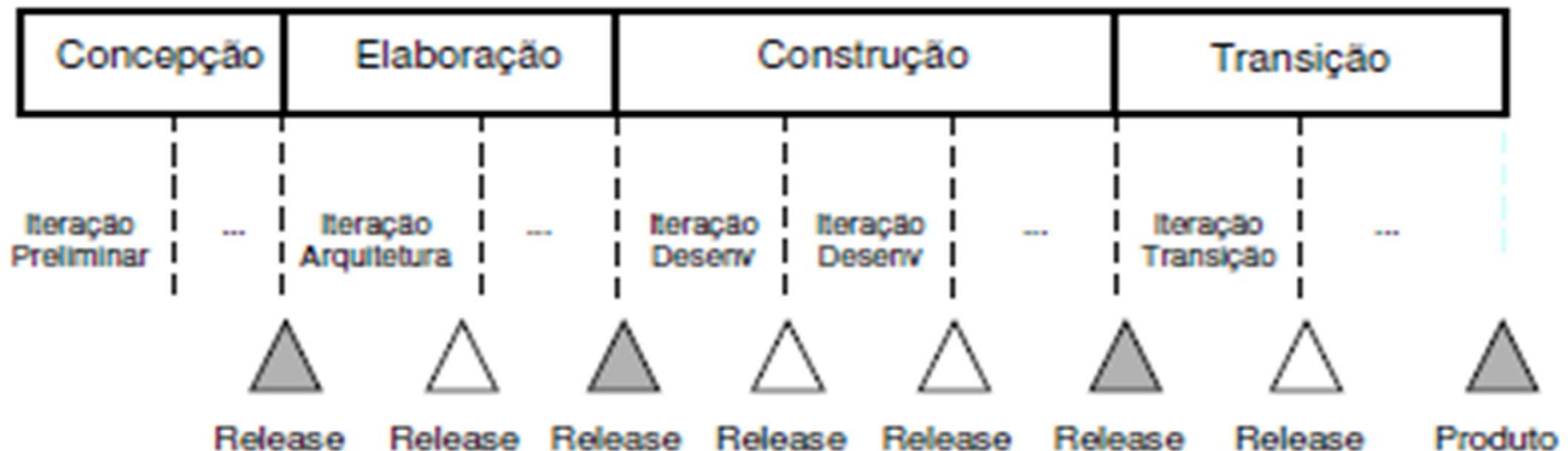
O método Clássico mais difundido é o RUP.



Visão geral do RUP (Rational Unified Process)

Fases

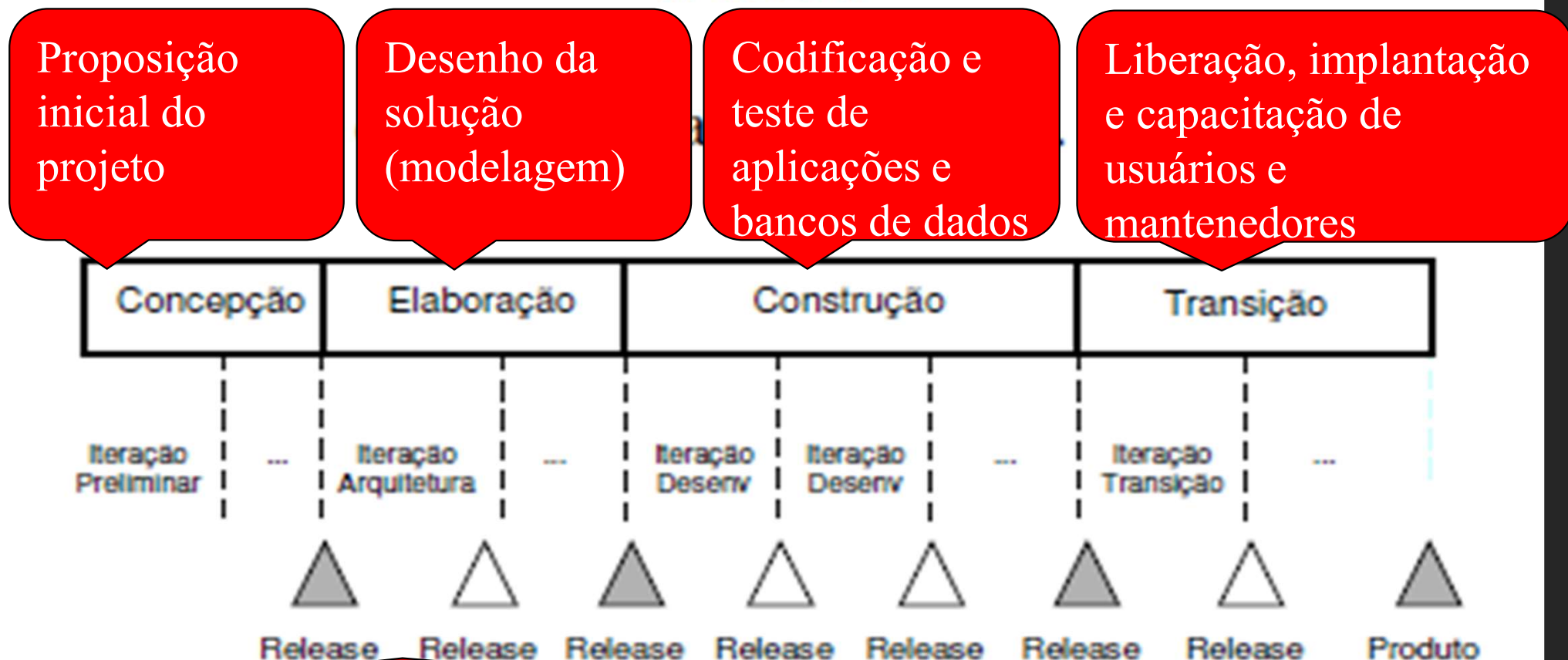
- Cada fase é subdividida em iterações.



- Um conjunto de artefatos (release) é gerado a cada iteração.
- Um milestone (marco) é gerado a cada fase.

Visão geral do RUP (Rational Unified Process)

Fases

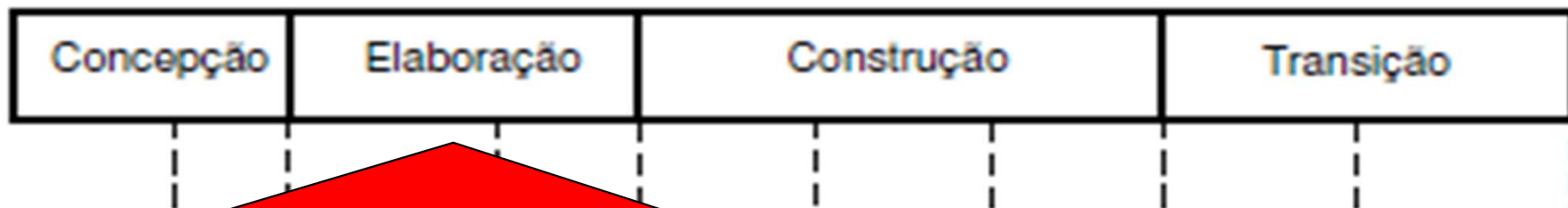


As interações representam frentes de trabalho que geram entregas (releases). Podem existir várias frentes de trabalho modelando partes do sistema, outras construindo módulos e componentes, outras testando e liberando, etc.

Visão geral do RUP (Rational Unified Process)

Fases

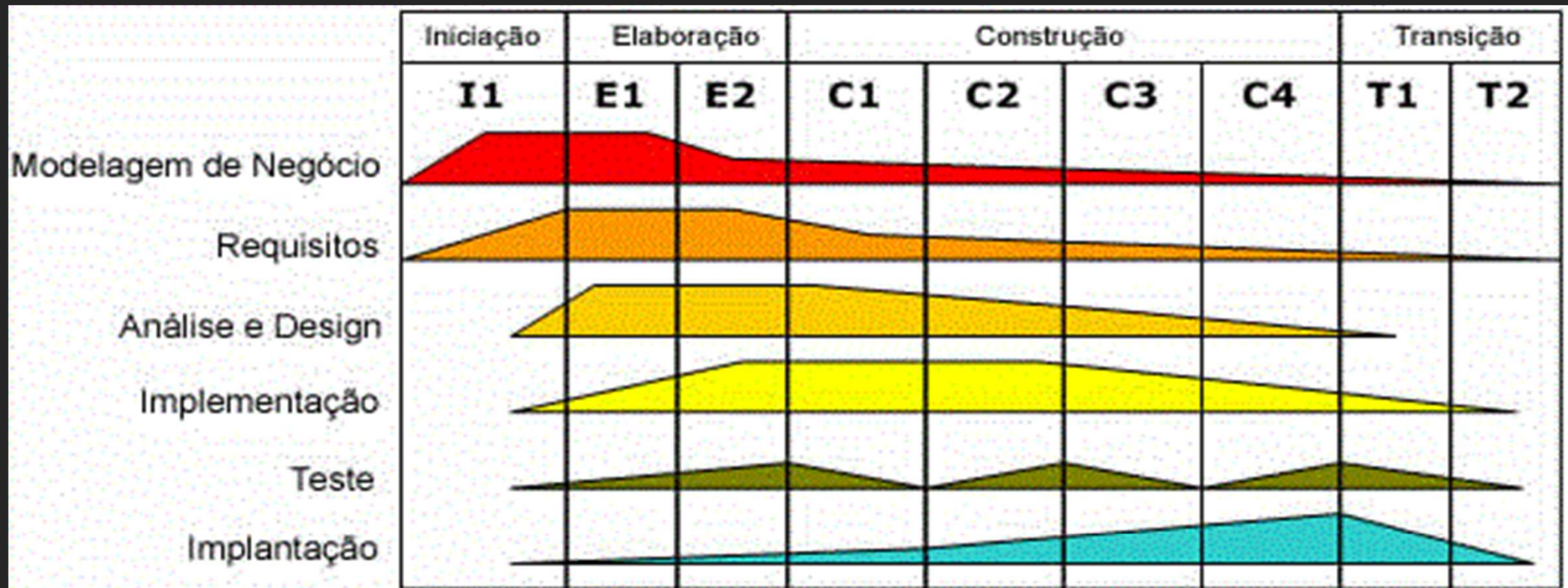
- Cada fase é subdividida em iterações.



Para cada fase, são definidos os “Workers”:

- Papel (exemplo – analista de sistemas, programador, usuário final, analista de testes, etc.);
- Responsabilidades em termos de atividades que deve cumprir e produtos (artefatos) que deve gerar (código, caso de teste, plano de projeto, etc.).

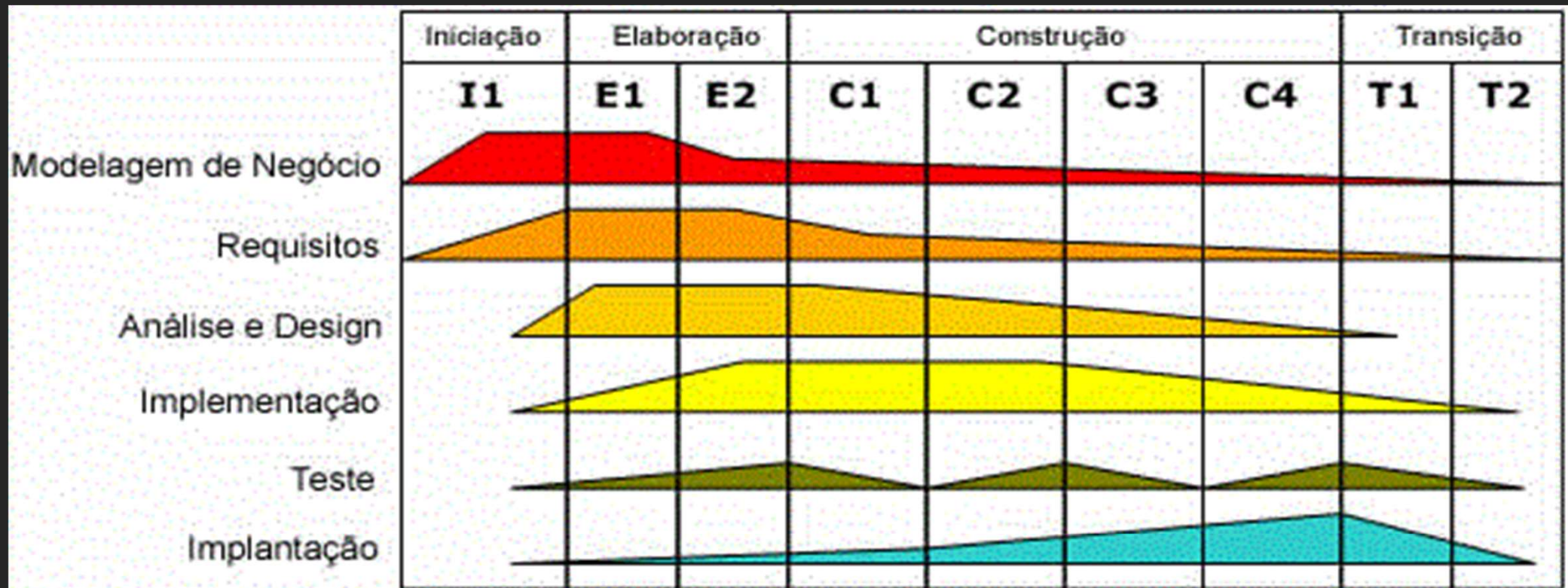
O RUP define uma previsão de distribuição de esforço para realizar os diversos tipos de atividades envolvidas no desenvolvimento do software, ao longo das fases do ciclo de vida (popularmente chamado de gráfico de baleias).



	<u>Inception</u>	<u>Elaboration</u>	<u>Construction</u>	<u>Transition</u>
Effort	~5 %	20 %	65 %	10%
Schedule	10 %	30 %	50 %	10%

Maiores informações sobre o RUP:

<https://sce.uhcl.edu/helm/rationalunifiedprocess/Ruphomepage.com>



	<u>Inception</u>	<u>Elaboration</u>	<u>Construction</u>	<u>Transition</u>
Effort	~5 %	20 %	65 %	10%
Schedule	10 %	30 %	50 %	10%

RUP

Define uma previsão de distribuição de esforço para realizar os diversos tipos de atividades envolvidas no desenvolvimento do software, ao longo das fases do ciclo de vida.

	<u>Inception</u>	<u>Elaboration</u>	<u>Construction</u>	<u>Transition</u>
Effort	5 %	20 %	65 %	10%
Schedule	10 %	30 %	50 %	10%

O RUP ainda apresenta descrições detalhadas de cada atividade a ser desenvolvida num projeto, com exemplificação de estruturas padrão (templates) de artefatos a serem produzidos a cada atividade realizada.

Ele traz bastante formalismo ao processo de desenvolvimento.

DEMONSTRAÇÃO DO RUP

<http://sce.uhcl.edu/helm/rationalunifiedprocess/>

Ruphomepage.com

O processo de software é usado como base para o planejamento e controle do ciclo de vida de produção de software!

Consulte o arquivo ProjetoReferencia-EngSW-RUP.mpp na sua área de apostilas!

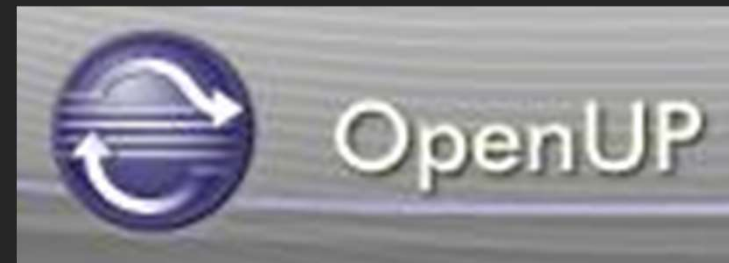
Ele demonstra um plano de projeto com base na abordagem RUP

METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

Métodos Ágeis mais difundidos:

RAD MODEL

**AGILE
METHODOLOGY**



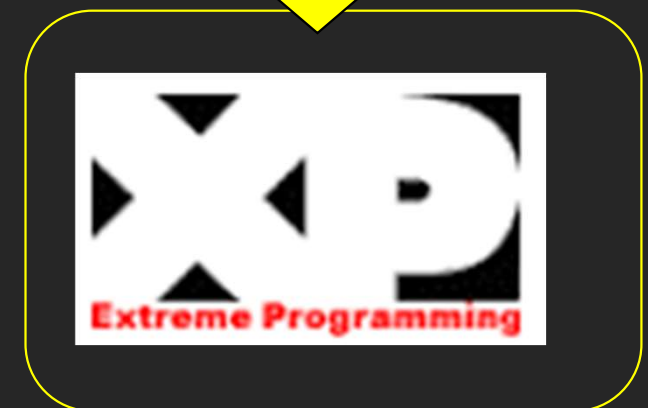
METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE



Métodos Ágeis mais difundidos:

1º padrão
internacional
de processo
ágil

Tornou-se um
padrão para o
mercado de Sw



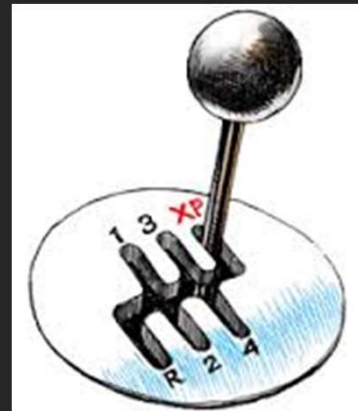
MÉTODO ÁGIL

Visão geral da
Metodologia Ágil

AGILE
METHODODOLOGY

PROJETO ÁGIL

Os projetos conduzidos sob uma **metodologia ágil** focam na **geração de valor para o cliente**, ou seja, procuram dedicar **maior atenção, tempo e capital na produção daquilo que o cliente vai usufruir**, sem dispendar recursos demasiados em controles acessórios.



MANIFESTO ÁGIL

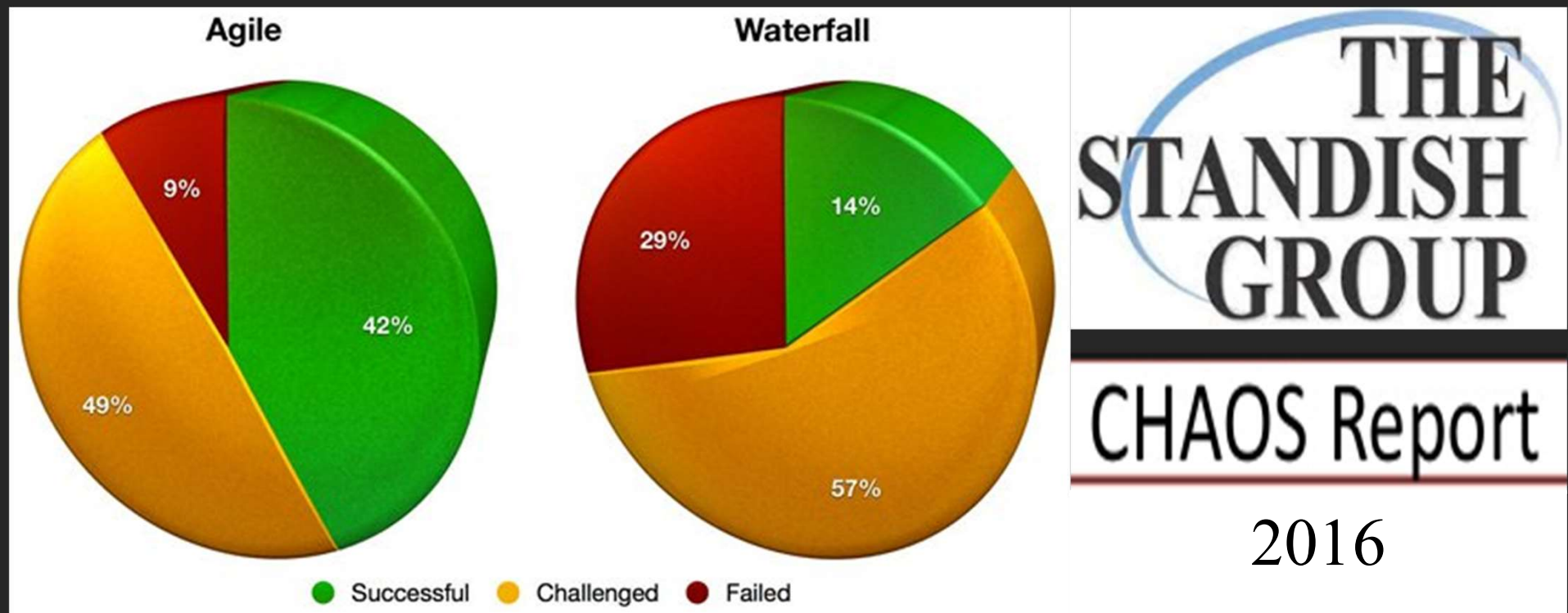
“Manifesto para o Desenvolvimento Ágil de Software”

- **Indivíduos e interações** acima de processos e ferramentas.
- **Software funcionando** acima de documentação abrangente.
- **Colaboração** com o cliente acima de negociação de contratos.
- **Responder a mudanças** acima de seguir um plano.

MÉTODOS ÁGEIS

Atualmente, os métodos ágeis ganham grande espaço no mercado por:

- Atenderem rapidamente as expectativas dos clientes;
- Focarem equipes de desenvolvimento naquilo que gostam de fazer
- Aumentarem as taxas de sucesso quanto a qualidade do projeto em termos de processo e produto



Valores do AGILE

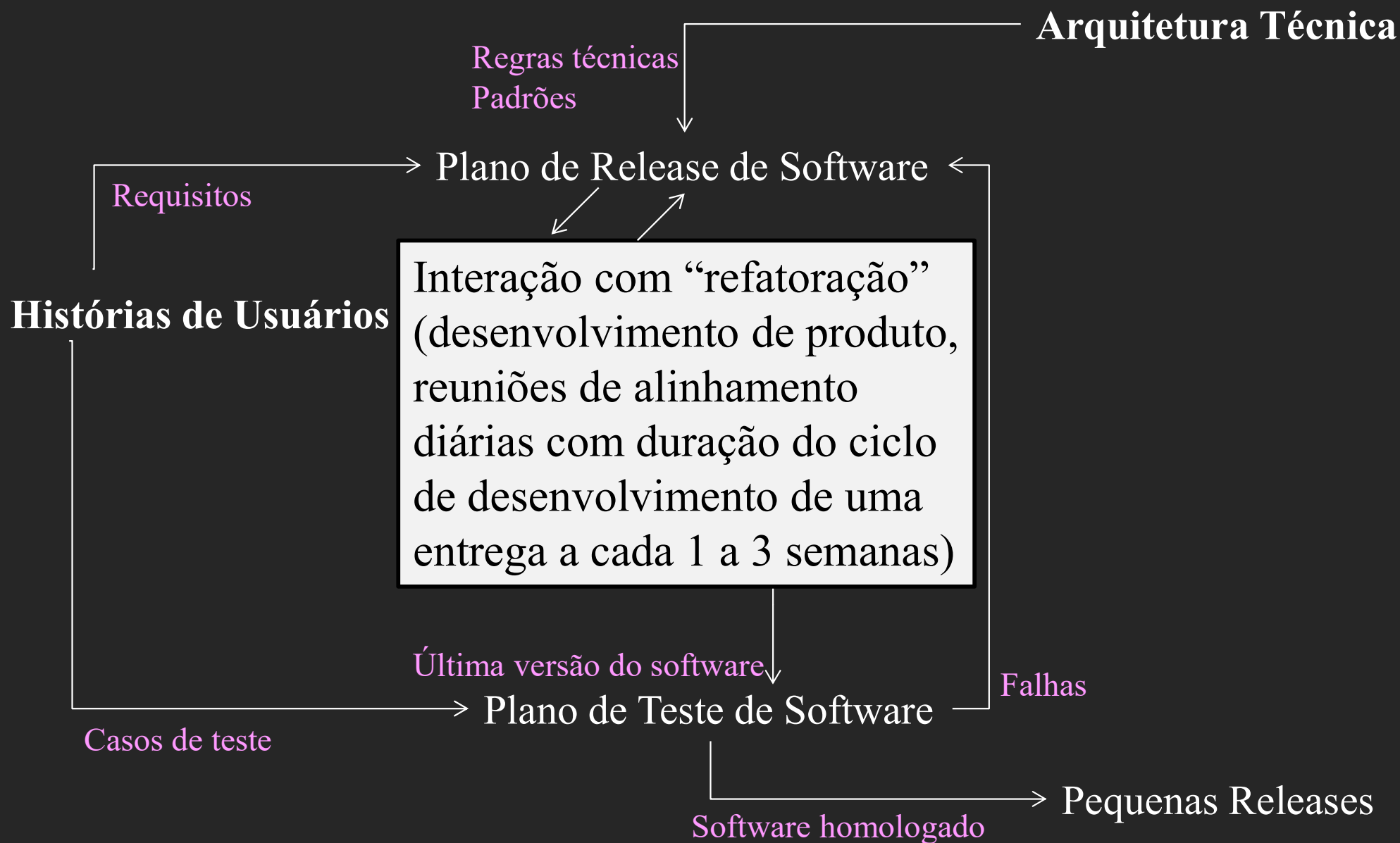
Comunicação: XP foca em construir um entendimento pessoa-a-pessoa do problema, com o uso mínimo de documentação formal e com o uso máximo de interação “cara -a-cara” entre as pessoas envolvidas no projeto. As práticas de XP como programação em pares, testes e comunicação com o cliente têm o objetivo de estimular a comunicação entre gerentes, programadores e clientes.

Simplicidade: XP sugere que cada membro da equipe adote a solução mais fácil que possa funcionar. O objetivo é fazer aquilo que é mais simples hoje e criar um ambiente em que o custo de mudanças no futuro seja baixo.

Feedback: Os programadores obtêm feedback sobre a lógica dos programas escrevendo e executando casos de teste. Os clientes obtêm feedback através dos testes funcionais criados para todas as histórias (casos de uso simplificados).

Coragem: Ela é necessária para que realmente se aplique XP como deve ser aplicado. Exemplos de atitude que exigem coragem são: alterar código já escrito e que está funcionando; jogar código fora e reescrever tudo de novo; e permitir código compartilhado por todos.

INTEGRAÇÃO DAS FASES DO NO DESENVOLVIMENTO ÁGIL



Escolha de um Processo de Software – SCRUM em detalhes

MÉTODO ÁGIL

Visão geral do SCRUM



MÉTODO ÁGIL

Estudaremos em detalhes o método mais difundido mundialmente para gerenciamento de projetos ágeis, o SCRUM!



*O SBOK é o livro que define o funcionamento do SCRUM

SCRUM

ATIVIDADE PRÁTICA



Pesquise na Internet alguns princípios do SCRUM e tente compreendê-los:

- *SCRUM TIME BOX*
- *SCRUM ARTIFACTS*
- *SCRUM TEAM ROLES*

MÉTODO ÁGIL

O SCRUM traz um conceito forte de trabalho em TIME para concretizar resultados rápidos, tendo clara definição de papéis e responsabilidades!

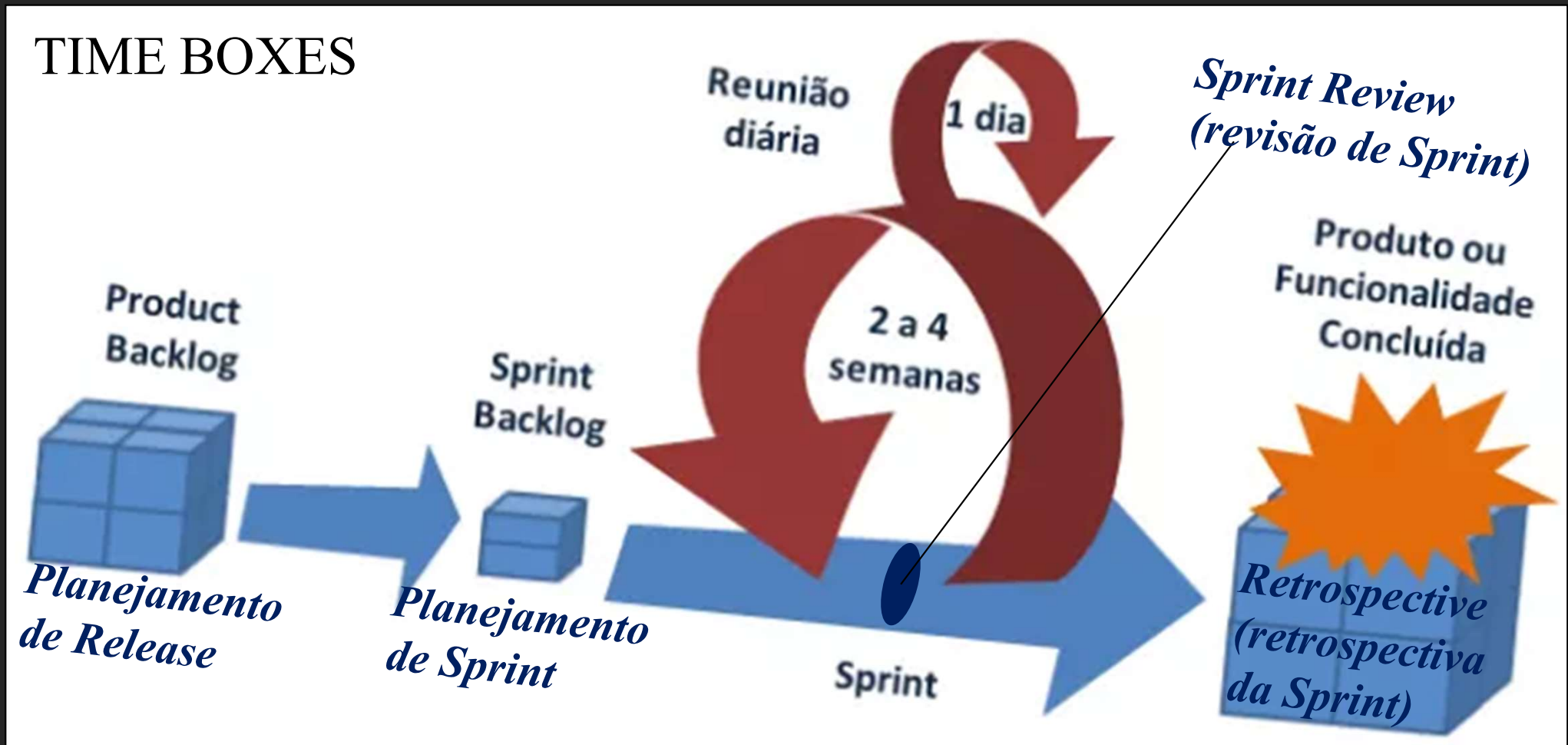
Ele é organizado em um processo definido em TIME BOXES, que são caixas de tempo pré-fixado, mantendo assim um ritmo de produção, tendo o compromisso de gerar frequentemente uma entrega útil para o cliente.

Ele inclui ainda, um conjunto de ARTEFATOS de gestão, que nada mais são que documentações gerenciais padronizadas a serem produzidas ao longo do ciclo de processo produtivo.

SCRUM

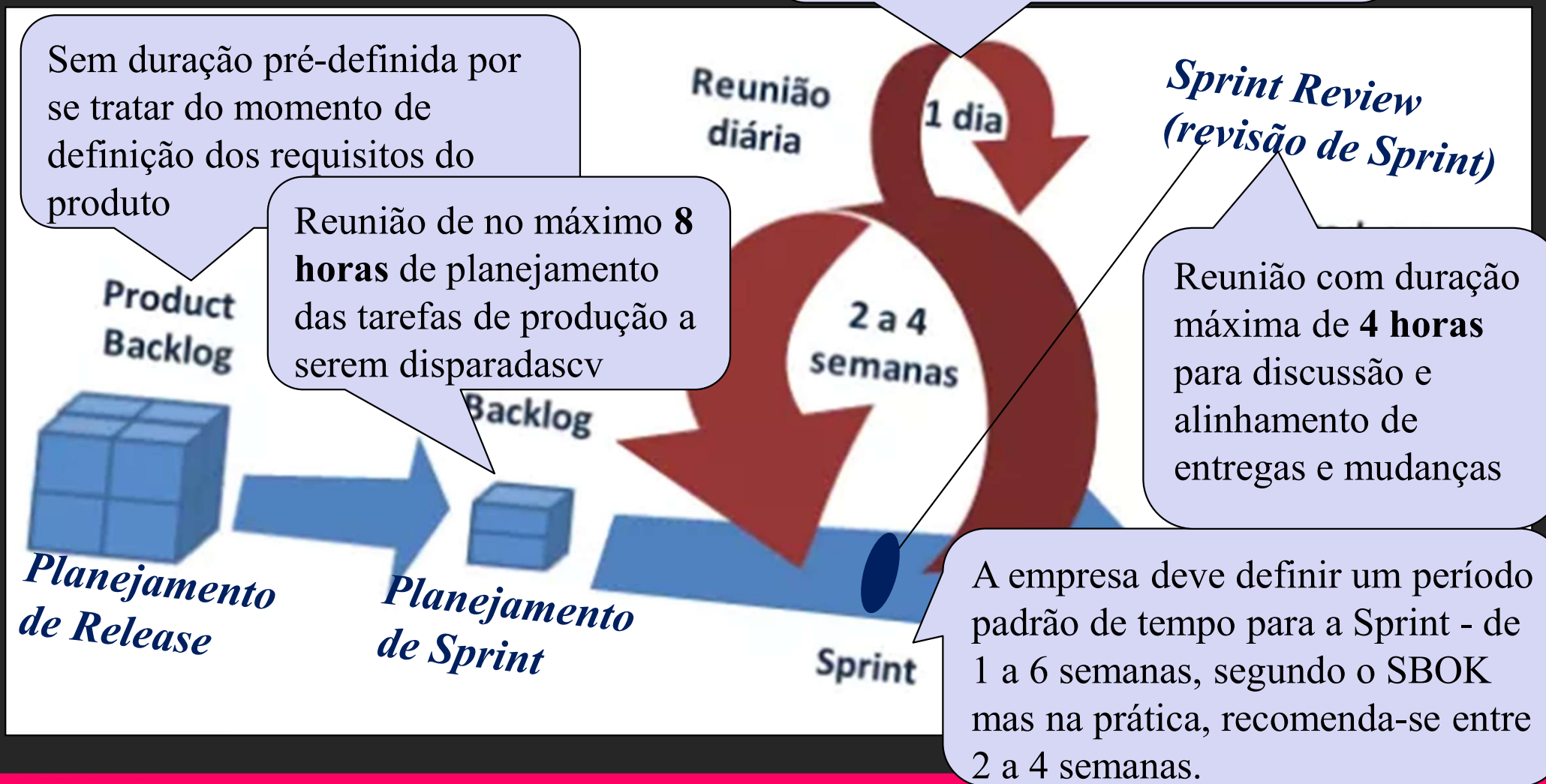
A figura a seguir ilustra o processo de produção de software com base na metodologia:

TIME BOXES



SCRUM

A figura a seguir ilustra o processo na metodologia:



SCRUM

A figura a seguir ilustra o processo de produção de software com base na metodologia:

TIME BOXES



SCRUM

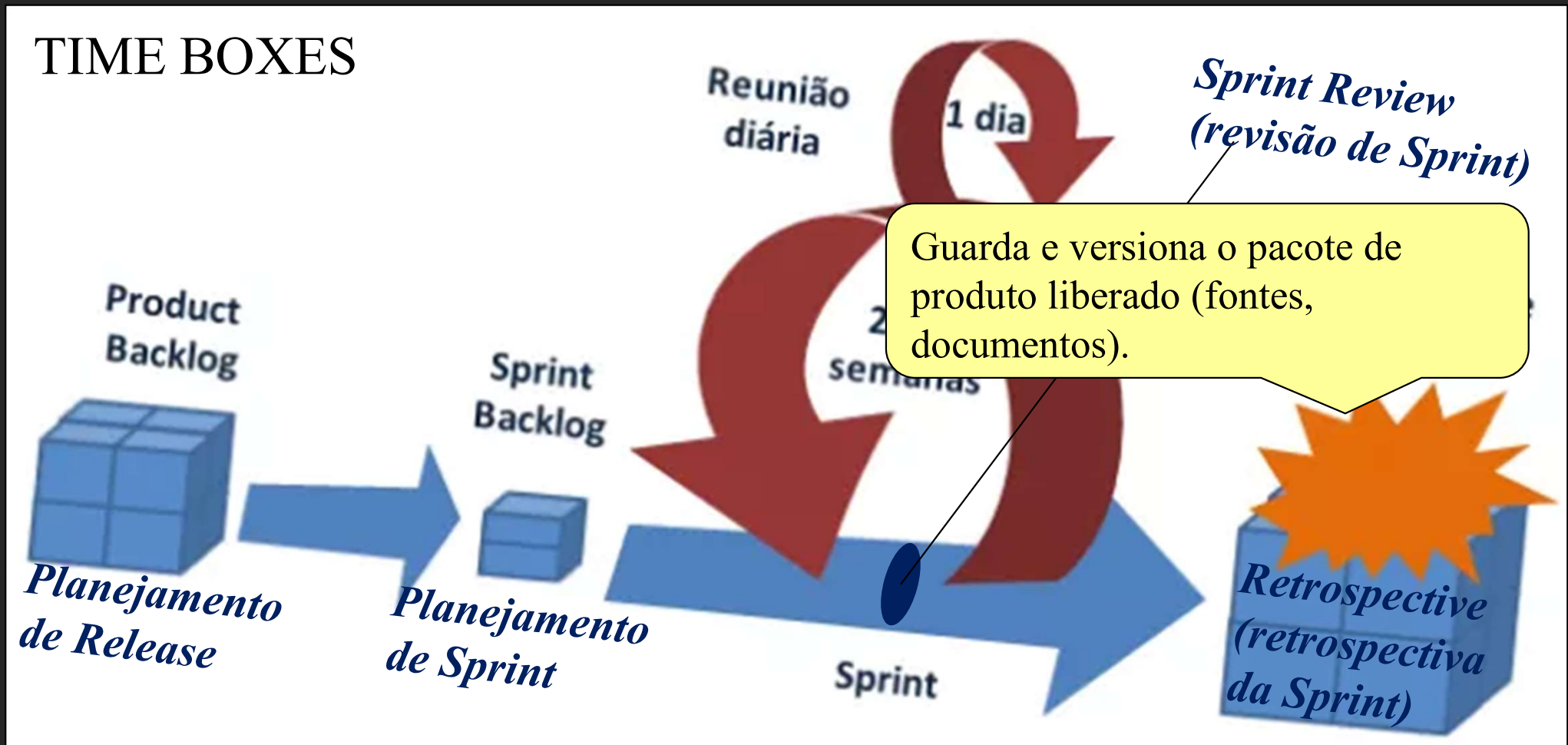
A figura a seguir ilustra o processo na metodologia:



SCRUM

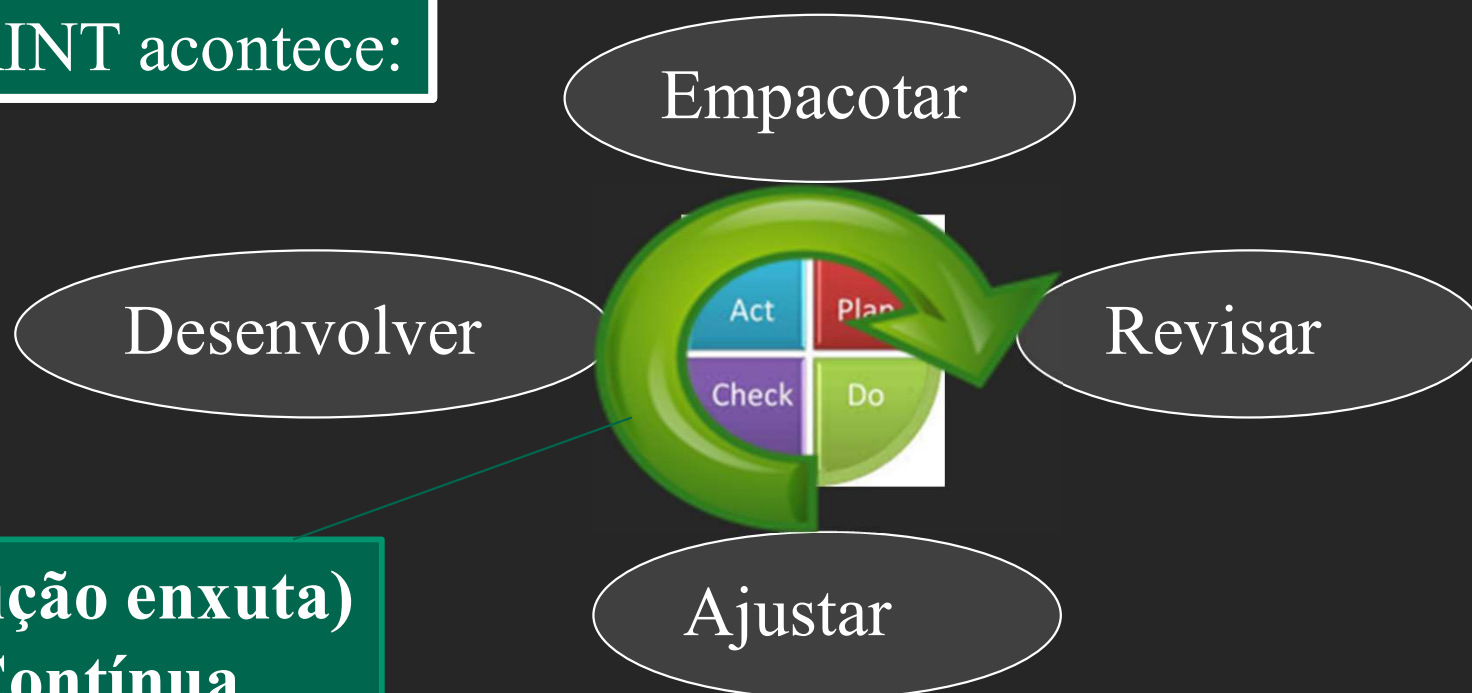
A figura a seguir ilustra o processo de produção de software com base na metodologia:

TIME BOXES



SCRUM

Dentro da SPRINT acontece:



LEAN (produção enxuta) & Melhoria Contínua

Busca da *Eliminação ou Minimização de Perda*, que é tudo aquilo em um processo *que não agrega valor*.

Por exemplo, o tempo que um documento está na mesa de alguém aguardando alguma ação ou a própria construção de um documento que não será usado em tomadas de decisão).

SCRUM Funções da equipe e responsabilidades

A equipe de um projeto SCRUM tem a seguinte distribuição de papéis e responsabilidades:

TIME SCRUM



Product Owner



SCRUM Master



SQUAD -Time de
Desenvolvimento

SCRUM Funções da equipe e responsabilidades

A equipe de um projeto SCRUM tem a seguinte distribuição de papéis e responsabilidades:

TIME SCRUM



Product Owner



SCRUM Master



Time de
Desenvolvimento

Define a necessidade de negócio sendo um canal entre stakeholders de negócio e o Time de desenv.

Capacita e avalia a aplicação da metodologia SCRUM

Produz a solução e gerencia o projeto

SCRUM Funções da equipe e responsabilidades

A equipe de um projeto SCRUM tem a seguinte distribuição de papéis e responsabilidades:



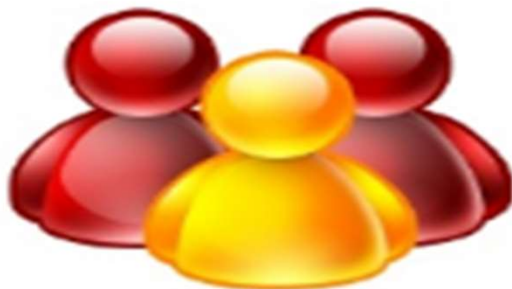
Product Owner (PO)

Responsável por garantir o ROI (Retorno de Investimento)
Responsável por conhecer as necessidades do(s) cliente(s)
(único por produto a entregar)



ScrumMaster (SM)

Responsável por remover os impedimentos do time
Responsável por garantir o uso de Scrum
Protege o time de interferências externas
(único por time Scrum, podendo ser compartilhado com outros times)



Time

Definir metas das iterações
Auto-gerenciamento
Produzir produto com qualidade e valor para o cliente

SCRUM Funções da equipe e responsabilidades

A equipe de um projeto SCRUM tem a seguinte distribuição de papéis e responsabilidades (segundo o SBOK):



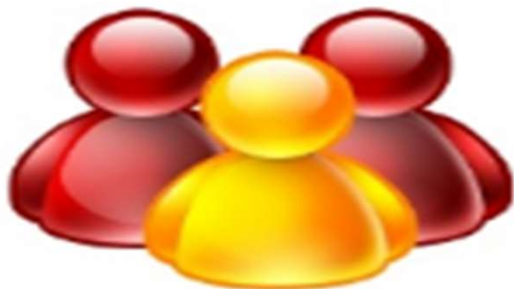
PO: Único por produto a entregar

Se se tratar de um projeto complexo ou um programa com vários projetos, existindo vários produtos/soluções de negócio a entregar, existirá um PO por frente de solução



MASTER: Único por time Scrum

Dependendo da característica de demanda dos projetos, o Scrum Master pode ser compartilhado entre mais de um time/frente de desenvolvimento de solução.



TIME: Composto pode várias pessoas

Os profissionais do Time devem ter múltiplas e complementares competências para lidar com todas as tarefas de desenvolvimento (gestão de projeto, modelagem e sistema, construção, teste, etc.)

SCRUM Funções da equipe e responsabilidades

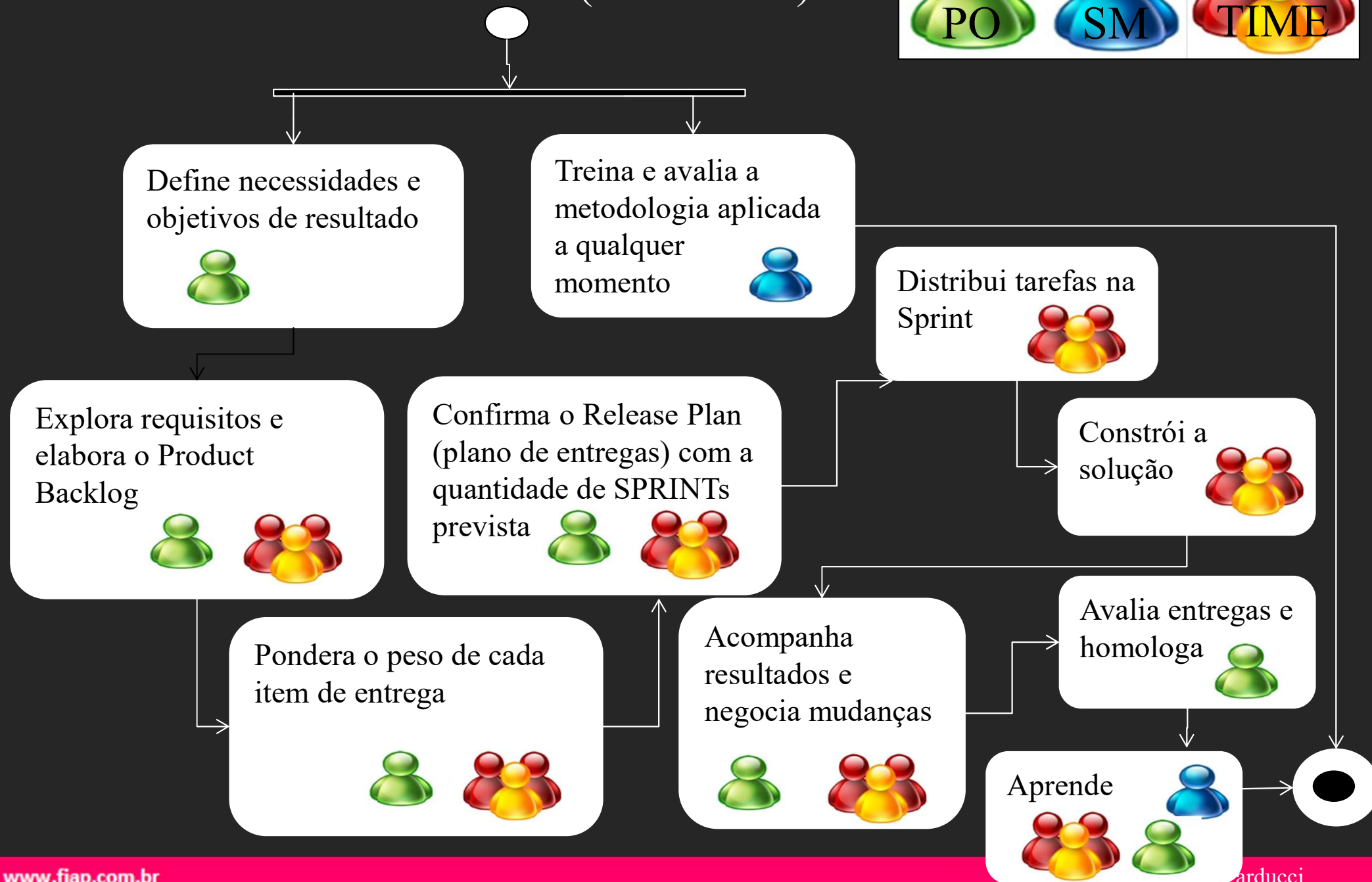
Durante o projeto, ocorrerão Cerimônias, que são encontros entre as partes interessadas do projeto para realizar planejamentos ou avaliações de resultados:

(PO)

- Planejamento de produto
- Planejamento de Sprint
- Reunião de avaliação de status de projeto
- Revisão de plano e entregas
- Retrospectiva de avaliação de lições aprendidas na condução do projeto



SCRUM Fluxo de trabalho (cerimônias)

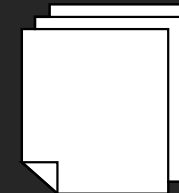


SCRUM Artefatos

A aplicação do SCRUM demanda os seguintes materiais para documentação de apoio ao gerenciamento:

ID	NOME	DATA DA TAREFA DE INÍCIO	NOME DA TAREFA	IMPORTE	ESTIMATIVA INICIAL	COMO DOCUMENTAR	NOTAS	PREVIDÊNCIA	NÚMERO DA PRÓXIMA TAREFA REALIZADA	RESPONSÁVEL
P000-001	Gestão de cadastro de cliente	p000-001-001	Incluir cliente	Alta	1 ponto	Tela de manutenção		p000-001-005	Sprint 1	João
		p000-001-002	Alterar dados do cliente	Alta	1 ponto	Tela de manutenção		p000-001-005	Sprint 1	Rita
		p000-001-003	Consultar cadastro de cliente	Alta	1 ponto	Tela de manutenção		p000-001-005	Sprint 1	Maria
		p000-001-004	Deletar cliente	Alta	1 ponto	Tela de manutenção	Deletar e não excluir clientes	p000-001-005	Sprint 1	João
		p000-001-005	Car tabela de cliente	Alta	1 ponto	Recuperar dados guardados			Sprint 1	Pedro
P000-002	Consulta de vendas por cliente	p000-002-001	Total de compras por cliente	Média	3 pontos	Cabecalho consulta no BI			Sprint 2	João
		p000-002-002	Cliente e Local e Total comprado	Média	5 pontos	Cabecalho consulta no BI			Sprint 2	Pedro
		p000-002-003	Cliente e Região Geográfica e Total	Média	8 pontos	Cabecalho consulta no BI			Sprint 2	João

Product backlog

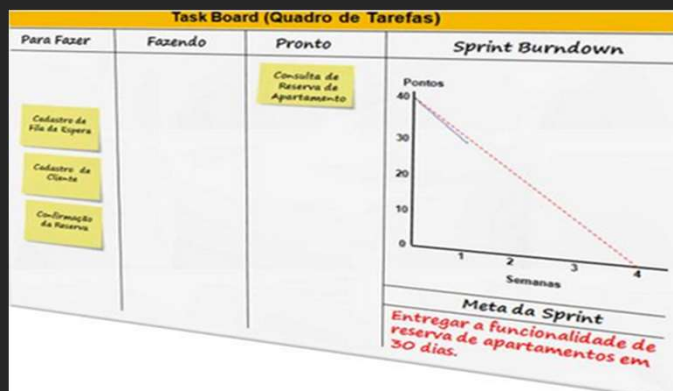


*Declaração
para compor
uma visão de
produto*

FICHA DE TRABALHO	
ID DA ATIVIDADE/TAREFA:	
BREVE DESCRIÇÃO (NOME DA ATIVIDADE/TAREFA):	
RESPONSÁVEL POR REALIZAR A TAREFA:	
NÚMERO DE PONTOS (PESO ESTIMADO) ATRIBUÍDO:	
TEMPO ESTIMADO, ASSOCIADO AOS PONTOS:	
TEMPO REALIZADO ATÉ O MOMENTO (CASO A TAREFA NÃO TENHA SIDO CONCLUÍDA NO MOMENTO DO REPORTE):	
TEMPO TOTAL DECORRIDO ATÉ A FINALIZAÇÃO:	
DATA ESTIMADA DE INÍCIO:	
DATA REAL DE INÍCIO:	
DATA ESTIMADA DE FINALIZAÇÃO:	
DATA REAL DE FINALIZAÇÃO:	

Sprint backlog

(task cards – cartões de trabalho)



Quadro de tarefas

(KANBAN Board e Sprint BurnDown Graph)

Gestão de Grandes Projetos somente com Framework SCRUM

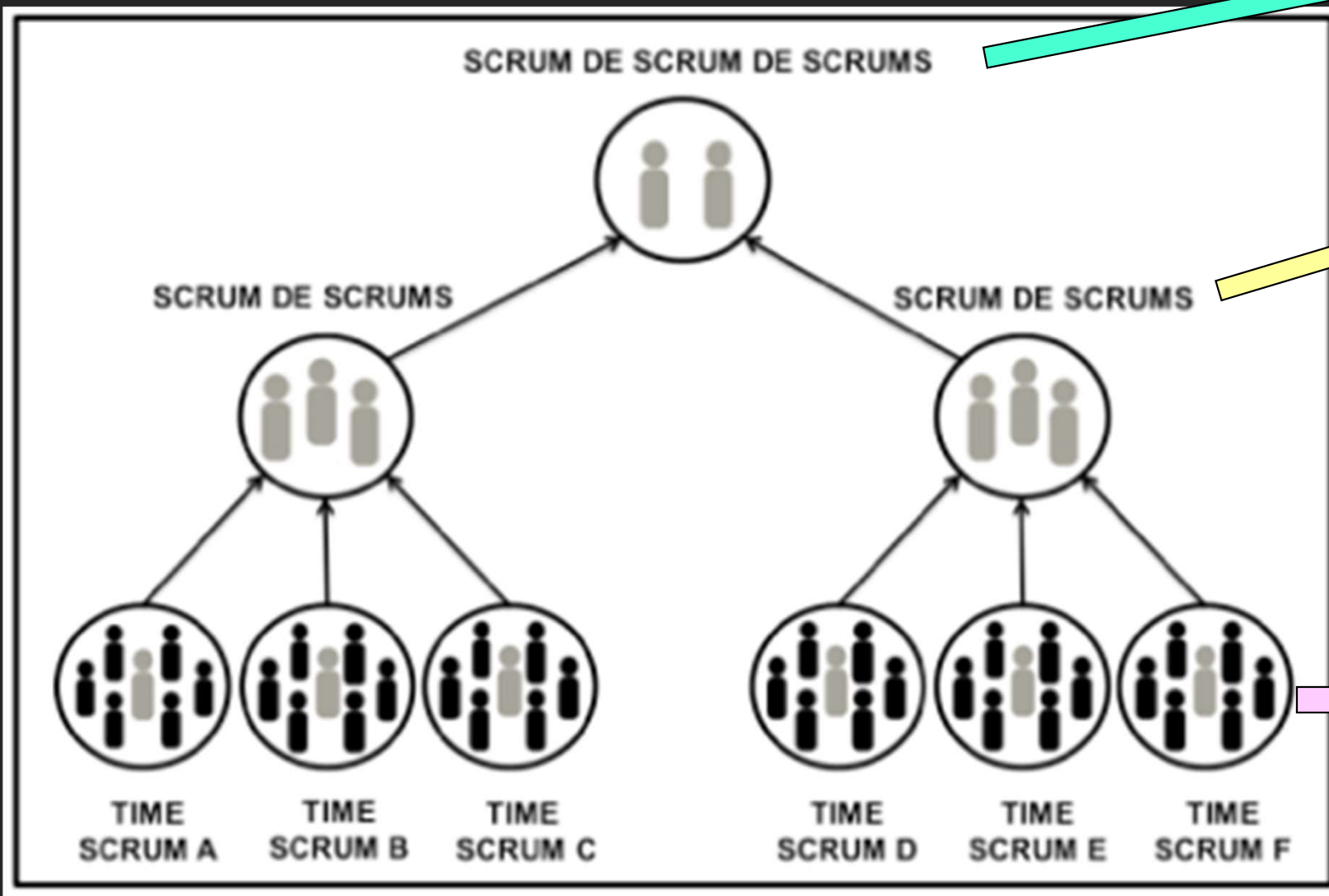
Em programas grandes como o desenvolvimento de um ERP que informatiza todas as atividades de uma empresa, podem ser criadas frentes de trabalho com um PO de cada frente/assunto empresarial tratado, um Scrum master (dedicado ou não, sendo recomendado estar dedicado ao time Scrum em um grande projeto) e uma equipe (time) de desenvolvimento para cada frente.

Quando existe integração entre as frentes de trabalho (dependências entre o que cada uma produz), pode ser adotado SCRUM de SCRUM, ou seja, existirá uma gestão do Backlog geral do sistema, o Kanban geral de entrega de produto e ocorrerão reuniões de acompanhamento, revisão e aprendizado que unirão os vários times SCRUM, criando uma visão unificada sobre o plano e resultados do projeto do sistema integrado.

**Abordagem prevista no SBOK.*

Gestão de Grandes Projetos somente com Framework SCRUM

SCRUM de SCRUM...



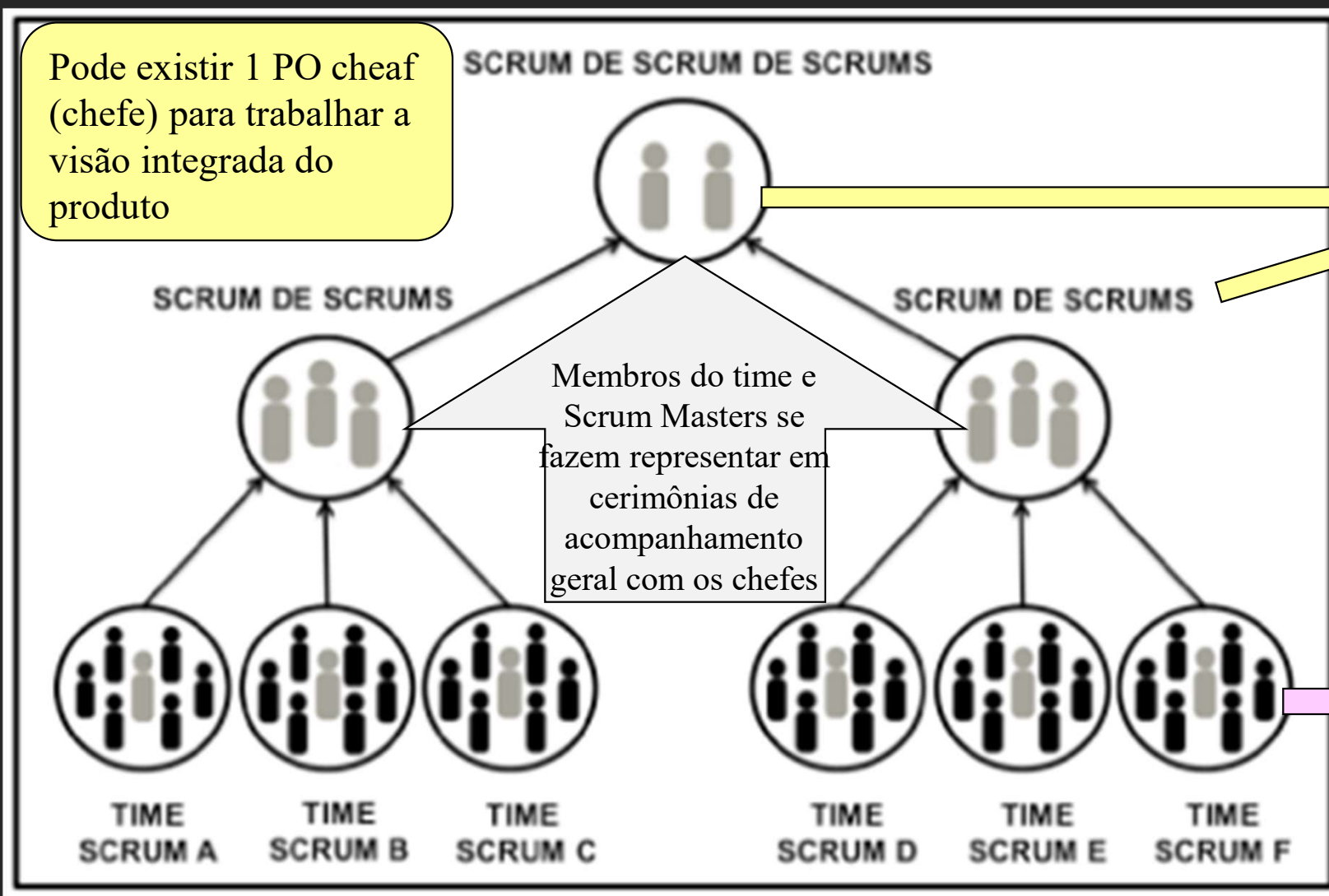
Gestão do Sistema ERP

Gestão dos Módulos RH e Estoques, por exemplo

Gestão do produto Folha de Pagamento e Contratação do RH; e Recebimento e expedição de Mercadorias do Estoque

Gestão de Grandes Projetos somente com Framework SCRUM

SCRUM de SCRUM (reuniões SOS)...



Pode existir 1 PO cheaf (chefe) para trabalhar a visão integrada do produto

1 Scrum Master Chefe e 1 representante de Time (opcional)são designados p/ orquestrar o programa/

Temos 1 Scrum Master e 1 Product Owner para cada produto a desenvolver (dependendo da demanda, o Scrum master pode cuidar de mais de um time)

Gestão de Grandes Projetos somente com Framework SCRUM

Entenda...

Podem existir vários POs em uma empresa, pois, claro, essa empresa tem inúmeros projetos que vão gerar produtos com razoável ou total independência.

Em um único produto a ser gerado só pode existir um PO (chefe) para que exista um único direcionamento de escopo e prioridades.

O Scrum Master preferencialmente deve se dedicar a uma frente de produto e a um Time Scrum, porém, quando a demanda por seus serviços é baixa, ele pode apoiar mais de um projeto de produto. Em grandes projetos pode existir um Scrum Master chief (chefe).

O Time é multifuncional e auto gerenciado, composto por um grupo pequeno de pessoas (não se aplica usar times com mais de 9 pessoas) dedicado a um produto a desenvolver.

Gestão de Grandes Projetos somente com Framework SCRUM

Entenda...

Imagine que tenhamos um programa de desenvolvimento de soluções corporativas de uma multinacional, como ERP transacional, Datawarehouse, Analytics, sendo que cada projeto é regionalizado (trata as características de um país específico).

Nesse caso, podemos ter 1 PO e 1 Scrum Master para cada frente específica (1 para o ERP, 1 para Analytics, etc para cada região operacional).

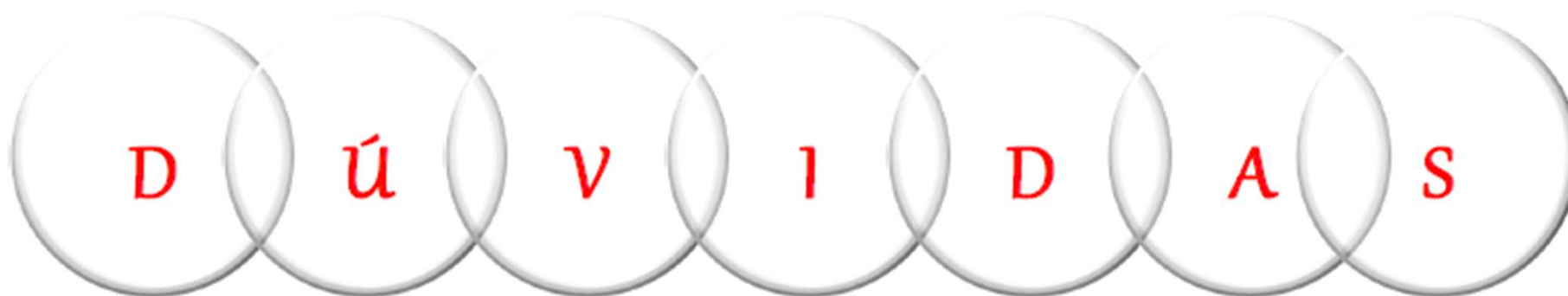
Se for aplicado o Scrum de Scrums, isso ocorrerá em cada uma das frentes (Scrum de Scrum do ERP Brasil, por exemplo) e não misturando as frentes e os POs.

Gestão de Grandes Projetos somente com Framework SCRUM

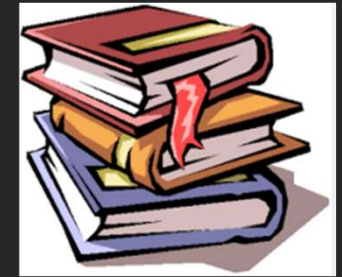
Ao longo do tempo, foram sendo desenvolvidas Frameworks (estruturas de trabalho) derivadas do Scrum original, com a finalidade de resolver os desafios da gestão ágil em grandes projetos.

Cada framework trata de maneira diferente a organização das equipes e a execução das cerimônias entre os participantes – na sua empresa, um desses frameworks deve ser adotado:

- *Scrum of Scrums – múltiplos POs para tratar os múltiplos times com uso de chiefs*
- *Nexus – PO único para manter visão integrada de projeto com cerimônias Nexus*
- *Less – Scrum máster compartilhado, com PO único e sem reuniões de integração*
- *SAFe – Trabalha com Temas e Épicas com Epic Owners (EO) e Product Managers, além dos Pos. Adiciona um arquiteto de solução corporativo*



Referência bibliográficas



BIBLIOGRAFIA :

- COSTA, Hélio. FLEKS – Guia completo. www.flexmodel.com, 2023.
- RUSCHEL, Ana Alice de Lucas. Gerência de projetos alinhada ao PMI. PMBoK 7ª edição. Editora SENAC, 2023.
- SURJAN, Jakov Trofo - Stelutti, Danilo - Fonseca, Douglas Henrique da - Oliveira, José Fabio Saraiva de. Sistema de gestão de projetos baseado na metodologia ágil scrum , 2011.
- **PRESSMAN**, Roger S.. Engenharia de software. - Uma abordagem profissional, 7ª edição. São Paulo, AMGH.
- **SOMMERVILLE**. Engenharia de software. Pearson.

Material de aula estará no site após a aula.

BONS ESTUDOS!

Bibliografia

- Guide to the Scrum Body of Knowledge (SBOK™ Guide), 2013.
- KAWAMOTO Júnior, Luiz Teruo - Gimenés Jr., Carlos - Pinho, Luciane Cristina Magalhães, Análise das boas práticas de gerenciamento de projetos no processo de desenvolvimento de software utilizando metodologia PMI e SCRUM em uma empresa multinacional, 2010.
- SURJAN, Jakov Trofo - Stelutti, Danilo - Fonseca, Douglas Henrique da - Oliveira, José Fabio Saraiva de. Sistema de gestão de projetos baseado na metodologia ágil scrum , 2011.
- VARGAS, Ricardo Viana. Gerenciamento de projetos: estabelecendo diferenciais competitivos. 7. ed Rio de Janeiro Brasport, 2010.