FIAP GRADUAÇÃO

Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

SOFTWARE DESIGN & TX

DISCIPLINA:
SOFTWARE DESIGN & TOTAL EXPERIENCE

AULA:
2-CICLOS DE VIDA APLICADOS AOS PROJETOS DE SOFTWARE

PROFESSOR:
RENATO JARDIM PARDUCCI

PROFRENATO.PARDUCCI@FIAP.COM.BR

AGENDA DA AULA

ENGENHARIA DE SOFTWARE:
- ✓ Conceito: Desenho de software e Engenharia
- ✓ Importância da sua prática para a indústria de software

CICLO DA VIDA DO SOFTWARE:
- ✓ Ciclo de vida do produto de software em uso
- ✓ Importância da administração do ciclo de vida para a indústria de software

GERENCIAMENTO DE PROJETOS:
- ✓ Processos clássicos
- ✓ Processos ágeis
- ✓ Áreas dos conhecimento do gerenciamento de projetos

INTRODUÇÃO À ENGENHARIA DE SOFTWARE

Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

O princípio da Engenharia de Software

A engenharia de software aplica modelos formais precisos que permitam abstrair e documentar requerimentos de software e que permitam ao engenheiro analisar, especificar, projetar, implementar e manter sistemas de software, avaliando e garantindo suas qualidades e possibilitando a capacitação para o uso do software desenvolvido, bem como garantindo as condições técnicas e econômicas que permitam a evolução e ajuste do software.

A engenharia de software oferece os métodos para que aconteça o gerenciamento adequado do ciclo de vida de software sob uma estrutura de processos definidos que garantam a formalização documental do software e que apliquem métodos e ferramentas que possibilitem a produção de software de forma sistemática, repetitiva e em escala.

A engenharia de software é portanto "O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais." (Fritz Bauer).

TOTAL EXPERIENCE

A Engenharia de Software vai auxiliar com suas práticas para que se alcance a TOTAL EXPERIENCE que consiste em:

- Tratar a jornada do consumidor/cliente e do colaborador da empresa que trabalha na cadeia de valor para o cliente (Customer + Employee + User + Multi experience);
- Criar facilidades e reduzir o esforço necessário para que o consumidor/cliente realize suas atividades;
- Gerar benefício de troca de experiência com facilidades de comunicação entre as pessoas para otimizar trabalhos e desenvolver competências;
- Como consequência, reduzir a rotatividade de profissionais talentosos (perda deles para outras empresas) e melhorar a qualidade de vida e de trabalho.

A GESTÃO DA INFORMAÇÃO É O PILAR PARA QUE SE CONHEÇAM COMPORTAMENTOS E INTERESSES DAS PESSOAS E SEJA POSSÍVEL DESENVOLVER SOLUÇÕES INTELIGENTES E ADERENTES!

SOFTWARE DESIGN & TX

O desafio da Engenharia de Software é **garantir padrões de qualidade no produto final através de um processo formal** que canalize a CRIATIVIDADE humana, criando algo CONCRETO, FUNCIONAL e ÚTIL.

Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

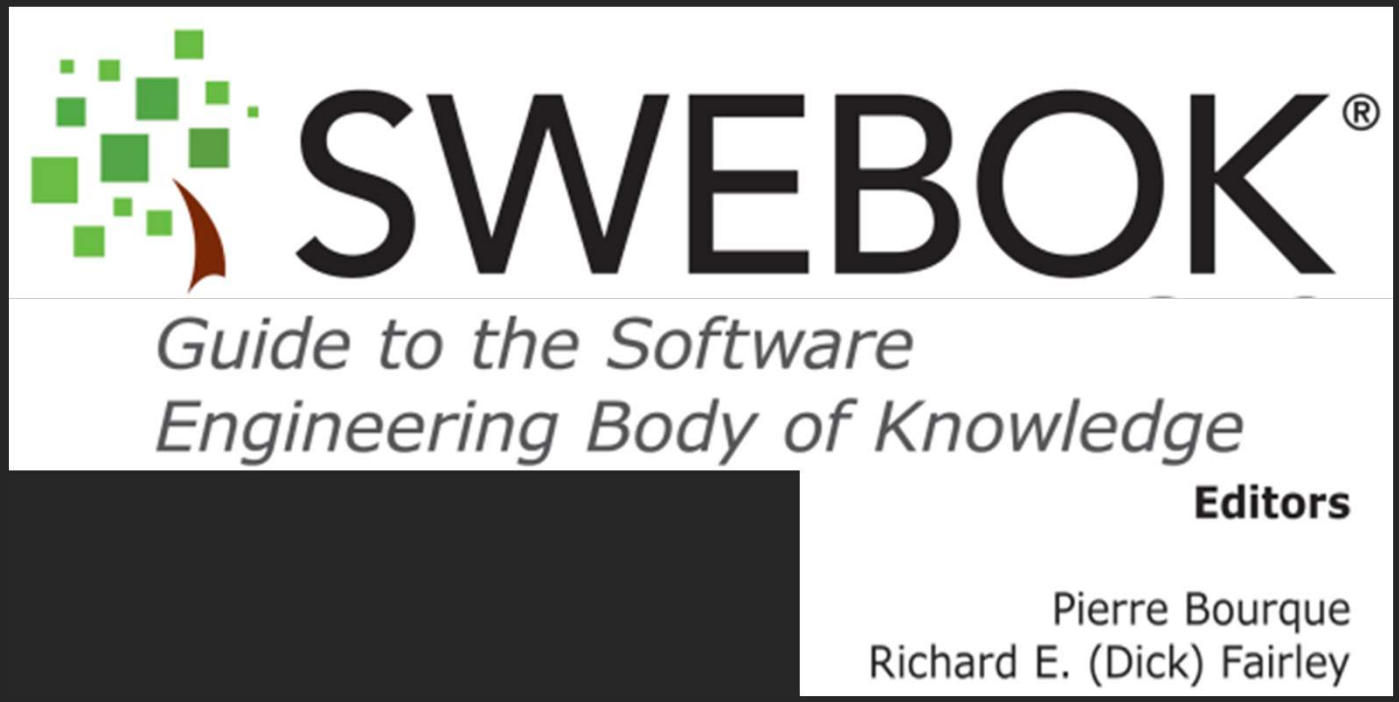
Todas as empresas consagradas que você conhece no ramo de tecnologia utilizam engenharia de software para desenvolver seu produtos!

**Ainda existem exemplos de sistemas sofisticados de automação que operam em missão crítica (não podem parar de funcionar) como os hospitalares, controle de aeronaves, pesquisa espacial, entre outros.*

A adoção de **métodos formais que organizam a produção do software** em suas diversas etapas do processo de manufatura é que **permitem elevar os patamares de qualidade e trazer retorno sobre o investimento!**

A IEEE (Institute of Electrical and Electronic Engineers**) publica o SWEBOK (**Software Engineering Body of Knowledge**) – um guia de práticas de engenharia de software reconhecidas internacionalmente.**

Esse material está disponível livremente na Internet.



Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

A VIDA DE UM SOFTWARE

APRENDENDO NA PRÁTICA

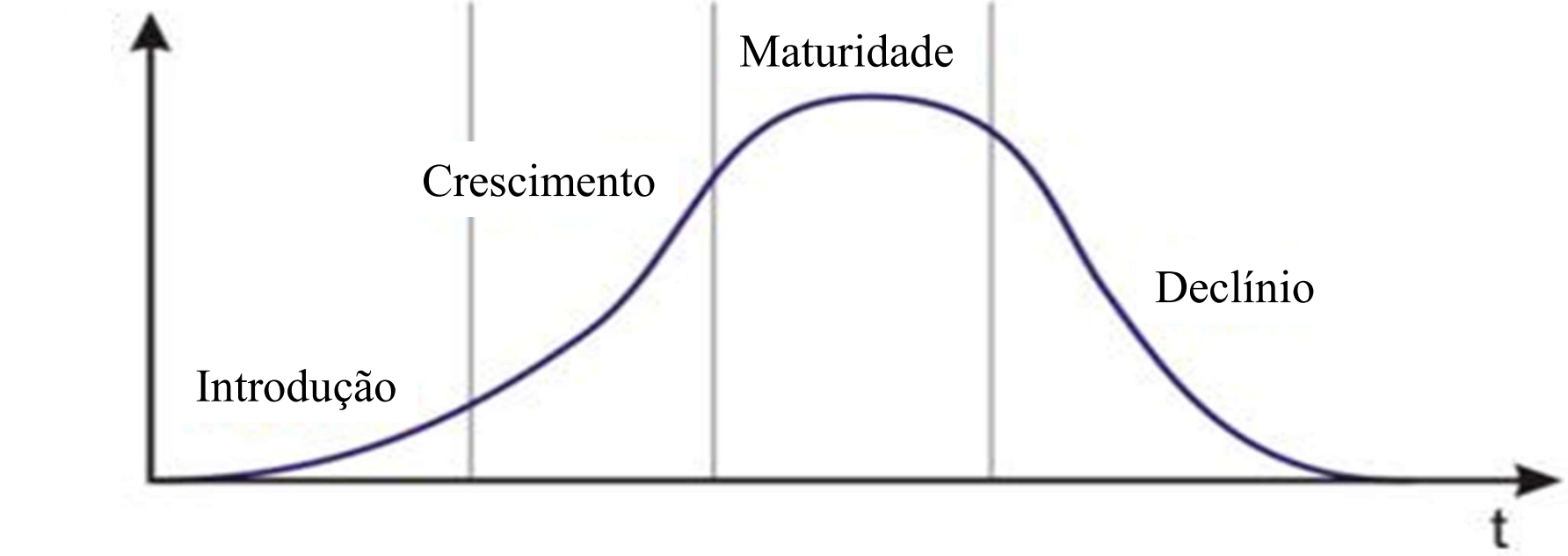
Compreendido em linhas gerais que será aplicada a Engenharia de Software, utilizando profissionais de TI de várias especialidades para produzir um sistema que é composto de Hardware, Aplicação, Dados e Processos e Pessoas preparados para operá-lo, o proprietário quer saber se vai valer a pena investir um capital que não deve ser menor que R$ 100.000,00 no novo sistema.

Ele quer saber se uma vez feito esse investimento, nunca mais será necessário investir.

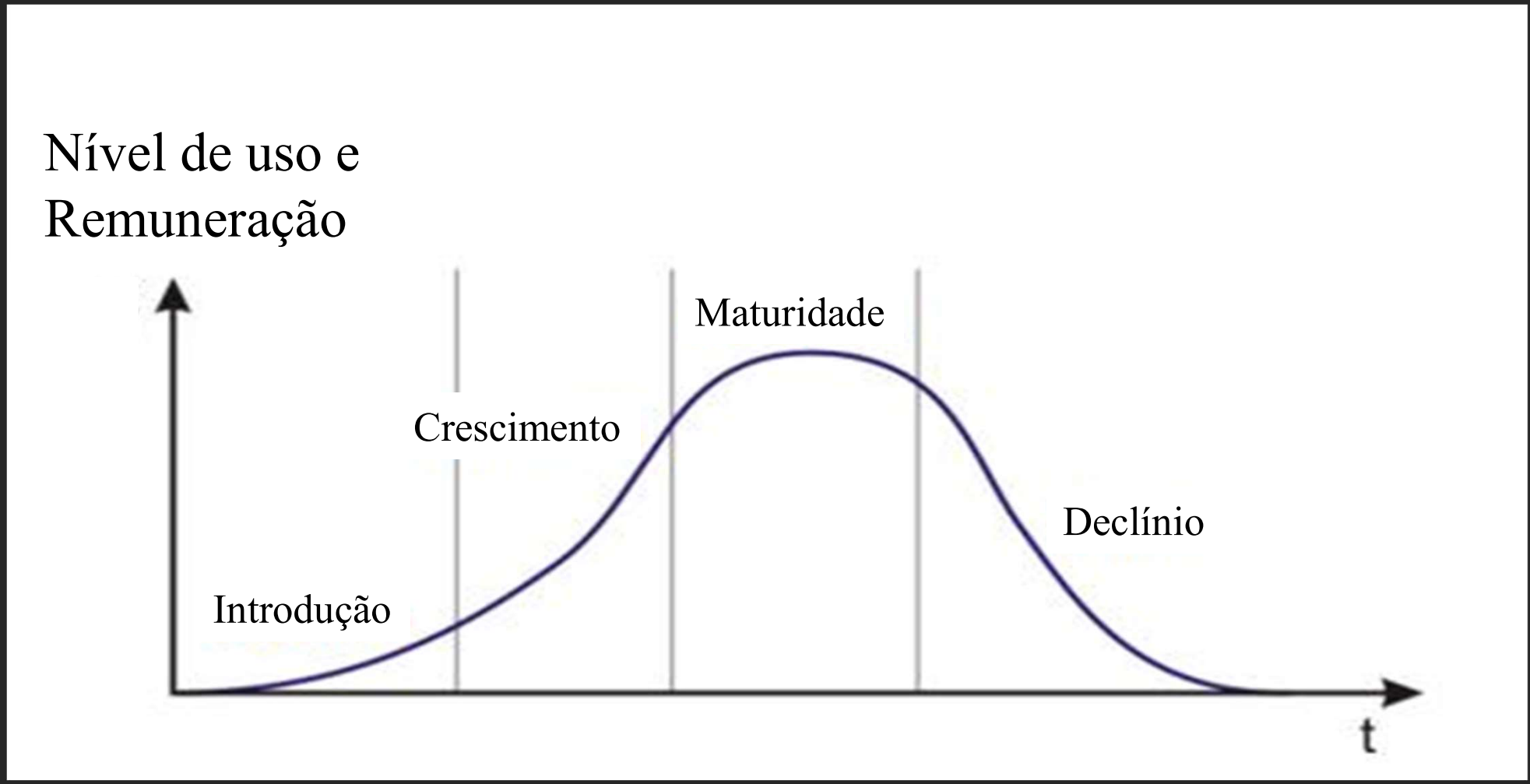
É hora de você explicar para ele como funciona o ciclo de vida de um software...

A base para a Engenharia de Software

Todo produto tem uma CURVA DE VIDA



Essa CURVA DE VIDA é acompanhada por um CICLO DE VIDA que são as Fases pelas quais passa o Software



A base para a Engenharia de Software

O Ciclo da Vida de QUALQUER PRODUTO

- Define as **fases pelas quais se passa**, da introdução à aposentadoria, estabelecendo um escopo de atividades
- Define **o que se pode esperar como resultado** do cumprimento de cada fase

Observe a curva de expectativa de vida dos seguintes produtos e responda:

- Produtos de software diferentes têm ciclos de vida diferentes?
- Qual a razão (explique conforme a resposta da questão anterior)?

ESTUDO DE CASO: de quanto em quanto tempo os seguintes produtos recebem uma nova versão (nova geração) com novos recursos funcionais?

- Microsoft Windows
- SAP - ERP

O Ciclo de Vida de Software

De que adianta compreender que o software tem uma vida que passa pela sua produção, entrega, uso e aposentadoria?

Como o gerenciamento do ciclo de vida de software ajuda a INDÚSTRIA DE SOFTWARE?

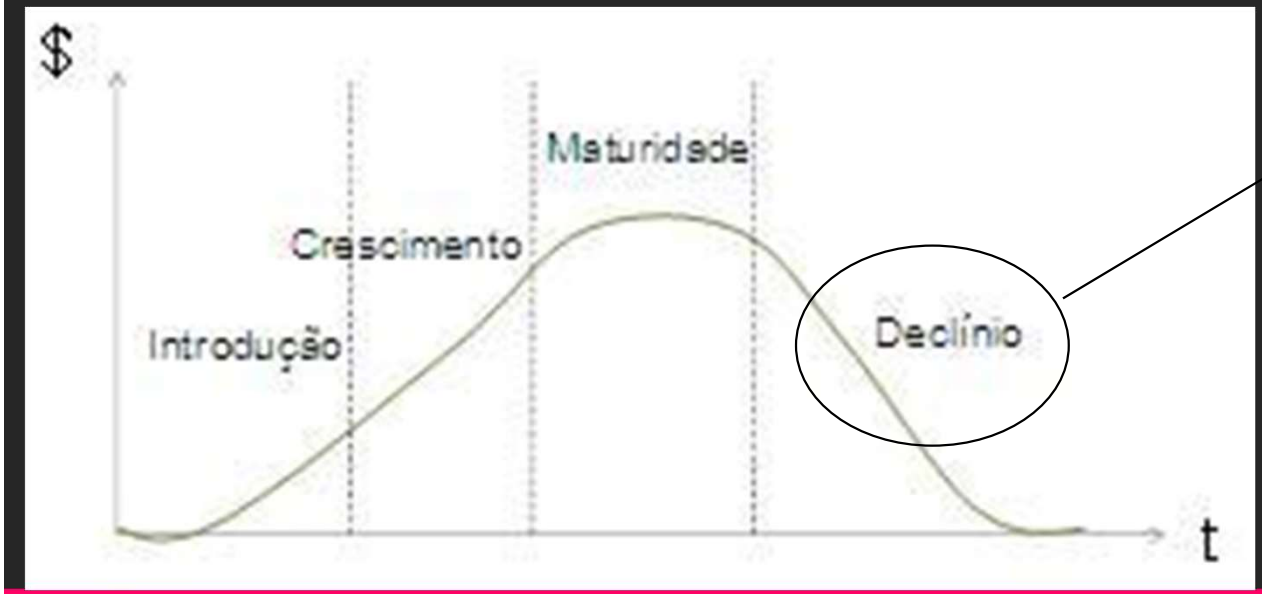
A base para a Engenharia de Software

O Ciclo da Vida bem administrado permite:

- **Retorno do capital investido no desenvolvimento** do produto, gerando lucratividades e sustentabilidade para as fábricas de software;
- **Retorno do capital investido pelo cliente**, que precisa ver melhoria em seus negócios com o uso do software, de forma a pagar o investimento na sua compra e implantação;
- **Manter a fábrica de software** plenamente e permanentemente **ocupada**, otimizando o custo de capital.

O ciclo de vida pode ajudar no planejamento e obtenção do retorno financeiro esperado.

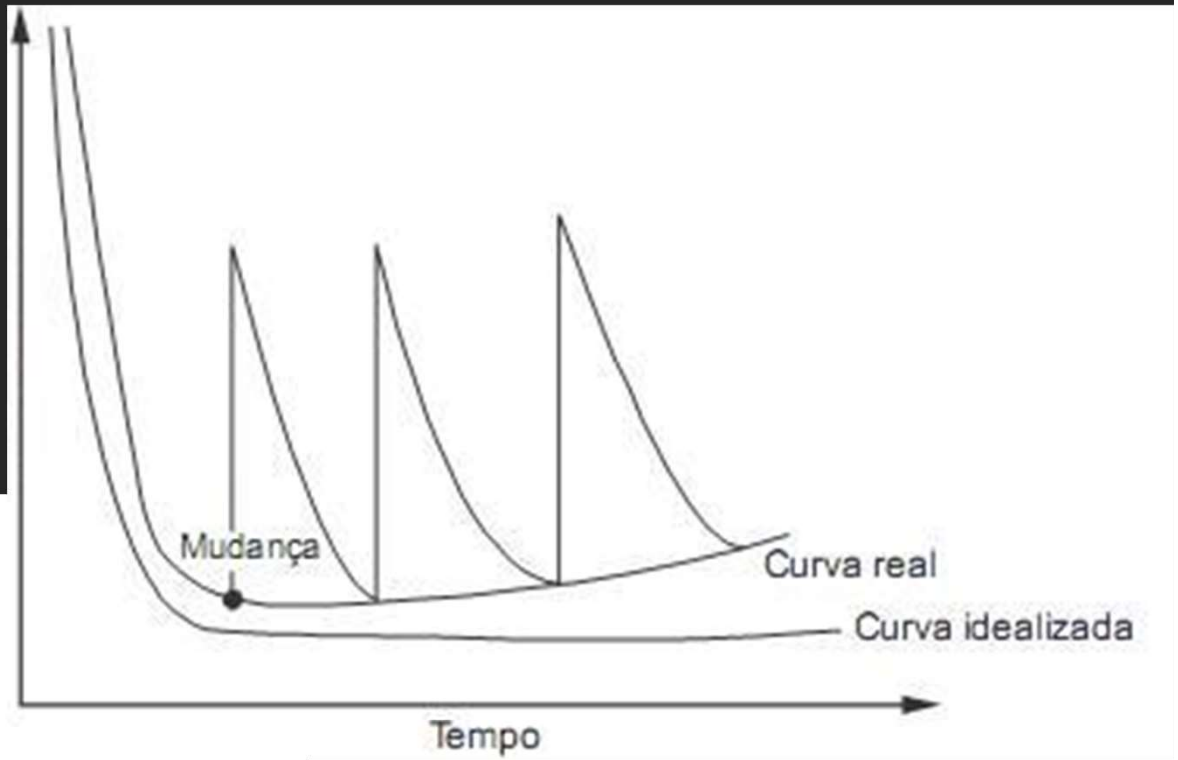
Seja qual for o modelo de consumo, quanto mais o produto estiver vivo na praça, maior será o retorno que o produto trará!



Como dar sobrevida?

Manutenções e adaptações em produtos prolongam a sua vida.

É o que acontece no versionamento de automóveis e software!



Prolongamento da vida

Manutenções podem ser do tipo:

- **Corretivas:** eliminam defeitos do Banco de Dados e programas associados
- **Adaptativas:** ajustam o Banco de Dados e programas associados para acomodar mudanças em regras de negócio que exigem novas funções e informações
- **Evolutivas:** incluem novas estruturas em Bancos de Dados e funcionalidades em programas associados aos Bancos, para criar novas oportunidades de aplicação prática para os usuários, aumentando a atratividade do produto
- **Perfectivas:** buscam ajustes de desempenho (Performance Tunning), otimização de ocupação de espaços de armazenamento de dados e consumo de recursos infraestrutururais (processamento, RAM, telecomunicações).

O Ciclo de Vida de Software

Através do estudo de Ciclo de Vida, **toma-se as ações corretas para a preservação da vida** e seu pleno aproveitamento.

No caso de software, o estudo do Ciclo de Vida trata da compreensão e atendimento das necessidades fundamentais de cada etapa do desenvolvimento e uso do software para que aconteça o seu máximo aproveitamento, pelo maior tempo possível, gerando uma relação econômica ideal.

Exemplo do ciclo de vida dos produtos da Microsoft

CICLO DE VIDA DA PRODUÇÃO DE SOFTWARE

Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

Agora que **já é conhecida** a característica do **ciclo de vida** de um software **APÓS o início da sua utilização**, **vamos conhecer** como é o **ciclo de vida ANTES do software ser lançado**!

APRENDENDO NA PRÁTICA

Você precisa agora, começar a planejar como vai desenvolver o seu software – precisa definir o chamado Ciclo de Vida de Desenvolvimento e escolher um processo de produção adequado.

O ciclo de vida de desenvolvimento vai do momento da concepção de ideias até a entrega do produto para uso e entrada no período de manutenção.

O ciclo de vida que você escolher vai determinar fases do projeto, formas de conduzir as atividades produtivas e ritmo de entregas de produtos.

Os Ciclos de Vida retratam uma forma de pensar e analisar a evolução das coisas ao longo do tempo da sua utilização.

Ciclos de Vida são formas de pensar!
São paradigmas!

Esse modelos...

Definem etapas ou fases a serem cumpridas para que o desenvolvimento de software ocorra de forma disciplinada, organizada, progressiva, compreensível, gerenciável e que alcance as expectativas de seus idealizadores.

Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

SOFTWARE DESIGN & TX

Toda produção de software envolve as seguintes etapas que precisam ser administradas:

- **Determinação de escopo:** especificação da **necessidade a ser atendida** pelo sistema.
- **Desenho lógico:** explicação **descritiva e/ou ilustrada das funções que o software** deve ter e dados que deve guardar;
- **Desenho físico: explicação descritiva e/ou ilustrada** para apresentar as escolhas de **plataforma** de desenvolvimento, arquitetura dos ambientes de hardware e software onde o sistema executará, padrões de **interface com o usuário**, padrões técnicos de construção e integração de **componentes** do software, **protocolos** e arquitetura da comunicação de dados local e remota, regras de **segurança** que serão aplicadas sobre redes de computadores e no software.
- **Implementação:** produção do software em uma **linguagem de programação**, respeitando as diretrizes de desenho lógico e físico.

Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

Na Engenharia de Software, desenvolveu-se diversos modelos conceituais de Ciclo de Vida para administrar as etapas de produção do software e sua manutenção pós-liberação para uso:

-Modelo de processo Cascata

-Modelo de processo Incremental

-Modelo de Prototipação

-Modelo de processo Espiral

CICLOS DE VIDA DE SOFTWARE

MODELO CASCATA

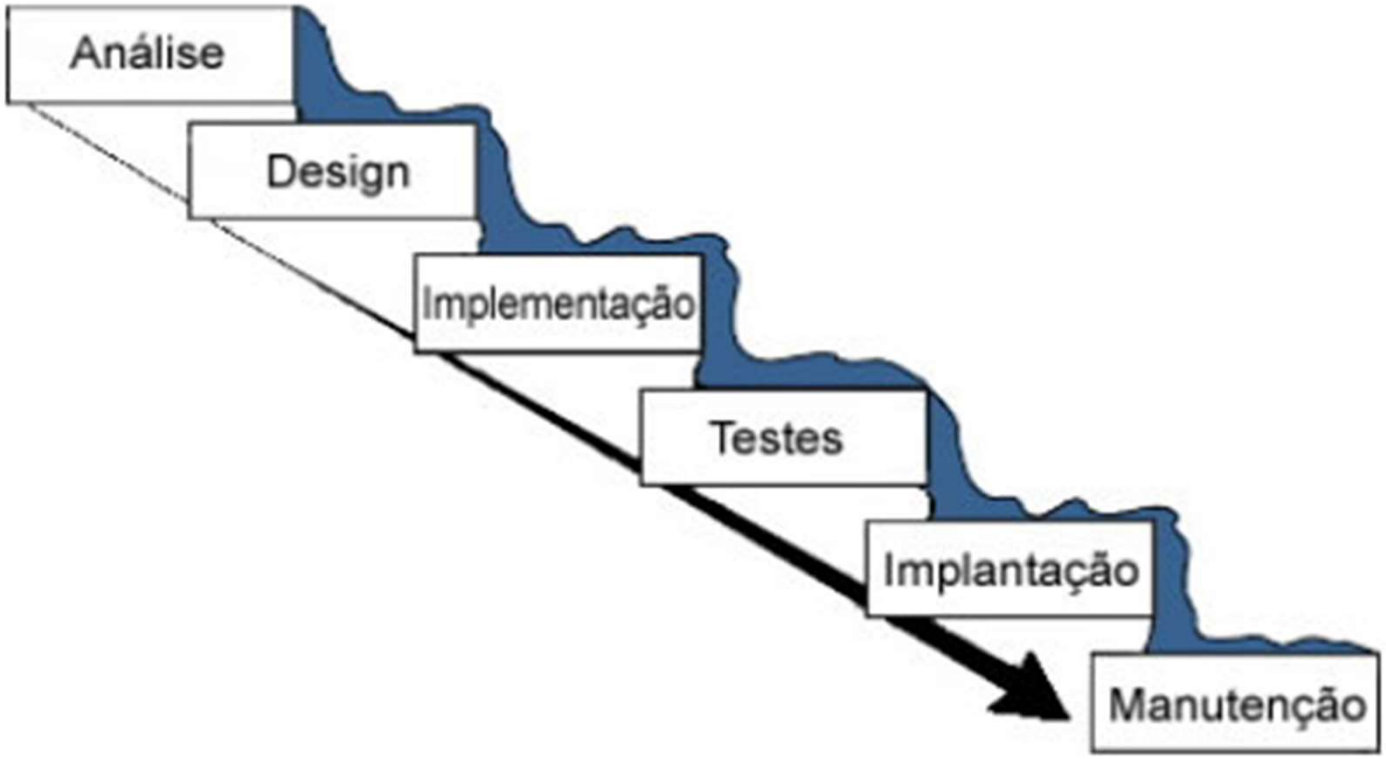
O projeto inteiro é tratado como uma frente de desenvolvimento única!

Seus passos são claros e sequenciais!

A evolução do projeto acontece como um todo: se estiver modelando o sistema, não existirá nada sendo construído. Se estiver construindo, nada poderá ser implantado no mesmo momento!

CICLOS DE VIDA DE SOFTWARE

MODELO CASCATA



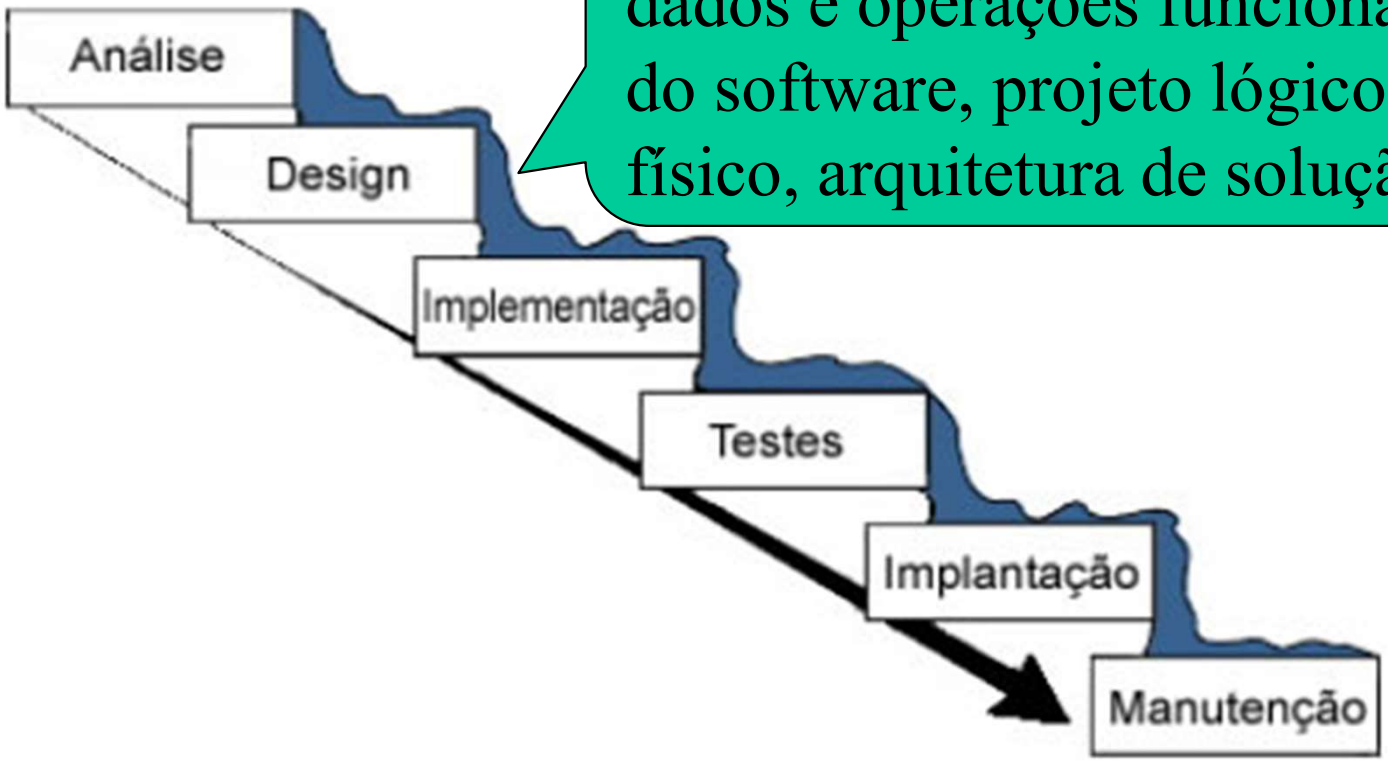
CICLOS DE VIDA DE SOFTWARE

MODELO CASCATA

Levantar e avaliar a necessidade do usuário e estudar os sistemas existentes que serão substituídos (legados). Estudar a viabilidade e prioridade dos requisitos

CICLOS DE VIDA DE SOFTWARE

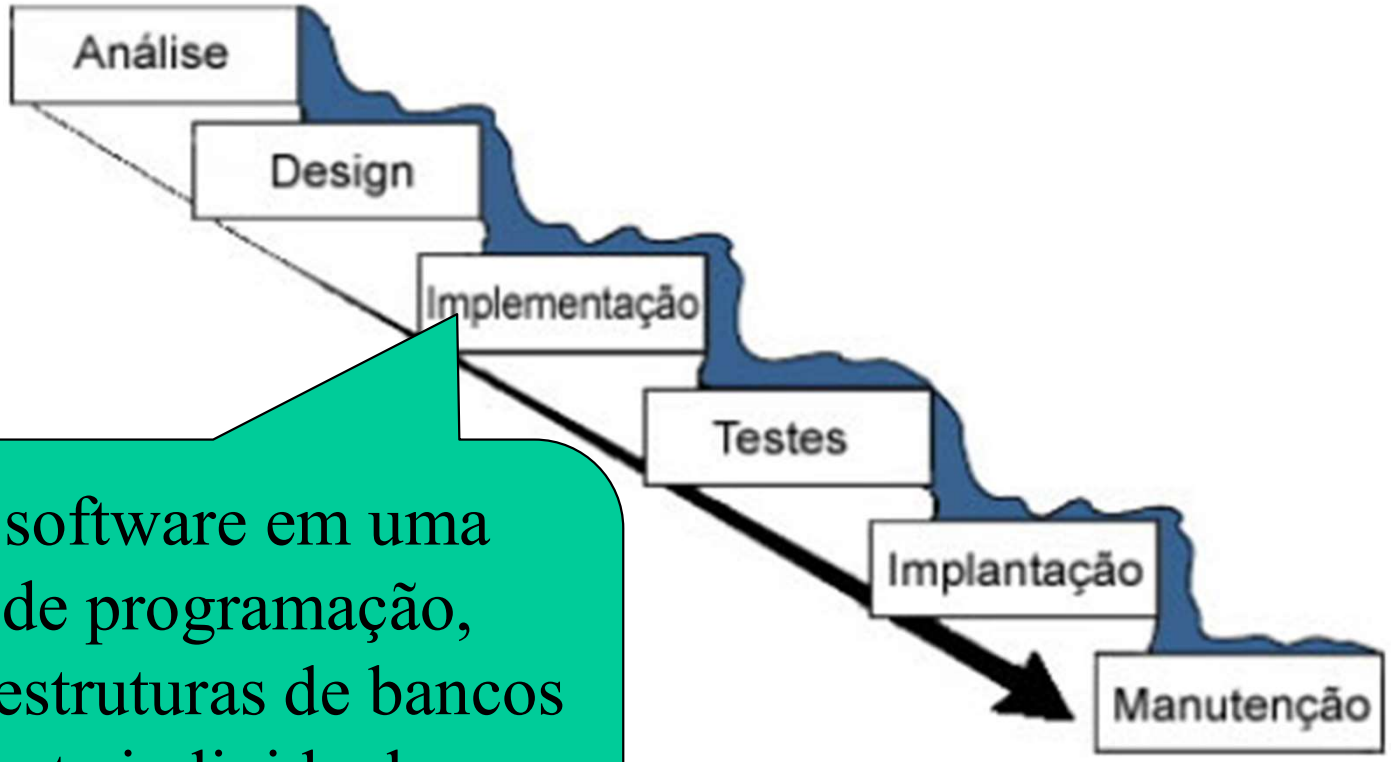
MODELO CASCATA



Modelagem das estruturas de dados e operações funcionais do software, projeto lógico e físico, arquitetura de solução.

CICLOS DE VIDA DE SOFTWARE

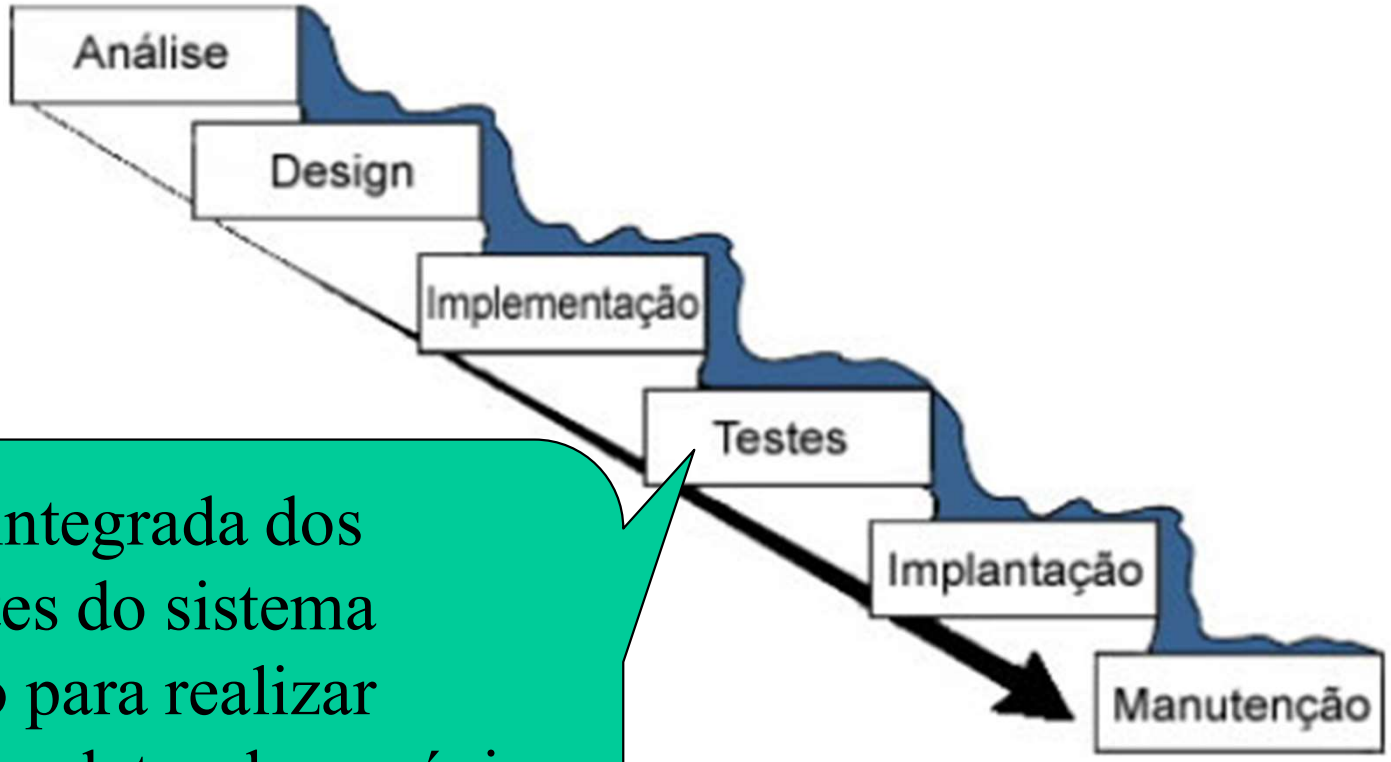
MODELO CASCATA



Escrever o software em uma linguagem de programação, criação de estruturas de bancos de dados, teste individual (unitário) dos programas. Criação de helps e manuais.

CICLOS DE VIDA DE SOFTWARE

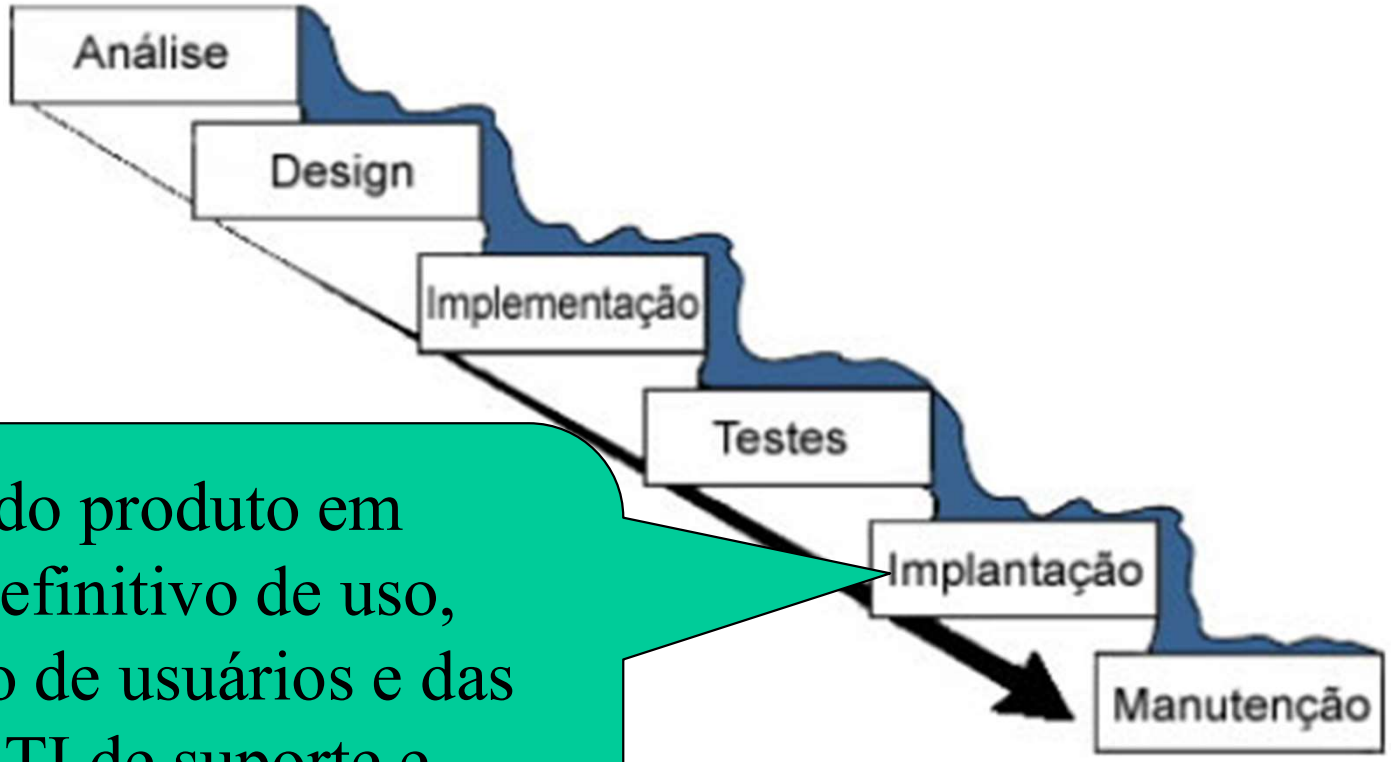
MODELO CASCATA



Avaliação integrada dos componentes do sistema interagindo para realizar funções completas de negócio, homologação com o usuário em ambiente controlado

CICLOS DE VIDA DE SOFTWARE

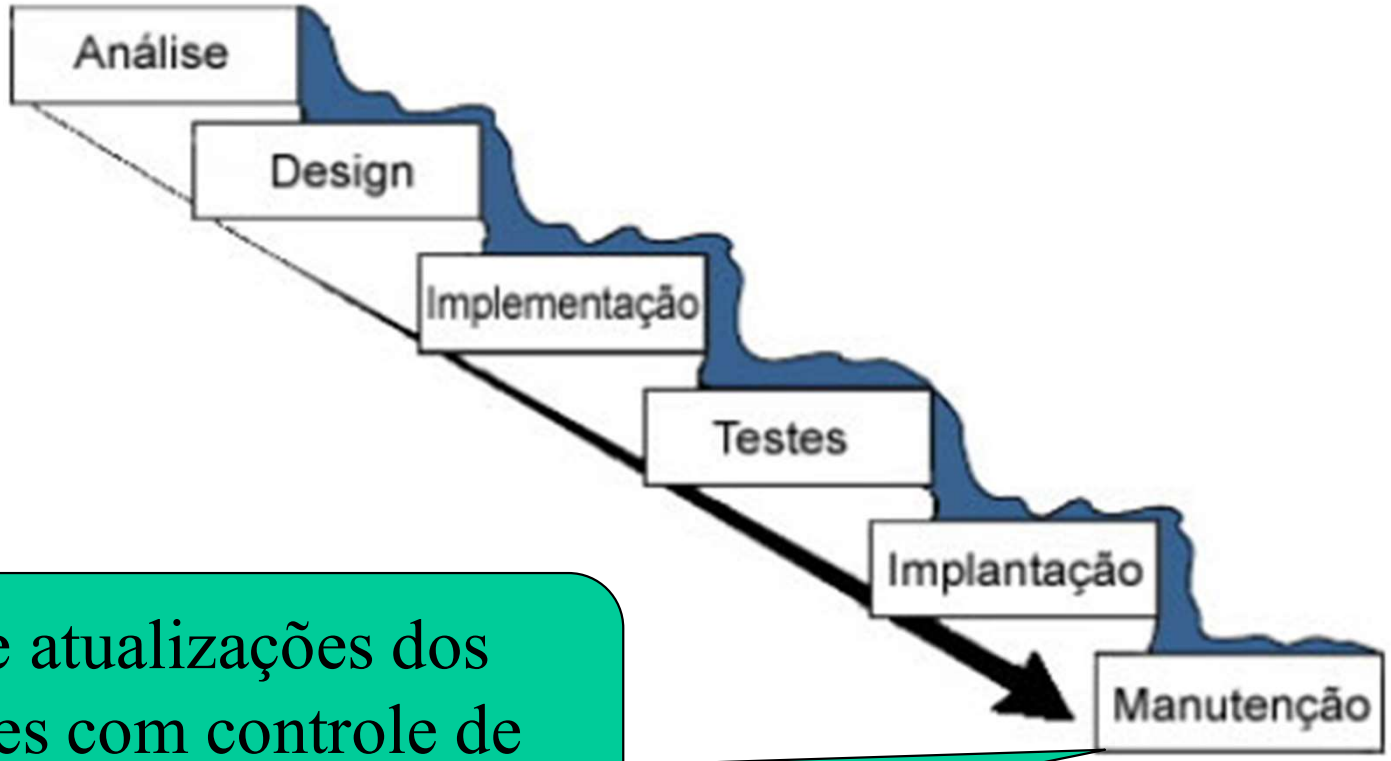
MODELO CASCATA



Instalação do produto em ambiente definitivo de uso, capacitação de usuários e das equipes de TI de suporte e operações. Configuração de acessos e segurança de dados.

CICLOS DE VIDA DE SOFTWARE

MODELO CASCATA

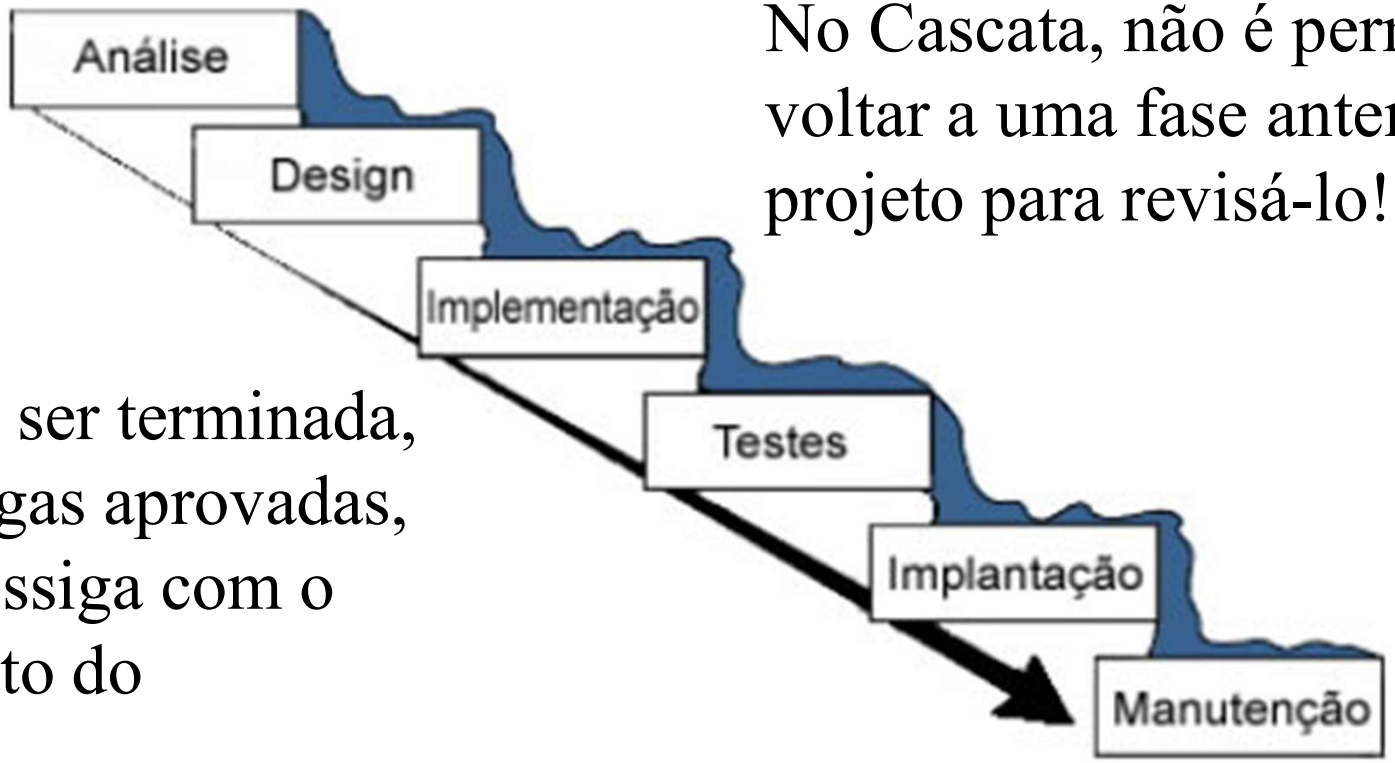


Correções e atualizações dos componentes com controle de versionamento

CICLOS DE VIDA DE SOFTWARE

MODELO CASCATA

A água que desce não volta!
No Cascata, não é permitido voltar a uma fase anterior do projeto para revisá-lo!



Cada fase deve ser terminada, com suas entregas aprovadas, para que se prossiga com o desenvolvimento do software!
As entregas de uma fase subsidiam a seguinte!

ATIVIDADE PRÁTICA

Vamos estudar o cronograma de referência para projetos de software conduzidos no modelo Cascata, da área de apostilas!

Atividade	Duração	Início	Término
PROJETO DE SOFTWARE - Condução Clássica	**185 days**	**Mon 13/02/17**	**Fri 27/10/17**
Fase de Iniciação	**10 days**	**Mon 13/02/17**	**Fri 24/02/17**
Definição do Ciclo de Vida do Software (Modelo Cascata, Incremental, Prototipação ou Espiral)	5 days	Mon 13/02/17	Fri 17/02/17
Atribuição da equipe inicial responsável pelo projeto	1 day	Mon 20/02/17	Mon 20/02/17
Identificação dos demandantes do projeto (Stakeholders)	1 day	Tue 21/02/17	Tue 21/02/17
Concepção da demanda	1 day	Wed 22/02/17	Wed 22/02/17
Estimativa de esforço, tempo e custo com base em histórico	1 day	Thu 23/02/17	Thu 23/02/17
Documentação do termo/acordo de abertura de projeto (TAP)	1 day	Fri 24/02/17	Fri 24/02/17
Documento de Visão de escopo inicial, premissas e restrições de projeto, definido	0 days	Fri 24/02/17	Fri 24/02/17

Trata-se de um exemplo de um plano de projeto, baseado em um ciclo de vida Cascata.

Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

CICLOS DE VIDA DE SOFTWARE

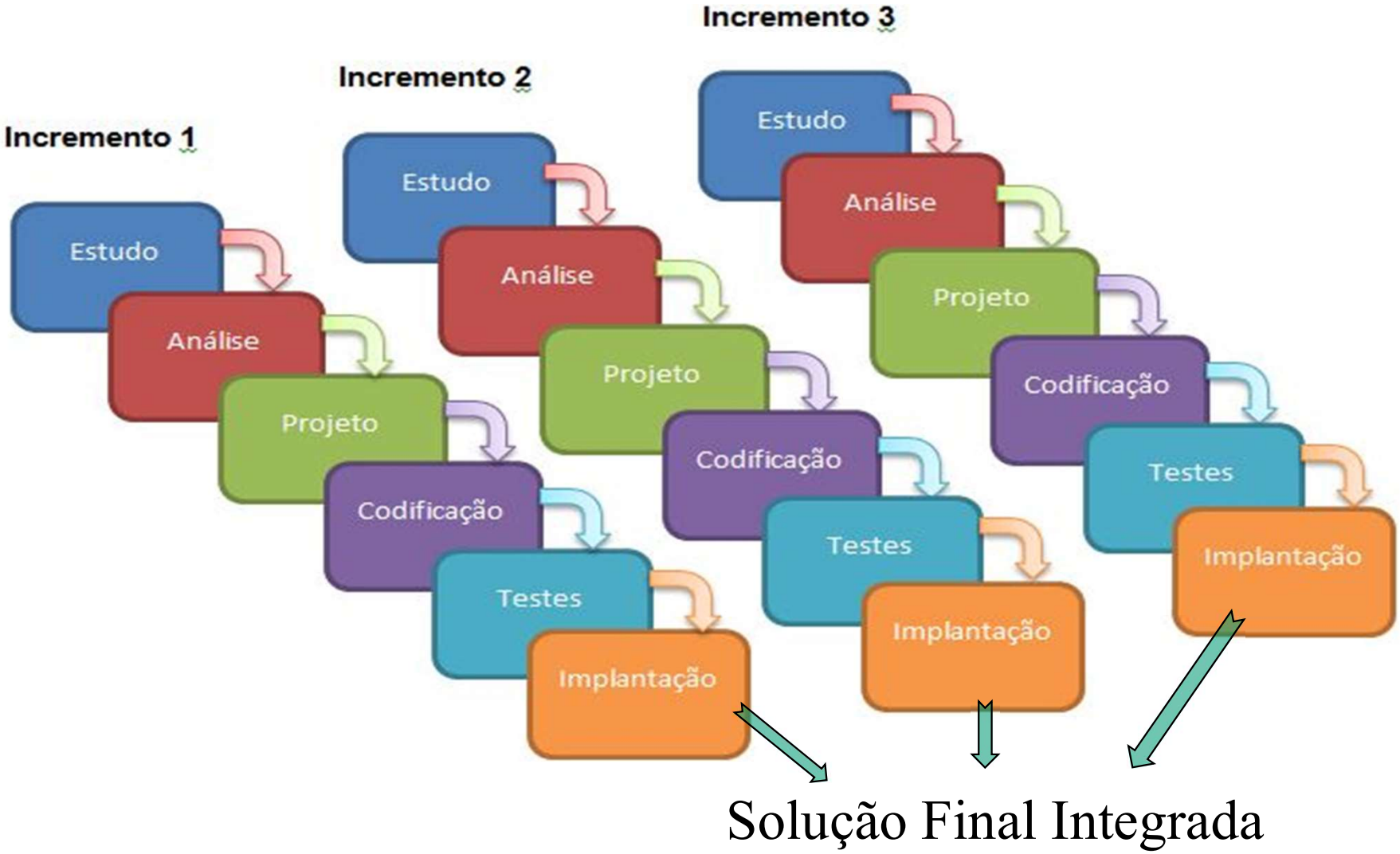
MODELO INCREMENTAL

O projeto é fatiado em módulos que podem ser produzidos com certa independência e em paralelo!

Os módulos devem ser integrados para compor o sistema/solução final para o cliente!

CICLOS DE VIDA DE SOFTWARE

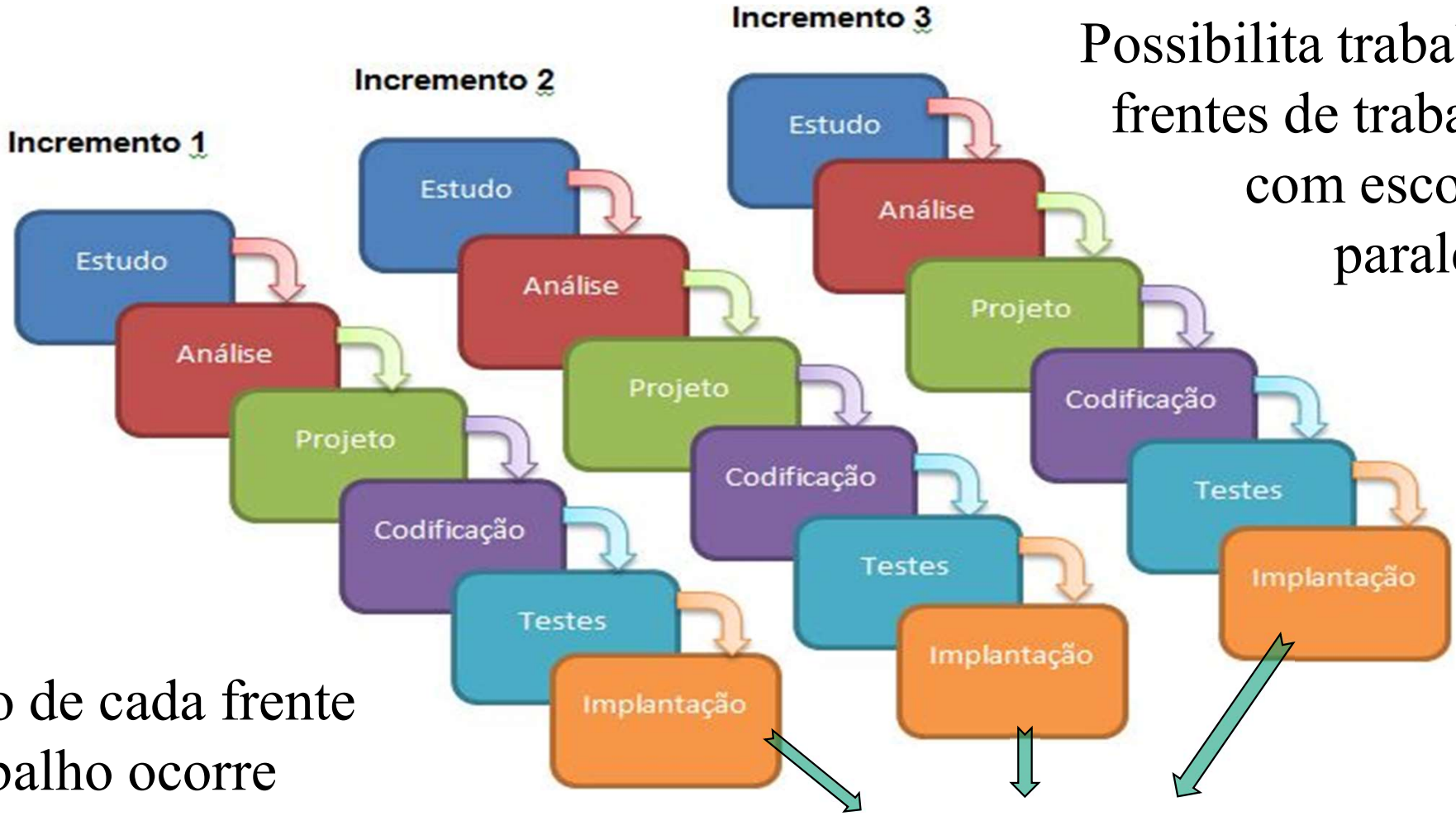
MODELO INCREMENTAL



CICLOS DE VIDA DE SOFTWARE

MODELO INCREMENTAL

Possibilita trabalhar frentes de trabalho com escopos paralelos



Dentro de cada frente de trabalho ocorre uma Cascata

Solução Final Integrada

CICLOS DE VIDA DE SOFTWARE

MODELO PROTOTIPAÇÃO EVOLUCIONÁRIA

O projeto evolui a partir de protótipos (mockup)!

Constrói-se o protótipo, valida-se o protótipo e depois, constrói-se a solução definitiva!

Um protótipo feito pode ser descartado, revisado ou aproveitado na construção definitiva, possibilitando revisão e adaptação evolutiva!

O desenvolvimento ocorre em ciclos e partes do produto de software podem ser desenvolvidas em separado e depois integradas. O cliente permanece em contato constante durante o projeto que evolui em seus requisitos e construção a cada ciclo de produção.

CICLOS DE VIDA DE SOFTWARE

MODELO PROTOTIPAÇÃO EVOLUCIONÁRIA

CICLOS DE VIDA DE SOFTWARE

MODELO PROTOTIPAÇÃO EVOLUCIONÁR

Os ciclos de desenvolvimento acontecem até que o produto esteja completo

Levantar e documentar requisitos. Definir escopo do produto

Definir atividades de trabalho, distribuir tarefas, definir prazos

Elaborar prova de conceito (POC), e protótipo do produtto (MOCUP)

Desenvolver código fonte, aplicar testes, liberar componente, integrar componentes

Liberar o software p/uso, instalar o software, treinar usuários e técnicos de operação e suporte

CICLOS DE VIDA DE SOFTWARE

MODELO ESPIRAL

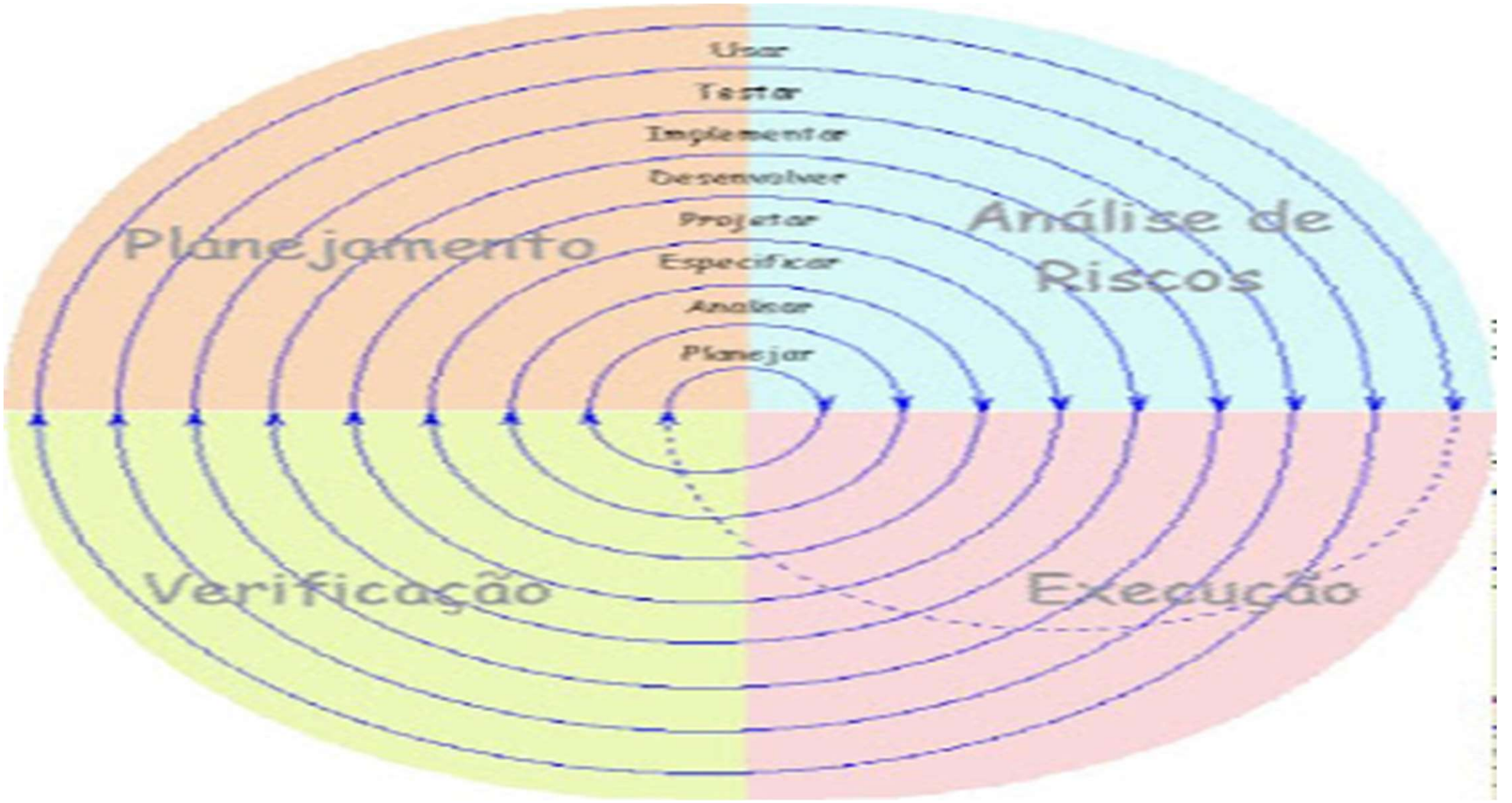
O projeto pode ser desenvolvido com flexibilidade de adaptação, onde o software pode ser repartido e ter módulos e até componentes individuais sendo evoluídos em ritmo distinto dos demais.

Contém os princípios:

- O projeto deve passar a todo tempo por Planejamento, Avaliação de Riscos observados mediante o plano, Execução do plano, Monitoração e Controle de resultados, de forma a garantir melhoria contínua no projeto;
- O desenvolvimento não é linear, ou seja, **é possível ir e voltar nas etapas do desenvolvimento** como modelagem, construção, teste sendo mais importante garantir a aderência do software aos requisitos do que o cumprimento de um plano traçado preliminarmente
- O projeto envolve negociação constante em busca do ganho mútuo entre os desenvolvedores e o cliente da solução

CICLOS DE VIDA DE SOFTWARE

MODELO ESPIRAL

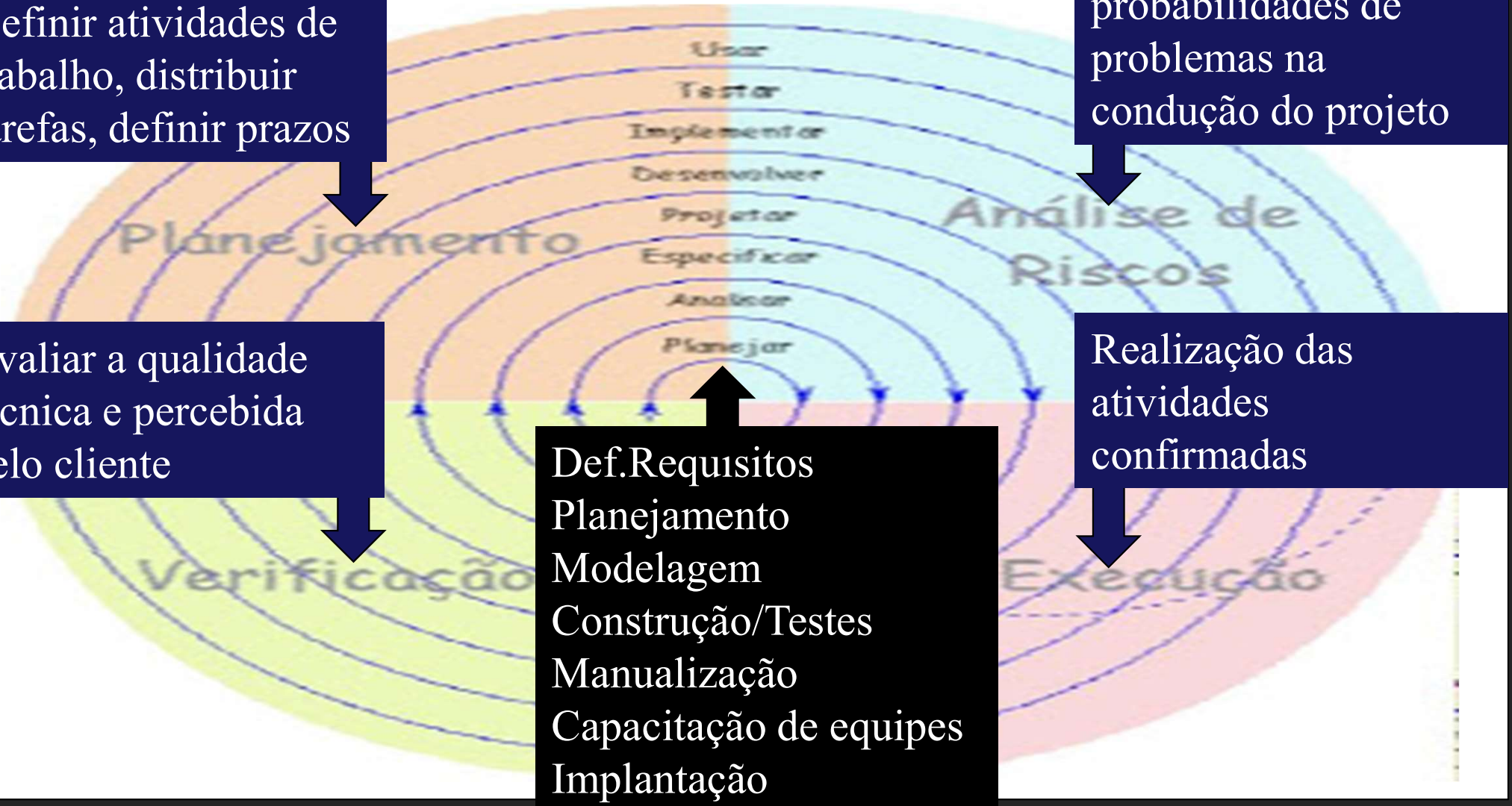


CICLOS DE VIDA DE SOFTWARE

MODELO ESPIRAL

Definir atividades de trabalho, distribuir tarefas, definir prazos

Avaliar impactos e probabilidades de problemas na condução do projeto



Avaliar a qualidade técnica e percebida pelo cliente

Realização das atividades confirmadas

Def.Requisitos
Planejamento
Modelagem
Construção/Testes
Manualização
Capacitação de equipes
Implantação

CICLOS DE VIDA DE SOFTWARE

MODELO ESPIRAL

Partes do projeto (alguns módulos e componentes de software) podem estar sendo construídos, enquanto outros estão sendo modelados e outros ainda estão tendo requisitos negociados

MÉTODOS ÁGEIS X TRADICIONAIS

Os ciclos de vida são empregados em dois modelos de condução de projetos, também chamados processos de produção de software:

-Clássicos ou Tradicionais: aplicam os ciclos de vida Cascata ou Incremental, sendo marcados por burocracia administrativa, aversão a mudanças durante o projeto e muita divisão de tarefas e responsabilidades entre os membros do projeto (a exemplo do processo RUP*);

-Ágeis: aplicam ciclos de vida de Prototipação Evolucionária (a exemplo do RAD*, DSDM*) ou Espiral (a exemplo de processos XP* e SCRUM).

**Não entraremos em detalhes desses processos, nesse momento*

ATIVIDADE PRÁTICA

No projeto que será desenvolvido para o estacionamento, o proprietário deseja que entreguemos o sistema o mais rapidamente possível.

Se o projeto for longo, ele prefere que entreguemos o sistema mesmo que parcialmente, o quanto antes.

Para o proprietário, controlar os pagamentos de forma digital é a primeira prioridade, depois o controle de entrada e saída de carros, em seguida o controle de ocupação de vagas e por fim (se ele ainda tiver dinheiro para investir), quer automatizar a leitura de placas e abertura de cancelas.

Qual ciclo de vida de projeto você propõe usar?

ATIVIDADE PRÁTICA

Decidimos que em nosso projeto para o estacionamento, vamos usar o modelo de ciclo de vida Espiral para podermos entregar rapidamente algumas partes do software para uso, de forma a cativar o cliente que está relutante quanto aos investimentos que fará.

Para empregar o ciclo Espiral, vamos adotar um processo Ágil de produção.

Algumas opções são o RAD, o DSDM, o XP, o SCRUM.

Optamos pelo SCRUM, que é o framework mais adotado internacionalmente em projetos ágeis de software.

CICLOS DE VIDA DE SOFTWARE

CICLO DE VIDA E O PROCESSO DE PRODUÇÃO DE SOFTWARE

Os **processos de produção de software** envolvem a aplicação de técnicas para gerar o produto final (o software) e administrar as atividades de trabalho da concepção até a entrega.

Existem processos **Clássicos ou Tradicionais** e processos Ágeis.

Os Processos aplicam um Ciclo de Vida como referência.

Os processos Clássicos alinham com perfeição com CVS (Ciclo de Vida de Software) Cascata ou Incremental; já os Ágeis combinam mais com abordagens Evolucionária e principalmente Espiral.

CICLOS DE VIDA DE SOFTWARE

CICLO DE VIDA E O PROCESSO DE PRODUÇÃO DE SOFTWARE

Todas as práticas que estudaremos nesta disciplina podem ser aplicadas a qualquer processo de software.

Em nosso projeto de disciplina e no desafio do ano (Challenge), aplicaremos a abordagem de condução Ágil, usando o Ciclo de Vida Espiral.

Sem entrar em detalhes neste momento, estaremos guiados pelo SCRUM nesses projetos (o modelo SCRUM será estudado em todos os seus detalhes no próximo ano, na disciplina que trata boas práticas, qualidade de software e governança em projetos).

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)

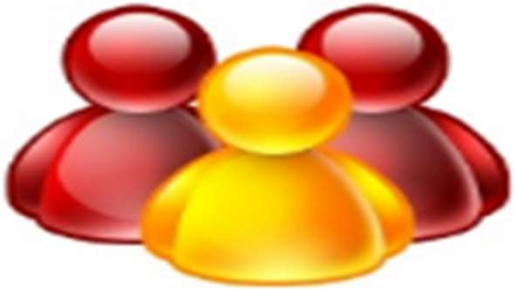
ScrumMaster (SM)

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

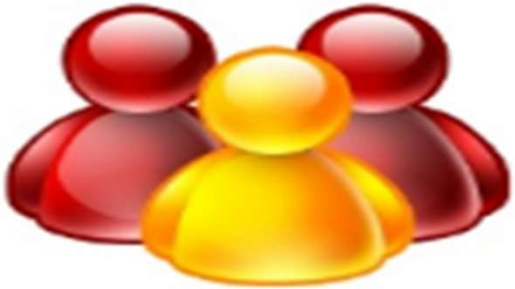
Produzir produto com qualidade e valor para o cliente

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.



SQUAD (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.)

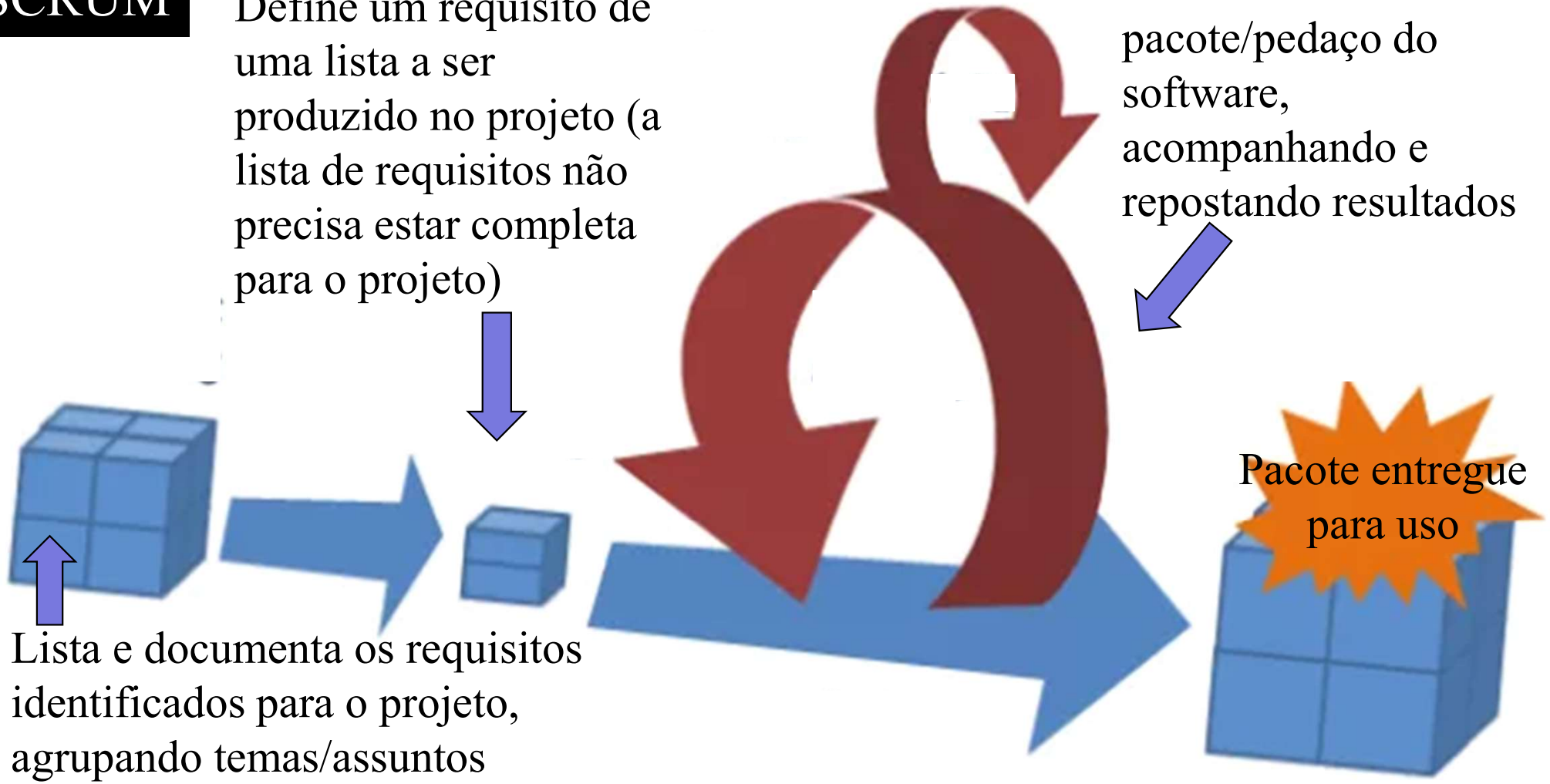
CICLOS DE VIDA DE SOFTWARE

CICLO DE VIDA E O PROCESSO DE PRODUÇÃO DE SOFTWARE

SCRUM

Define um requisito de uma lista a ser produzido no projeto (a lista de requisitos não precisa estar completa para o projeto)

Produz o pacote/pedaço do software, acompanhando e repostando resultados



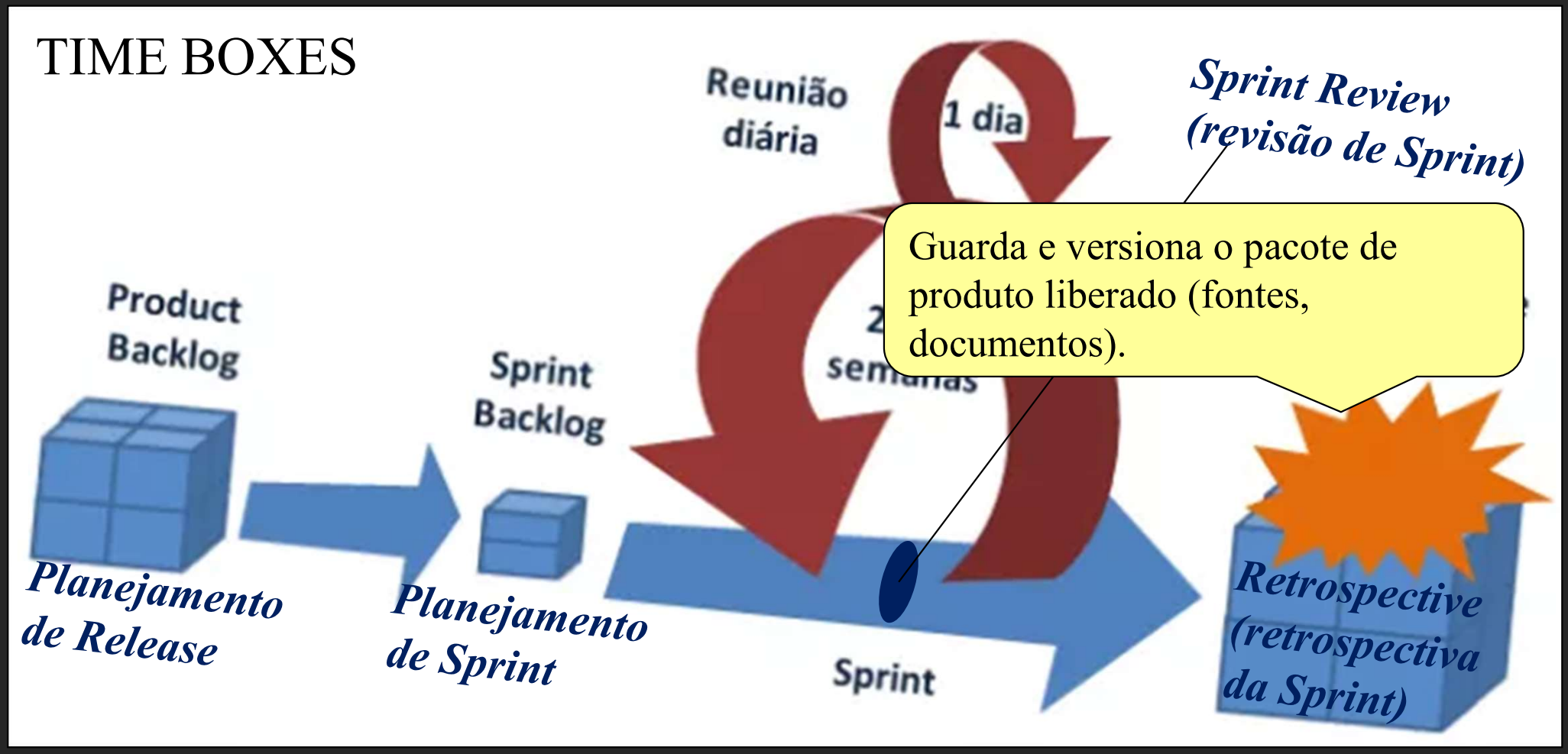
Lista e documenta os requisitos identificados para o projeto, agrupando temas/assuntos

SCRUM

A figura a seguir ilustra o processo base na metodologia:

SCRUM

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



Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

CICLOS DE VIDA DE SOFTWARE

No momento de **implantar o software** desenvolvido (momento do Go-Live) quando o produto ganha vida e utilidade real, podem ser adotadas as seguintes **estratégias** de entrega para os usuários:

TURN KEY: desliga-se o sistema antigo e liga-se o novo no dia da "virada" para a produção.

PILOT & ROLL OUTo sistema antigo vai sendo substituído pelo novo aos poucos. Em geral, escolhe-se uma unidade da empresa ou atividade para começar a virada e depois rola para as demais, seguindo um cronograma que dê segurança ao processo de mudança. Tem custo e esforço de manter os ambientes antigo e novo operando juntos mas esse custo decai conforme as implantações vão acontecendo.

PARALLEL: o sistema antigo continua operando junto com o novo. Implica em dupla digitação de dados, processamento e operação dos dois sistemas com as rotinas de segurança e backup acontecendo juntas. Possui custo elevado em função do trabalho dobrado da equipe técnica e usuários e da infraestrutura que deve atender aos dois ambientes. Os custos elevados se mantém até o final das implantações.

CICLOS DE VIDA DE SOFTWARE

DEBATE

Quando devemos usar cada uma dessas estratégias de implanação?

TURN KEY (VIRADA DE CHAVE)

PILOTO E ROLAGEM (PILOT & ROLL OUT)

PARALLEL (PARALELO)

CICLOS DE VIDA DE SOFTWARE

Quando usar cada estratégia de implantação?

TURN KEY: os custos de manutenção dos dois ambientes (antigo e novo) são proibitivos ou os riscos operacionais de uma mudança completa são baixos (existe como recuperar dados da empresa e reestabelecer processos com certa rapidez em caso de falha no novo software).

PILOT & ROLL OUT: existirão resistências à implantação por parte dos usuários e a implantação faseada, iniciando pelos usuários menos resistentes irá gerar exemplificação de sucesso e convencimento dos demais usuários, facilitando a introdução da nova cultura de trabalho que o sistema impõe.

PARALLEL: os processos da rotina empresarial comprometidos no software são muito sensíveis e pertentem à cadeia de valor ao cliente (gera produtos e serviços percebidos diretamente pelo cliente), e existe risco de uma falha no software comprometer a imagem e os resultados financeiros e operacionais finais.

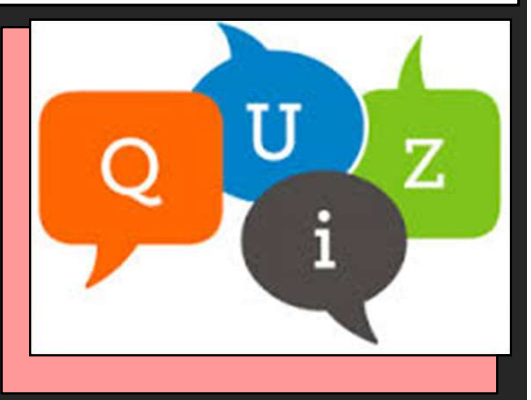
OBS: O momento imediatamente após o Go-Live deve ser acompanhado de perto pela equipe de TI com o pronto apoio dos desenvolvedores, se for necessário. Esse período conhecido como estabilização pode durar de alguns dias até alguns meses, dependendo da complexidade do software e a quantidade de usuários e suas localizações de trabalho.

CICLOS DE VIDA DE SOFTWARE

Você deve optar por um ciclo de vida de software em seus projetos!

Uma empresa pode adotar mais de um tipo de ciclo de vida, dependendo das características do projeto de software.

CICLOS DE VIDA DE SOFTWARE



Vamos jogar!

Valendo um brinde!

Kahoot!

Jogo de aprendizado de CVS

https://kahoot.it/#/

Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

Os **ciclos de vida** de software determinam:

- Fases da produção de software com objetivos bem definidos

Os **processos** de software aplicam um dos modelos de ciclo de vida e o complementam com:

- Detalhes de como executar as tarefas em cada fase;
- Papéis e responsabilidades em cada atividade;
- Entradas e saídas esperadas por atividade e fase;
- Artefatos a serem empregados;
- Indicadores e controles de desempenho e resultados.

Você conhecerá mais sobre processo de sw na disciplina de gestão de qualidade!

EXECUÇÃO DE UM PROJETO DE SOFTWARE – VISÃO GERAL DE UM CRONOGRAMA DE ATIVIDADES

Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

SOFTWARE DESIGN & TX

Os **ciclos de vida de software são a base** para a definição do processo de trabalho que contempla as **tarefas a serem feitas** pela equipe de projeto ao longo do tempo.

Escolhido um ciclo de vida, podem ser desenvolvidos cronogramas de projeto com detalhes de tarefas, duração, período de execução, recursos aplicados.

Projeto Exemplo - Planejamento Clássico - Cascata	138 dias?	Qua 10/04/19	Sex 18/10/19	
Fase I - Engenharia de Requisitos	**23 dias**	**Qua 10/04/19**	**Sex 10/05/19**	
Decclaração da visão de projeto (texto descritivo/Pitch com Problema a resolver e Solução idealizada))	17 dias	Qua 10/04/19	Qui 02/05/19	
Levantamento de requisitos	17 dias	Qua 10/04/19	Qui 02/05/19	
Prototipação para validar ideias sobre os requisitos	17 dias	Qua 10/04/19	Qui 02/05/19	
Documentação da lista de requisitos	17 dias	Qua 10/04/19	Qui 02/05/19	
APRESENTAÇÃO DO PROJETO CONCLUÍDA PARA INICIAR OS TRABALHOS	0 dias	Sex 03/05/19	Sex 03/05/19	6
Análise de viabilidade (anotação da justificativa de inviabilização de requisitos)	5 dias	Sex 03/05/19	Qui 09/05/19	6
Priorização (ajuste da lista de requisitos, sequenciando no mais prioritário para o menos)	1 dia	Sex 10/05/19	Sex 10/05/19	8
LISTA DE REQUISITOS CONCLUÍDA	0 dias	Sex 10/05/19	Sex 10/05/19	9
Fase II - Modelagem do software	**93 dias?**	**Seg 13/05/19**	**Qua 18/09/19**	
Identificação de entidades e relações de dados	1 dia?	Seg 13/05/19	Seg 13/05/19	10
Modelagem do banco de dados	1 dia?	Ter 14/05/19	Ter 14/05/19	12
Elaboração do dicionário de dados	1 dia?	Qua 15/05/19	Qua 15/05/19	13
Diagramar Casos de Uso	20 dias	Qui 16/05/19	Qua 12/06/19	14
Documentar Casos de Uso	40 dias	Qui 13/06/19	Qua 07/08/19	15
Modelar Classes	30 dias	Qui 08/08/19	Qua 18/09/19	16
DOCUMENTAÇÃO DE ENGENHARIA FINALIZADA	0 dias	Qua 18/09/19	Qua 18/09/19	17
Fase III - Construção	**53 dias**	**Seg 05/08/19**	**Qua 16/10/19**	
Construção de banco de dsdos	20 dias	Qui 08/08/19	Qua 04/09/19	17II
Codificação de software	50 dias	Qui 08/08/19	Qua 16/10/19	17II
Teste unitário de cada componente finalizado	50 dias	Seg 05/08/19	Sex 11/10/19	
CODIGO FINALIZADO COM O SEU BANCO DE DADOS	0 dias	Sex 11/10/19	Sex 11/10/19	22
Fase IV - Homologação	**5 dias**	**Seg 14/10/19**	**Sex 18/10/19**	
Teste integrado da solução	5 dias	Seg 14/10/19	Sex 18/10/19	22
PRODUTO ENTREGUE	0 dias	Sex 18/10/19	Sex 18/10/19	25
Refinamento da apresentação final do projeto/produto	5 dias	Seg 14/10/19	Sex 18/10/19	22
APRESENTAÇÃO FINAL PREPARADA	0 dias	Sex 18/10/19	Sex 18/10/19	27

Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

O planejamento de um projeto passa pelas etapas de:

1. **Definição do escopo** do software (requisitos/entregas esperadas)
2. **Escolha do ciclo de vida** ideal para o tipo de projeto
3. **Definição das tarefas** para desenhar e produzir os itens do escopo
4. **Atribuição de responsáveis** por cada tarefa
5. **Definição de tempo** de trabalho com base na competência de cada profissional alocado para cada tarefa e nas ferramentas que cada profissional possui para apoiá-lo
6. **Determinação dos custos** de mão-de-obra, aquisição de software e hardware que serão usados como plataforma de desenvolvimento (servidores, equipamentos de usuário, sistemas operacionais, sistemas gerenciadores de bancos de dados, sistemas de segurança, ferramentas de apoio à engenharia de software).

O PMI (Project Management Institute) é a maior referência mundial em práticas de gerenciamento de projetos e estabelece 10 áreas de conhecimento a serem tratadas durante o ciclo de desenvolvimento, no seu livro PMBoK (Project Management Body of Knowlege).

Áreas de conhecimento PMBok7

1-Gerenciamento de integração: *fundamental nos atuais projetos que têm grandes equipes de desenvolvimento, produzindo partes de uma solução completa. Essa área do conhecimento pede que sejam planejados e controlados os pontos de intersecção de trabalhos.*

2-Gerenciamento do escopo: *focado na definição das entregas (características do produto) e tarefas a serem realizadas.*

3-Gerenciamento do tempo: *estabelecer uma proposta de ciclos de entregas que constituem um cronograma ou plano de entregas.*

4-Gerenciamento do custo: *estimar o valor a ser investido no desenvolvimento do projeto e gerenciar os gastos, mantendo foco no orçamento negociado no projeto. O custo dependerá do esforço ou seja, da quantidade de pessoas colocadas para trabalhar em uma janela de tempo.*

5-Gerenciamento de recursos: *recursos são pessoas e infraestrutura (equipamentos, softwares de apoio, por exemplo) que precisam ser previstos e controlados na sua aplicação ao longo do projeto.*

Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

6-Gerenciamento de aquisição/fornecimentos: *foca em planejar compras a serem feitas para suprir as necessidades de equipamentos, softwares e mão-de-obra terceirizada para o projeto..*

7-Gerenciamento de comunicação: *objetiva manter o alinhamento entre membros da equipe de projeto e todas as frentes/times de trabalho. Também objetiva manter o alinhamento com stakeholders (partes interessadas no projeto), notificando sobre planos e realizações.*

8-Gerenciamento de partes interessadas: *foco em registrar quem são as pessoas interessadas no projeto (tanto da área usuária quanto técnicos) e suas expectativas, identificando os afetados pela solução que será desenvolvida, os influenciadores das decisões de projeto e os financiadores.*

9-Gerenciamento do risco: *identificar quais são os fatores que geram risco ao projeto (risco de perder recursos, indefinição de escopo, por exemplo), estimar o grau de risco (probabilidade de ocorrência e impacto no prazo, custo e escopo, por exemplo) e planejar e executar ações de mitigação (eliminar o fator de risco, transferir o risco, criar alternativas de contingenciamento a serem acionadas se surgir problema).*

10-Gerenciamento de qualidade: *define indicadores de sucesso (métricas de desempenho) do projeto que devem incluir medições tangíveis de resultados de custos, cumprimento de metas de prazos para entregas, escopo atendido, distribuição de carga de trabalho entre as pessoas da equipe de projeto, cobertura de requisitos entregues versus solicitados, percentual de valor agregado das entregas, produtividade por desenvolvedor, entre outras métricas da escolha de cada empresa.*

São considerados como principais indicadores do PMBoK:

- Valor agregado (VA)
- Índice de Desempenho de Prazo (IDP)
- Índice de Desempenho de Custo (IDC)
- Taxa de tarefas realizadas
- Desvios de esforço

O PMBoK determina como TRIPLA RESTRIÇÃO de projeto as determinações de escopo, tempo, custo de projeto são os maiores desafios do planejamento e controle, sendo também fatores que afetam todos os demais aspectos da gestão:

-Se mudar escopo, isso afeta custo, risco, qualidade, integração, custo, prazo;

-Se mudar prazo, isso reduz a disponibilidade de agendamento de recursos, podendo implicar em custos adicionais para dar conta do trabalho, gerando novos riscos, modificando controles de qualidade e criando novas situações de comunicação e integração;

-Se mudar o orçamento (custo) do projeto, isso pode reduzir ou ampliar recursos que impactarão prazos, podendo gerar novos pontos de integração, comunicação e controle de qualidade.

Ao final, um cronograma com releases (entregas do escopo) previstas, recursos e custos associados é traçado, e depois acompanhado pela equipe de projeto.

A escolha de aplicar um modelo Clássico ou Ágil é da equipe de desenvolvimento, conforme as características do projeto.

As ferramentas de gerenciamento serão escolhidas em vista do modelo de ciclo de vida de projeto e processo escolhido.

Nosso foco será no modelo de CVS ágil em Espiral, com processo Scrum

As **determinações de tempo e custo de projeto são grandes desafios** do planejamento.

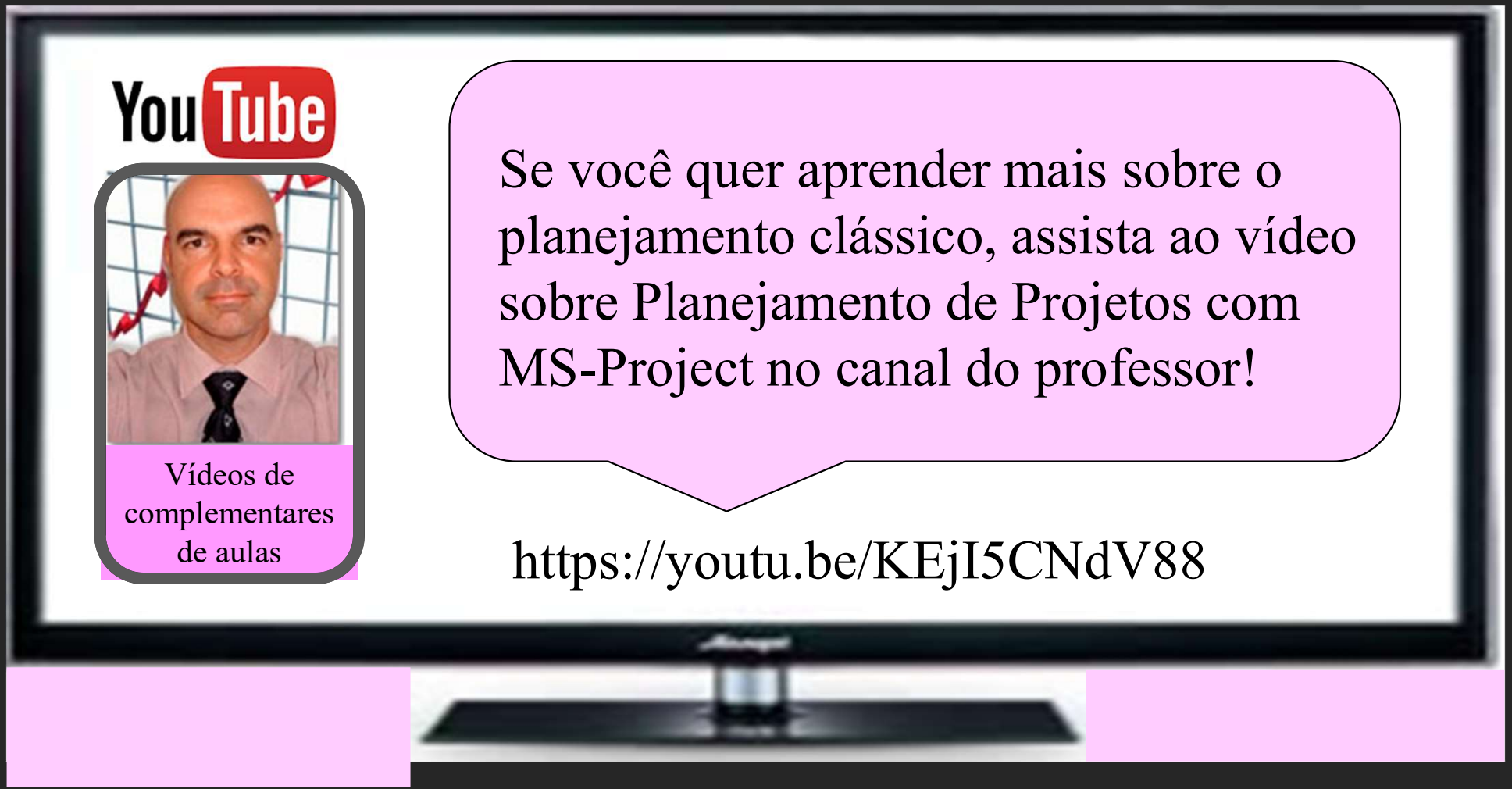
Para auxiliar nesse trabalho de estimativa, **existem modelos de previsão e projeção baseados em métricas** (que serão estudados em outra disciplina).

Outras formas de estimar projetos são: por **analogia** a p anteriormente realizados e por **opinião de especialistas**.

Ao final, um cronograma com releases (entregas) previst e custos associados é traçado, e depois acompanhado pel projeto.

Você conhecerá mais sobre processo de sw na disciplina de gestão de qualidade e governança!

CANAL DO PROFESSOR

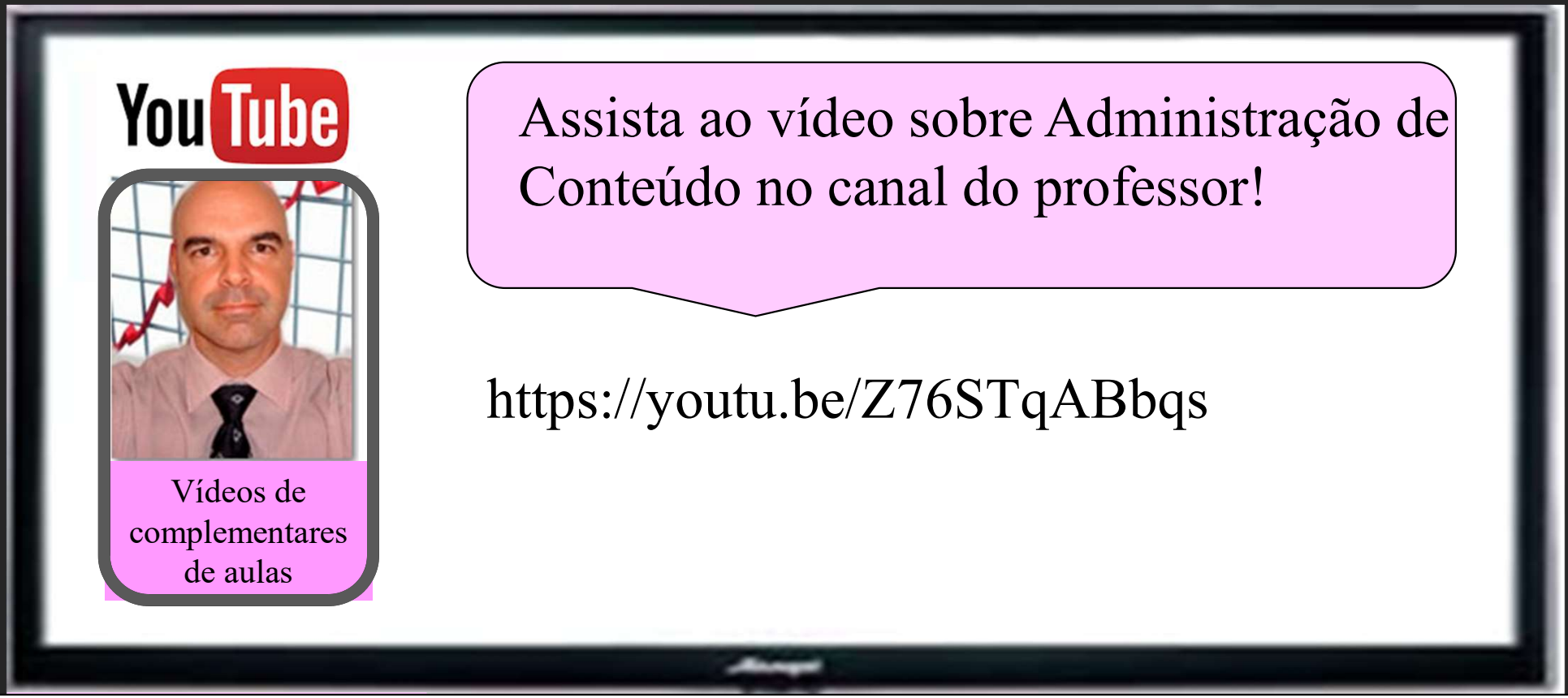


Gerenciamento da Documentação do Projeto

Toda a documentação do projeto deve ser administrada de forma a:

- Toda a **equipe** de desenvolvimento **saber onde está a fonte oficial** de informação;
- Ninguém **fazer alterações** diretamente na fonte oficial mas **em uma cópia**, sendo a fonte oficial atualizada após a confirmação de que as modificações na cópia estão corretas;
- Permitir **fácil e rápido acesso** dos projetistas aos documentos, **com regras de segurança** de acesso aplicadas;
- Todos **saberem quando um documento oficial** se encontra **em estado de revisão** com cópia sendo editada, sendo **conhecido o responsável** pela edição em curso.

CANAL DO PROFESSOR



Observação: **nesse momento, não estudaremos nenhum software de apoio ao gerenciamento de conteúdo!**
Use as dicas sobre padronização de nomes de arquivos e diretórios propostos no vídeo para você administrar seus arquivos de projeto!

ATIVIDADE EXTRA

Defina grupos de 2 pessoas para trabalharem o projeto que desenvolveremos durante a disciplina e as atividades que serão feitas em sala como exercícios de fixação.

No seu grupo, defina um conjunto de convenções de nomes de arquivos e pastas que permitam organizar toda a documentação que vocês produzirão nos projetos que virão. Tome por base o modelo Cascata e os tipos de conteúdos de cada fase trabalha.

Validem com o professor quando concluírem

ATIVIDADE EXTRA

Vamos trocar nossas ideias!

Objetivo: definirmos juntos um padrão para organizar os documentos e arquivos fonte de um projeto de software, usando uma estrutura de diretórios.

Cada grupo expõe a sua proposta, o professor anota de forma que todos vejam. **Ninguém debate durante essa exposição!**

Depois que todos falarem, negociaremos um padrão comum.

ATIVIDADE EXTRA

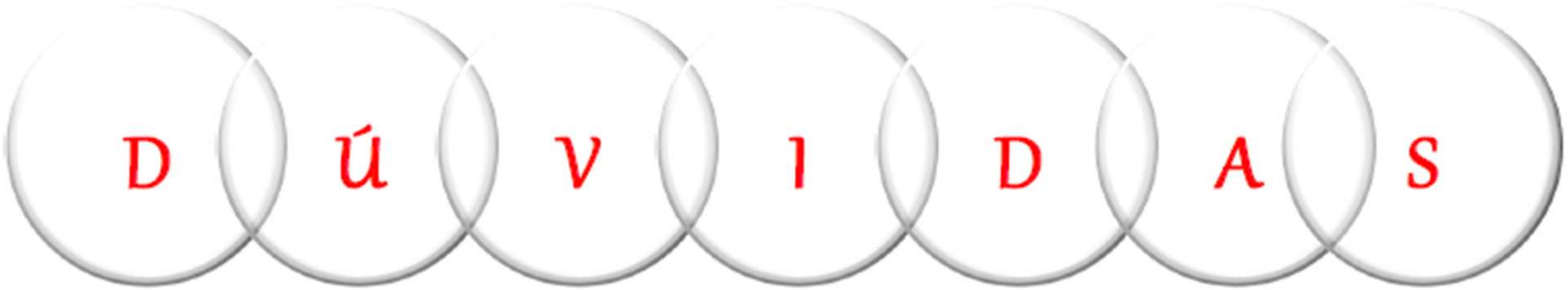
Agora que alinhamos um padrão para todas as equipes...

Crie a estrutura de pastas que servirá como repositório de arquivos em um Drive na nuvem.

Crie também uma estrutura de diretórios em outro drive na nuvem que funcionará como backup.

Defina a política para realizar as cópias de segurança (backup): periodicidade, responsável.

Chamem o professor ao terminarem



Prof. Renato Jardim Parducci – profrenato.parducci@fiap.com.br

Material de aula estará no site após a aula.

BONS ESTUDOS!

Bibliografia

- PRESSMAN, R. S. Engenharia de software. São Paulo: Editora McGraw-Hill, 2002.
- SOMMERVILLE, IAN. Engenharia de software. Editora Pearson, 9.ed. - São Paulo, 2014.
- BEZERRA, EDUARDO. Princípios de análise e projeto de sistemas com UML. Alta Books, Rio de Janeiro, 2006.