UNEMAT

Universidade do Estado de Mato Grosso

Carlos Alberto Reyes Maldonado

Engenharia de Software

Modelos de processo (cascata, incremental, espiral, etc.). Qualidade e métricas de processo.

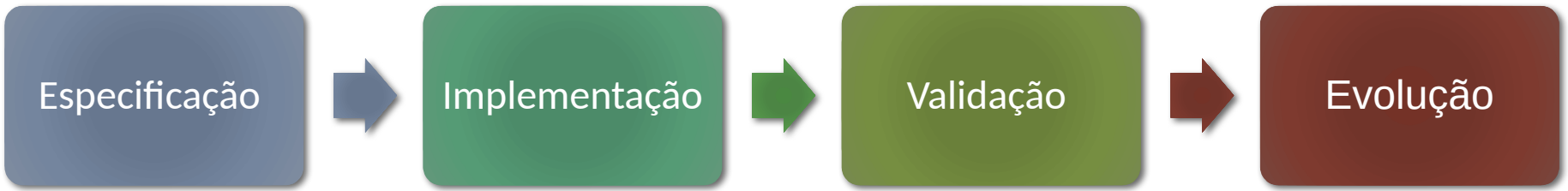
Prof. Me Ramilton Costa Gomes Júnior

Agenda

- Modelo de processo
- Qualidade
- Métricas de processos

Modelo de processo

- Um conjunto estruturado de atividades necessárias para desenvolver um sistema de software.



- Um modelo de processo de software é uma representação abstrata do processo.

Modelo de processo

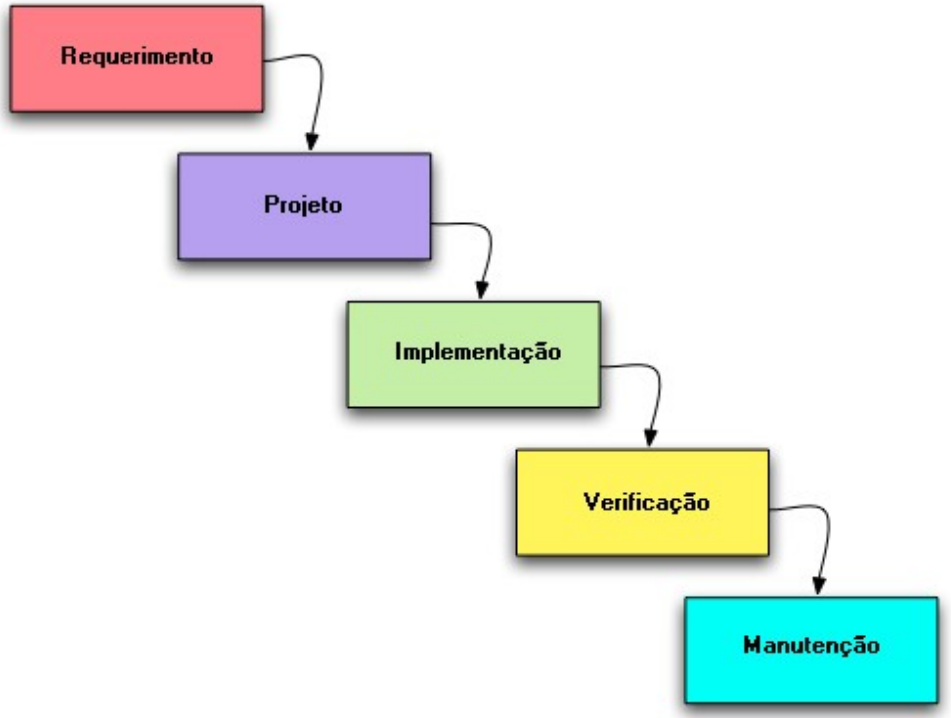
- Processo de software
  - Modelo cascata (Clássico)
    - Fases separadas e distintas de especificação e desenvolvimento;
  - Prototipação e espiral
    - Especificação e desenvolvimento são intercalados.

Modelo de processo

- Modelo cascata
  - Um dos principais modelos (Royce,1970);
  - O desenvolvimento de um estágio deve terminar antes do próximo começar;
  - Simples, mas, não reflete, efetivamente , o modo como o software é desenvolvido;
  - Derivado do mundo do hardware (Linhas de montagens).

Modelo de processo

- Modelo cascata

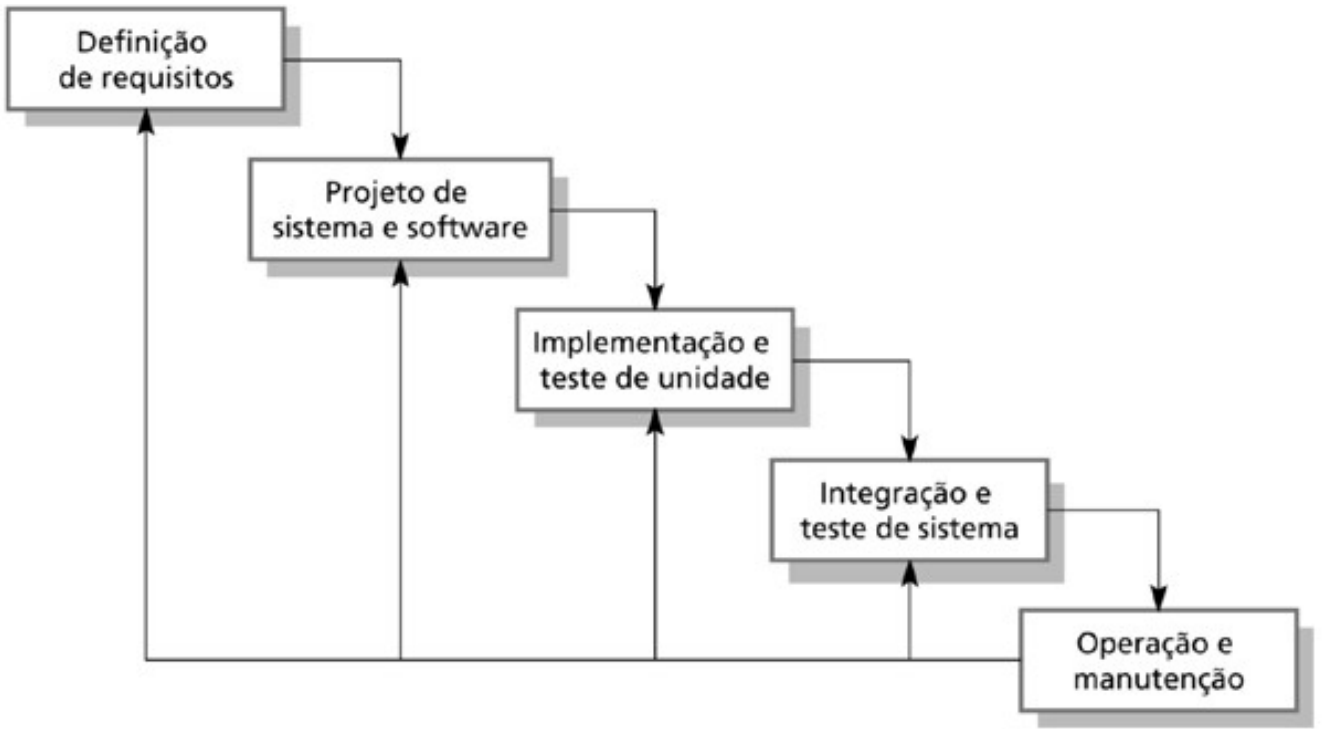


Modelo de processo

- Modelo cascata

Figura 4.1

Ciclo de vida de software.

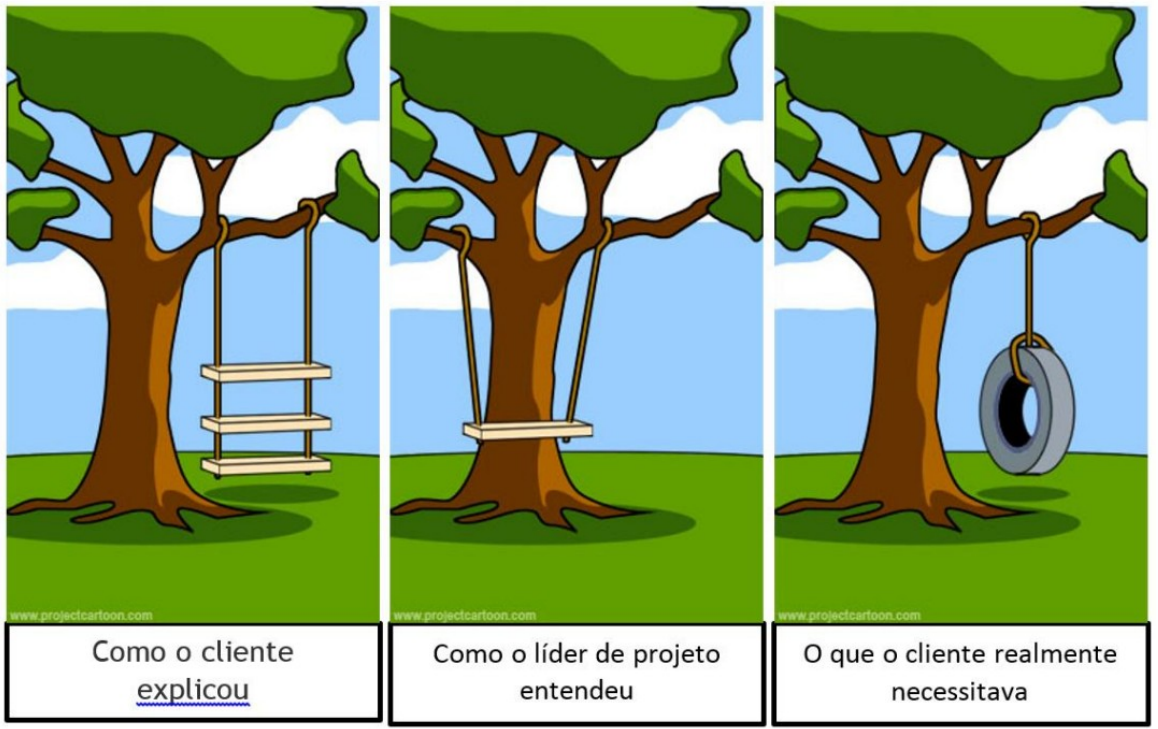


Modelo de processo

- Modelo cascata
  - Estabelecer os requisitos básicos para todos elementos que envolve o software, como hardware, pessoas e banco de dados;
  - Exige uma intensa comunicação entre o analista e o cliente;
  - Faz parte da análise do sistema;

Modelo de processo

- Análise dos requisitos



Modelo de processo

- Análise dos requisitos
  - Intensifica-se o processo de coleta dos requisitos;
  - Identifica as funções necessárias, o desempenho e interfaces exigidas (Funcionalidades e restrições);
  - Os requisitos para o sistema são documentados e revisto com o cliente;
  - Produz a especificação dos requisitos;
  - Faz parte da análise do sistema.

Modelo de processo

- Análise dos requisitos – Engenharia de Requisitos
  - A Engenharia de Requisitos (ER) é o ramo da Engenharia de Software que envolve as atividades relacionadas com a definição dos requisitos de software de um sistema, desenvolvidas ao longo do ciclo de vida de software.
  - O processo de ER envolve criatividade, interação entre pessoas, conhecimento e experiência para transformar informações diversas (sobre a organização, sobre leis, sobre o sistema a ser construído etc.) em documentos e modelos que direcionem o desenvolvimento de software.

Modelo de processo

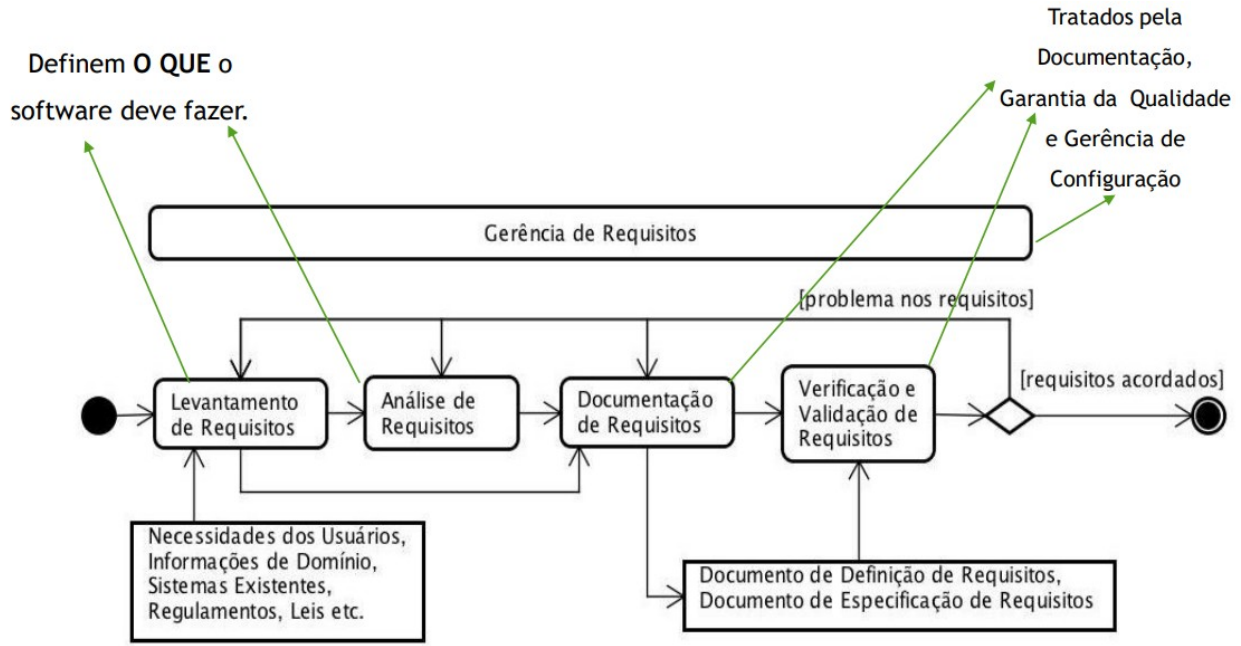
- Análise dos requisitos – Engenharia de Requisitos
  - A Engenharia de Requisitos é fundamental, pois possibilita, dentre outros, estimar custo e tempo de maneira mais precisas e melhor gerenciar mudanças em requisitos.
  - Dentre os problemas de um processo de engenharia de requisitos ineficiente, podem-se citar: (i) requisitos inconsistentes, (ii) produto final com custo maior do que o esperado, (iii) software instável e com altos custos de manutenção e (iv) clientes insatisfeitos.

Modelo de processo

- Análise dos requisitos – Engenharia de Requisitos
  - Alguns benefícios que um processo de ER de qualidade pode trazer são:
    - Menor quantidade de defeitos nos requisitos;
    - Redução de retrabalho;
    - Desenvolvimento de menos características desnecessárias;
    - Diminuição de custos;
    - Desenvolvimento mais rápido;
    - Menos problemas de comunicação;
    - Alterações de escopo reduzidas;
    - Estimativas mais confiáveis;
    - Maior satisfação de clientes e desenvolvedores

Modelo de processo

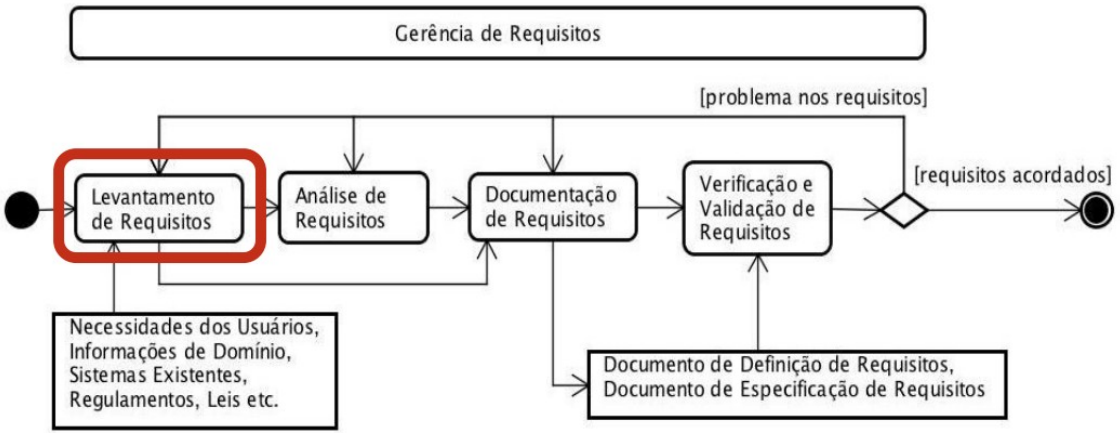
- Análise dos requisitos – Engenharia de Requisitos



Ao conjunto de atividades relacionadas aos requisitos, dá se o nome de **Engenharia de Requisitos**.

Modelo de processo

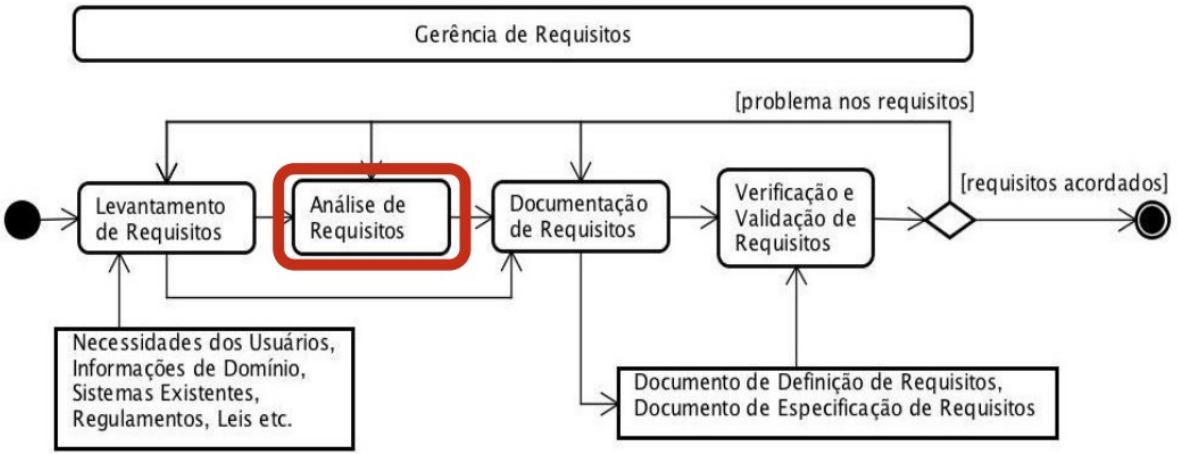
- Análise dos requisitos – Engenharia de Requisitos



Nesta fase, os usuários, clientes e especialistas de domínio são identificados e trabalham junto com os engenheiros de requisitos para entender a organização, o domínio da aplicação, os processos de negócio a serem apoiados, as necessidades que o software deve atender e os problemas e deficiências dos sistemas atuais. Os diferentes pontos de vista dos participantes do processo, bem como as oportunidades de melhoria, restrições existentes e problemas a serem resolvidos devem ser levantados.

Modelo de processo

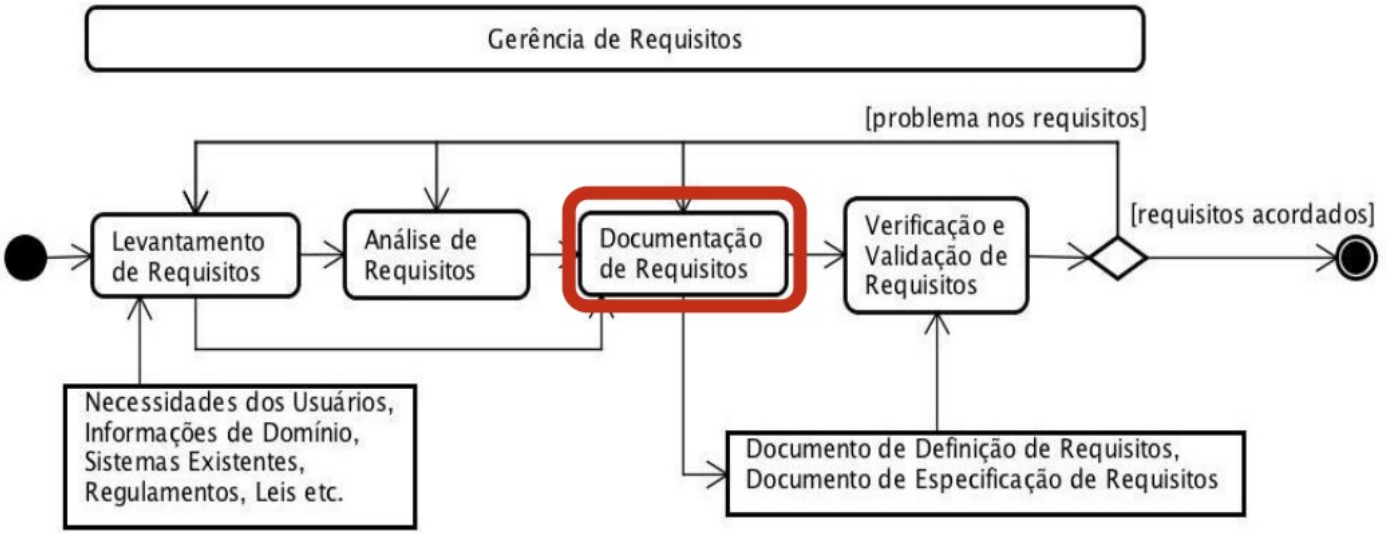
- Análise dos requisitos – Engenharia de Requisitos



Esta fase visa estabelecer um conjunto acordado de requisitos consistentes e sem ambiguidades, que possa ser usado como base para as atividades subsequentes do processo de software. Para tal, diversos tipos de modelos são construídos. Assim, a análise de requisitos é essencialmente uma atividade de modelagem. A análise de requisitos pode incluir, ainda, negociação entre usuários para resolver conflitos detectados.

Modelo de processo

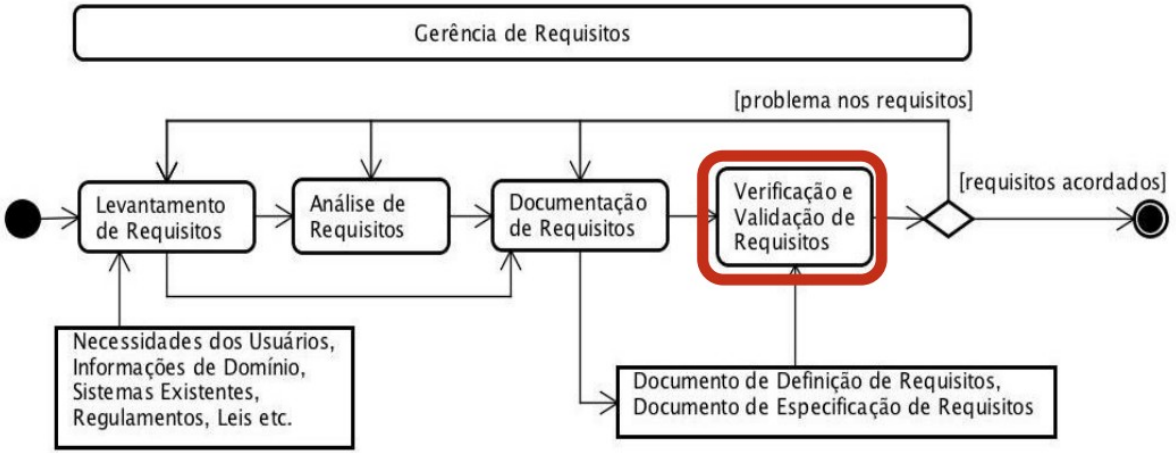
- Análise dos requisitos – Engenharia de Requisitos



Esta fase é a atividade de representar os resultados da Engenharia de Requisitos em um documento (ou conjunto de documentos), contendo os requisitos do software e os modelos que os especificam.

Modelo de processo

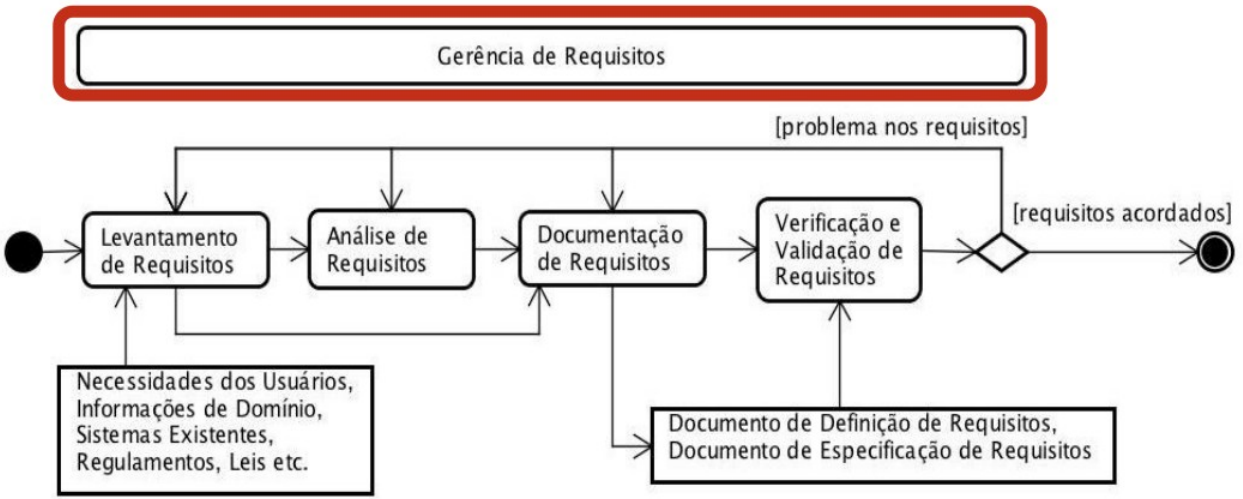
- Análise dos requisitos – Engenharia de Requisitos



A **verificação** de requisitos avalia se os requisitos estão sendo tratados de forma correta, de acordo com padrões previamente definidos, sem conter requisitos ambíguos, incompletos ou, ainda, requisitos incoerentes ou impossíveis de serem testados. Já a **validação** diz respeito a avaliar se os requisitos do sistema estão corretos, ou seja, se os requisitos e modelos documentados atendem às reais necessidades de usuários e clientes.

Modelo de processo

- Análise dos requisitos – Engenharia de Requisitos



Essa fase se preocupa em gerenciar as mudanças nos requisitos já acordados, manter uma trilha dessas mudanças e gerenciar os relacionamentos entre os requisitos e as dependências entre requisitos e outros artefatos produzidos no processo de software, de forma a garantir que mudanças nos requisitos sejam feitas de maneira controlada e documentada.

Modelo de processo

- Análise dos requisitos – Engenharia de Requisitos
  - Para levantar quais são os requisitos de um sistema, devem-se obter informações dos interessados (stakeholders), consultar documentos, obter conhecimentos do domínio e estudar o negócio da organização. Neste contexto, quatro dimensões devem ser consideradas:

Modelo de processo

- Análise dos requisitos – Engenharia de Requisitos
  - Alguns problemas tornam o levantamento de requisitos uma tarefa difícil:
    - Problemas de escopo: as fronteiras do sistema são mal definidas ou os clientes/usuários especificam detalhes técnicos desnecessários que podem confundir, em vez de esclarecer, os objetivos globais do sistema.

Modelo de processo

- Análise dos requisitos – Engenharia de Requisitos
  - Problemas de entendimento:
    - Os clientes/usuários não estão completamente certos do que é necessário;
    - Têm pouca compreensão das capacidades e limitações de um sistema computacional;
    - Não têm pleno entendimento do domínio do problema;
    - Têm dificuldade de comunicar suas necessidades;
    - Omitem informação que acreditam ser óbvia;
    - Especificam requisitos que conflitam com as necessidades de outros clientes/usuários ou
    - Especificam requisitos que são ambíguos ou impossíveis de testar.

Modelo de processo

- Análise dos requisitos – Engenharia de Requisitos
  - Problemas de volatilidade:
    - Os requisitos mudam ao longo do tempo;
    - Pode ser difícil compreender e coletar informações quando existem muitos termos desconhecidos, manuais técnicos etc.
    - Pessoas que entendem o problema a ser resolvido podem ser muito ocupadas e não ter muito tempo para, juntamente como analista, levantar os requisitos e entender o sistema.
    - Políticas organizacionais podem influenciar nos requisitos de um sistema.
    - Os interessados não sabem muito bem o que querem do sistema e não conhecem muitos termos.

Modelo de processo

- Análise dos requisitos – Engenharia de Requisitos
  - Diversas técnicas podem ser utilizadas no levantamento de requisitos, as quais podem possuir diferentes objetos de investigação ou podem ter foco em tipos diferentes de requisitos;
  - Entrevistas: técnica amplamente utilizada, que consiste em conversas direcionadas com um propósito específico e com formato “pergunta-resposta”;
    - Seu objetivo é descobrir problemas a serem tratados, levantar procedimentos importantes e saber a opinião e as expectativas do entrevistado sobre o sistema.

Modelo de processo

- Análise dos requisitos – Engenharia de Requisitos
  - Questionários: o uso de questionários possibilita ao analista obter informações como postura, crenças, comportamentos e características de várias pessoas que serão afetas pelo sistema.
  - Observação: consiste em observar o comportamento e o ambiente dos indivíduos de vários níveis organizacionais. Utilizando-se essa técnica, é possível capturar o que realmente é feito e qual tipo de suporte computacional é realmente necessário.

Modelo de processo

- Análise dos requisitos – Engenharia de Requisitos
  - Ajuda a confirmar ou refutar informações obtidas com outras técnicas e ajuda a identificar tarefas que podem ser automatizadas e que não foram identificadas pelos interessados.
  - Análise de documentos: pela análise de documentos existentes na organização, analistas capturam informações e detalhes difíceis de conseguir por entrevista e observação. Documentos revelam um histórico da organização e sua direção.

Modelo de processo

- Análise dos requisitos – Engenharia de Requisitos
  - Cenários: com o uso desta técnica, um cenário de interação entre o usuário final e o sistema é montado e o usuário simula sua interação com o sistema nesse cenário, explicando ao analista o que ele está fazendo e de que informações ele precisa para realizar a tarefa descrita no cenário;
    - O uso de cenários ajuda a entender requisitos, a expor o leque de possíveis interações e a revelar facilidades requeridas.

Modelo de processo

- Análise dos requisitos – Engenharia de Requisitos
  - Prototipagem: um protótipo é uma versão preliminar do sistema, muitas vezes não operacional e descartável, que é apresentada ao usuário para capturar informações específicas sobre seus requisitos de informação, observar reações iniciais e obter sugestões, inovações e informações para estabelecer prioridades e redirecionar planos.
    - Exemplos de ferramentas de prototipagem: Justinmind, InVision, Origami, Proto.io, Fluid, Marvel, NinjaMock, UXPin, AdobeXD, Sketch, Figma....

Modelo de processo

- Análise dos requisitos – Engenharia de Requisitos
  - Dinâmicas de Grupo: há várias técnicas de levantamento de requisitos que procuram explorar dinâmicas de grupo para a descoberta e o desenvolvimento de requisitos, tais como Brainstorming e JAD (Joint Application Development).
    - Na primeira, representantes de diferentes grupos de interessados engajam-se em uma discussÃ£o informal para rapidamente gerarem o maior nÃºmero possÃvel de ideias.

Modelo de processo

- Tipos de Requisitos
  - Requisitos Funcionais (RF)
    - Descreve as funcionalidades do Sistema, isto é, o que o Sistema deve fazer.
    - Exemplos:
      - RF 001 - O Sistema deve listar todos os alunos cadastrados em uma turma.
      - RF 002 - O Sistema deve calcular a média dos alunos da turma.

Modelo de processo

- Tipos de Requisitos
  - Requisitos de qualidade ou não Funcionais (RNF)
    - Expressam restrições ou limites que o Sistema deve atender ou qualidades especificas que sistema deve possuir.
    - Onde mais aparecem os RNF:
      - Critérios de Usabilidade;
      - Desempenho;
      - Segurança;
      - Restrições de Hardware e Software;
      - Questões sobre padronização e normatização;
      - Questões de distribuição e instalação.

Modelo de processo

- Tipos de Requisitos
  - Requisitos de qualidade ou não Funcionais (RNF)
    - Exemplo:
      - RNF 1: O Sistema deve emitir o relatório da média dos alunos em no máximo 5 segundos;
      - RNF 2: O Sistema deve ser executado no Sistema Operacional Windows 7 ou superior e Linux Ubuntu;
      - RNF 3: O produto serÃ¡ desenvolvido para mÃ¡quinas com pelo menos 1 GB de Ram.

Modelo de processo

- Tipos de Requisitos
  - Regras de negócio (RNE):
    - Descrevem como uma dada funcionalidade deve ser realizada.
    - Exemplos:
      - RNE 01: A média para aprovação na instituição é 6.
      - RNE 02: Um Professor pode lecionar em uma turma e ser aluno em outra.

Modelo de processo

- Os requisitos detectados devem ser:
  - Claros;
  - Bem escritos;
  - Sem ambiguidade;
  - Implementáveis.

Modelo de processo

- Como documentar o levantamento de requisitos

3.1. [RF001] Registrar avaliaçao de disciplina por um discente.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

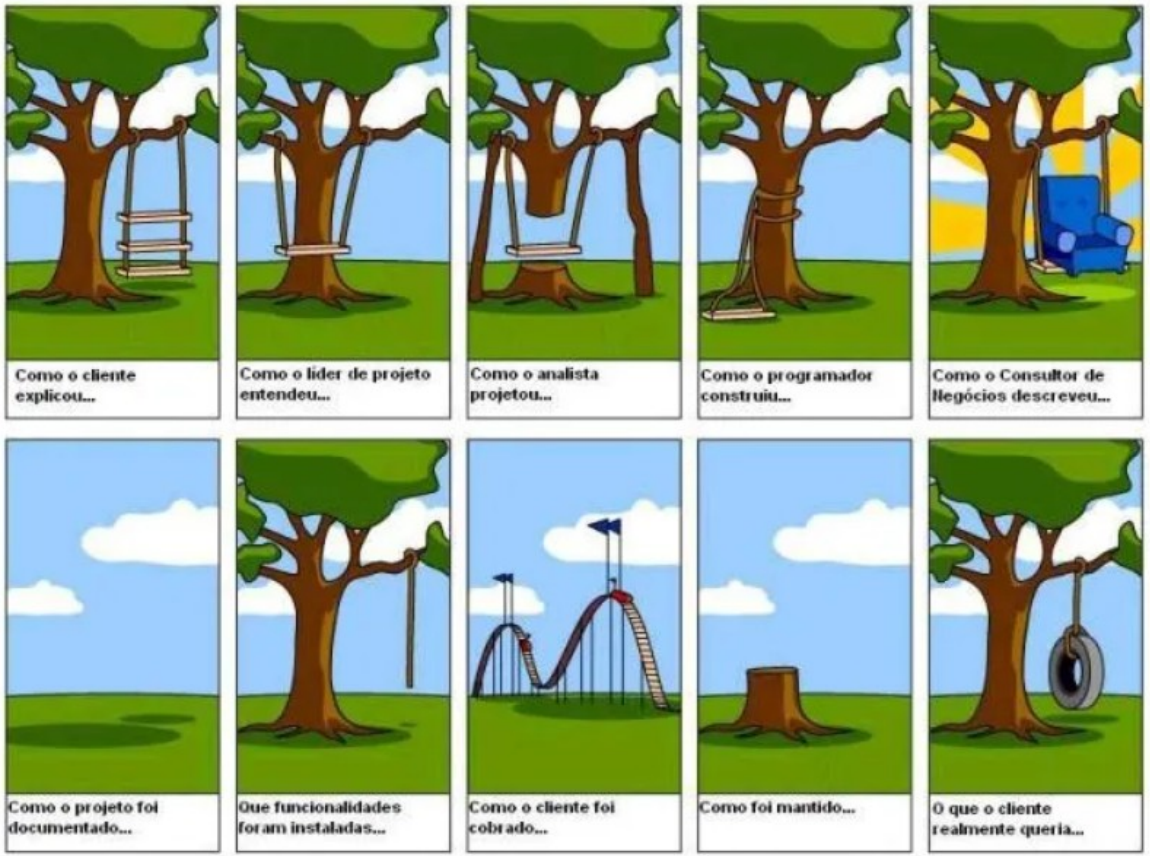
O sistema deve permitir ao discente em uma única tela, a avaliaçao de todas as disciplinas cursadas no período.

Modelo de processo

- Como documentar o levantamento de requisitos

RS003	**Registro de avaliação de turma**
Referência	[Registrar avaliação de turma por docente.RF003], [Impedir acesso direto ao SIGA.RF006], [Bloquear acesso direto ao SIGA.RF005]
Sumário	O caso de uso é responsável por registrar a avaliação de turma.
Pré-condições	O usuário deve estar ministrando as turmas das disciplinas avaliadas no período, o período de avaliação de turmas deve estar aberto.
Atores	Docente
Descrição	1. O usuário faz login no SIGA. 2. O sistema exibe uma tela, antes da tela principal do SIGA, contendo um formulário em forma de matriz de perguntas (linhas) x codigo de turma (colunas), um botão para "registrar avaliação" e um botão para "responder depois". As turmas exibidas são todas as ministradas pelo docente no período avaliado. 3. O usuário faz clique sobre o botão responder. 4. O sistema registra avaliação.
Alternativas	No passo 2, o botão "responder depois" ficará desativado a partir de determinada data de acordo com **regras definidas pela DIAVI.**
Exceção	O registro da avaliação não poderá ser concluído caso o usuário deixe de preencher algum campo.

Modelo de processo



Falha na especificação de Requisitos

Modelo de processo

- Projeto
  - Traduz o requisitos em um conjunto de representações que podem ser avaliadas quando a qualidade;
    - Estrutura de dados;
    - Arquitetura de software;
    - Detalhes procedimentais;
    - Caracterização da interface.
  - É avaliado antes de começar a ser implementado;
  - Junto com as etapas anteriores torna-se parte da documentação do sistema.

Modelo de processo

- Codificação
  - Projeto traduzido para a linguagem do computador (C, C++, Pascal, Delphi, Java, PHP, Python, etc);
  - Se o projeto for executado detalhadamente, a codificação pode ser executada mecanicamente?

Modelo de processo

- Testes
  - Concentra-se nos aspectos funcionais externos e lógicos internos do software;
  - Garante que todas as instruções tenham sido testados;
  - A entrada definida produz os resultados exigidos?

Modelo de processo

- Manutenção
  - Provavelmente o software deverá sofrer mudanças depois que for entregue ao cliente;
  - Tipo de manutenção
    - **Manutenção Corretiva:** diagnóstico e correção de erros;
    - **Manutenção adaptativa:** adaptação do software para acomodar mudanças em seu ambiente externo;
    - **Manutenção perfectiva:** exigência do cliente para acrescimo de funcionalidades e de desempenho;
    - **Manutenção preventiva:** melhorar a confiabilidade e manutenibilidade futura.

Modelo de processo

- Problemas
  - Projetos reais raramente seguem o fluxo sequencial que ele propõe. Ocorrem iterações que trazem problemas na aplicação do paradigma;
  - É difícil para o cliente declarar todas as exigências explicitamente. É difícil acomodar as incertezas naturais que existem no começo de muitos projetos;
  - O cliente deve ter paciência. Uma versão do software só estará disponível em um ponto tardio do cronograma. Um erro pode ser desastroso;

Modelo de processo

- Problemas
  - Desenvolvedores ocioso;
  - Só é apropriado quando os requisitos são bem conhecidos.

Modelo de processo

- Prototipação
  - Apropriado quando
    - O cliente definiu um conjunto de objetivos gerais para o software, mas, não definiu os requisitos de entrada, processamento e saída com detalhes;
    - Desenvolvedor não tem certeza da eficiência de um algoritmo, forma de interação homem/máquina;
    - Permite o refinamento iterativo dos requisitos;
    - Cada iteração é produzido um protótipo do software final;

Modelo de processo

- Prototipação
  - Apropriado quando
    - Este protótipo pode ser um:
      - Protótipo em papel, primeiras versões que permitem ao usuários ter uma visão abstrata do sistema;
      - Protótipo incompleto, implementa algum subconjunto de funções exigidas;
      - Protótipo final, um software que executa parte ou toda a função desejada, mas, que tem outras características que seram melhoradas e ainda não podem ser disponibilizado.

Modelo de processo

- Prototipação

Modelo de processo

- Coleta e refinamento dos requisitos;
  - Nesta etapa o desenvolvedor e o cliente deve definir os objetivos gerais do software;
  - Identificar quais requisitos são conhecidos e as áreas que necessitam de definição adicional;
  - Análise de sistema.
- Projeto rápido
  - Representação dos aspecto do software que são visíveis ao usuários;

Modelo de processo

- Construção do protótipo
  - Implementação rápida do projeto.
- Avaliação do protótipo
  - Cliente e desenvolvedor avaliam o protótipo;
  - No caso de sugestão ou mudanças serão trabalhadas na próxima fase.
- Refinamento do protótipo
  - São trabalhados os problemas encontrados na fase anterior.
  - Neste ponto pode ocorrer, no caso de necessidade de alterações, um retorno na fase de projeto rápido para desenvolver novo protótipo que incorpora as mudanças;

Modelo de processo

- Construção do produto
  - Identifica todos os requisitos necessários, o protótipo pode ser descartado e a versão final deve ser construída considerando os critérios de qualidade.

Modelo de processo

- Poblemas
  - O cliente muitas vezes não aceita mais uma iteração, aquela versão mesmo incompleta já serve;
  - Não há necessidade de desenvolver uma versão final, modifica-se o protótipo;
  - Desenvolver frequentimente faz uma implementação comprometida com o objetivo de produzir rapidamente o protótipo.
- Solução
  - Definir as regras do jogo logo no início, o cliente deve concordar que o protótipo seja construído para servir como um mecanismo a fim de definir os requisitos.

Modelo de processo

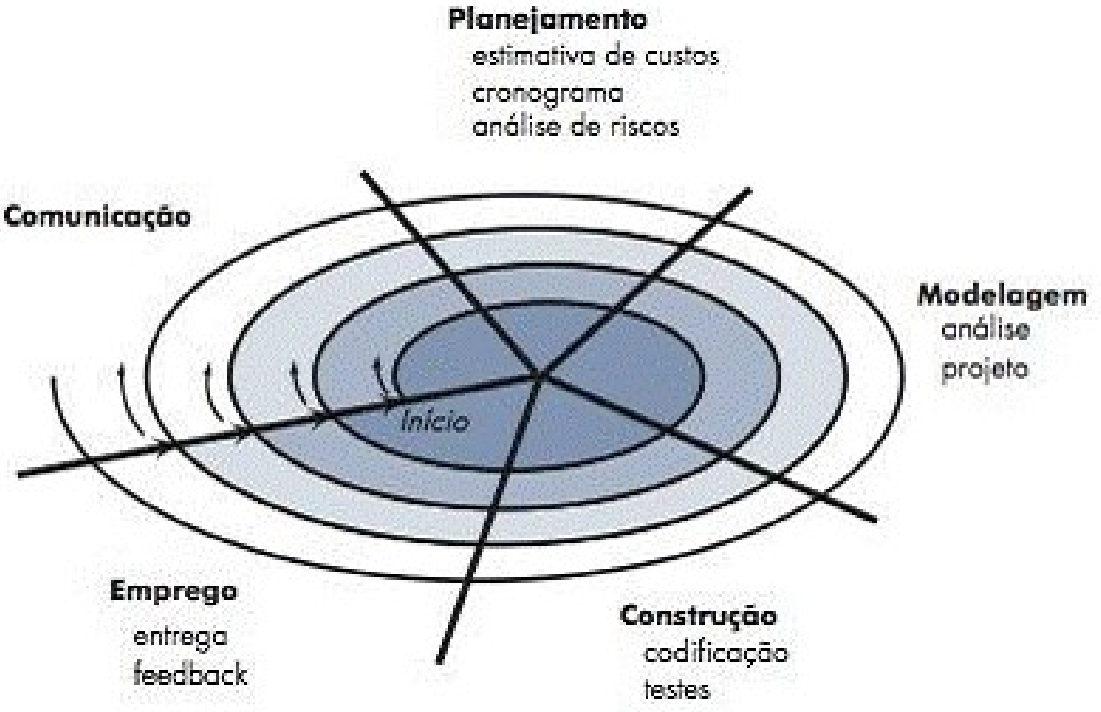
- Modelo Espiral
  - O modelo espiral é um modelo incremental;
  - Este tipo de modelo combina elementos do modelo cascata com a filosofia iterativa da prototipação;
  - Acrescenta mais uma atividade, análise de risco;
  - O objetivo é trabalhar junto do usuário para descobrir os seus requisitos, de maneira incremental, até que o produto final seja obtido;
  - O desenvolvimento com as partes do produto que são bem entendidas;

Modelo de processo

- Modelo Espiral
  - A evolução acontece quando novas características são adicionadas à medida que são sugeridas pelo usuário;
  - A cada iteração é desenvolvida uma versão usável, não um protótipo.

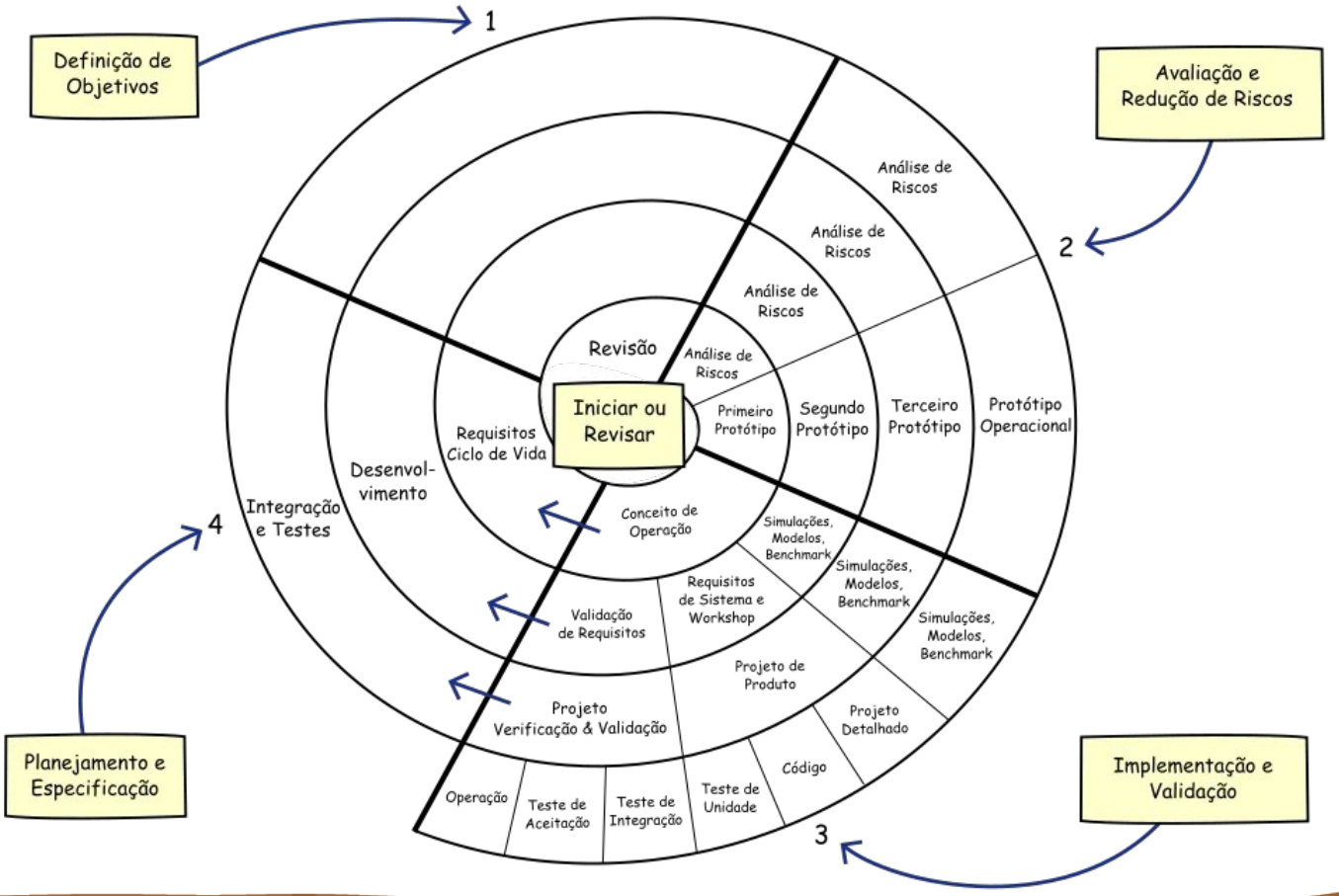
Modelo de processo

- Modelo Espiral



Modelo de processo

- Modelo Espiral



Modelo de processo

- Modelo espiral
  - Fornece o potêncial para o desenvolvimento rápido de versão incrementais do software;
  - Nas primeiras iterações são desenvolvidas versões que podem ser protótipos;
  - Nas iterações mais adiantadas são produzidas versões incrementais mais completas e melhoradas.

Modelo de processo

- Definição de objetivos
  - Onde são definidos os objetivos para essa fase do projeto, identificando as restrições e preparando um plano de gerenciamento detalhado que inclui todos os possíveis riscos do projeto;
- Avaliação e redução de riscos
  - Para cada risco identificado é feita uma “Análise de Risco” detalhada com o objetivo de identificar estratégias para reduzi-lo ou evitá-lo.

Modelo de processo

- Avaliação e redução de riscos
  - Por exemplo, caso exista uma dificuldade em especificar claramente um requisito, isso significa que existe um “risco de requisitos inadequados” e para amenizá-lo será preciso desenvolver um protótipo para apresentar ao cliente a fim de colher sugestões para refinar os requisitos.
- Implementação e validação:
  - Com as estratégias definidas, é escolhido um modelo de desenvolvimento, como por exemplo, o “Modelo em Cascata”, “Modelo Incremental”, etc. Pode-se utilizar modelos diferentes em cada volta de implementação, conforme a necessidade.

Modelo de processo

- Implementação e validação:
  - Desenvolvimento do produto no nível seguinte;
  - Constroi o protótipo ou versões mais avançadas do produto;
  - Realiza teste, implementação, suporte.

Modelo de processo

- Planejamento e Especificação:
  - O projeto todo é analisado para verificar o que foi realizado e planejar quais serão os próximos passos para iniciar novas voltas do espiral ou concluir o sistema.
    - Determinação dos objetivos, alternativas e restrições;
    - Comunicação com o cliente;
    - Definição de recursos.

Modelo de processo

- Avaliação do cliente
  - Obter um feedback do cliente baseado na avaliação da versão do software;
  - São levantado as necessidades de mudanças para o software.

Modelo de processo

- Modelo espiral
  - É uma abordagem realística para o desenvolvimento de software em grande escala;
  - Usa uma abordagem que capacita o desenvolvedor e o cliente a entender a reagir aos risco em cada etapa evolutiva;
  - Pode ser difícil convencer os clientes que uma abordagem evolutiva é controlável;
  - Exige considerável experiência na determinação de risco e depende da experiência para ter sucesso.

Modelo de processo

- Outros modelos
  - RAD(Rapid Application Development)
    - Adaptação do modelo cascata;
    - Modelo de desenvolvimento de software incremental que enfatiza um ciclo de desenvolvimento bastante curto (60 a 90 dias);
    - Desenvolvimento em equipe team e modular;
    - Fases: Modelagem de negócio, modelagem dos dados, modelagem do processo, geração de aplicação, teste e implementação.

Modelo de processo

- Outros modelos
  - DBC (Desenvolvimento baseado em componentes)
    - Evolução da tecnologia OO;
    - Adaptação do modelo espiral para o desenvolvimento de software;
    - Modelo de componentes: CORBA (Common Object Request Broker Architecture), COM (Common Object Model Microsoft), UML (Unified Modeling Language).
    - Componentes são construídos/empacotados para ser;m reutilizados em diferentes aplicações.

Modelo de processo

- Outros modelos
  - XP (Exterme Programming)
    - O processo consiste basicamente num ciclo de quatro etapas, que itera até o produto esteja pronto.
      - Planejamento: Coleta e negociação de requisitos como cliente;
      - Teste: Elaboração de testes com base nas estórias do cliente;
      - Implementação: codificação do sistema de modo a atender os testes;

Modelo de processo

- Outros modelos
  - XP (Exterme Programming)
    - O processo consiste basicamente num ciclo de quatro etapas, que itera até o produto esteja pronto.
      - Desenho: reconstrução do sistema para a incorporação de novas funcionalidades.

Modelo de processo

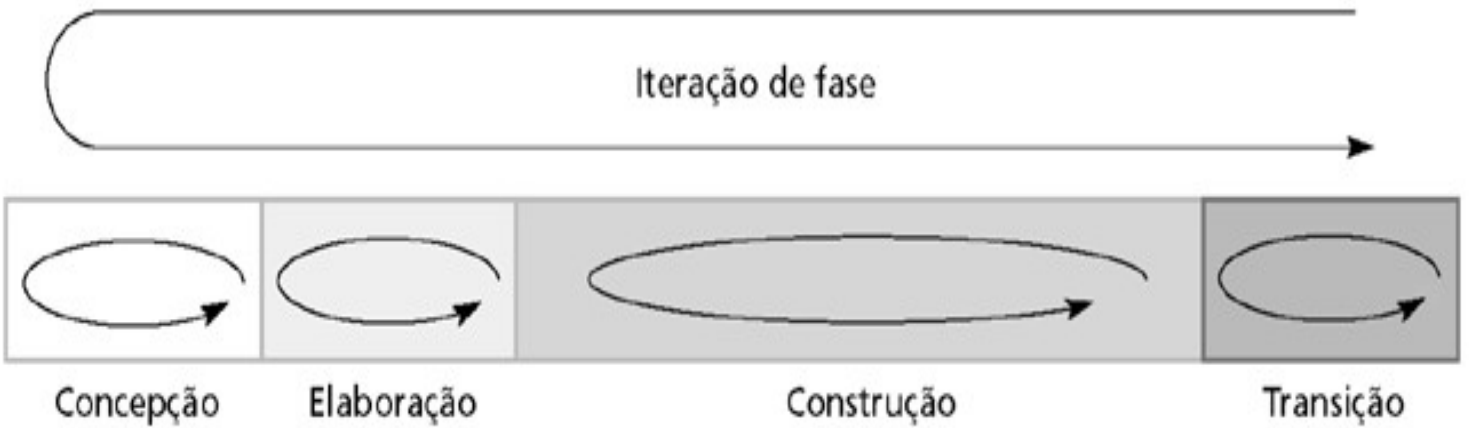
- Outros modelos
  - RUP (Rational Unified Process)
    - É um processo de engenharia de software desenvolvido pela empresa Rational;
    - Ele serve como um guia de como utilizar de maneira eficiente a UML;
    - Utiliza desenvolvimento iterativo e incremental;
    - Tem como objetivo oferecer um processo de desenvolvimento bem definido e bem gerido.

Modelo de processo

- Outros modelos
  - RUP (Rational Unified Process)

Figura 4.12

Fases no Rational Unified Process.



Qualidade

Qualidade

- Grau em que um produto de software atende às expectativas e necessidades dos usuários, além de estar em conformidade com requisitos e padrões estabelecidos.
- Basicamente o que fazem é:
  - Testam o software: testa o software para ter certeza de que ele funciona corretamente. Pressionam todos os botões, clicam em todos os links e tentam todas as combinações possíveis para verificar se algo quebra.

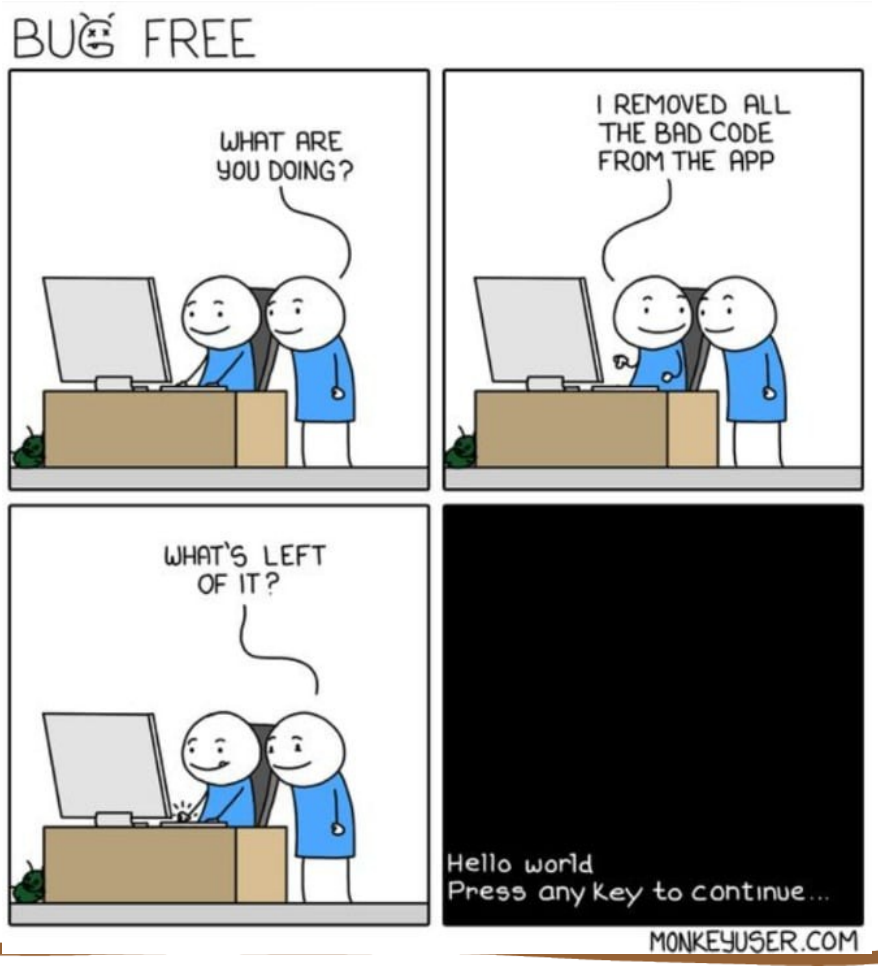
Qualidade

- Basicamente o que fazem é:
  - Eliminam bugs: procurar por bugs no software. Encontram erros que podem fazer o software travar ou não funcionar como deveria.
  - Seguem as especificações: verificam se o software está sendo feito de acordo com as especificações. Se algo estiver fora do lugar, alertam a equipe de desenvolvimento.
  - Tornam o user-friendly: certificam-se de que o software seja fácil de usar e agradável aos olhos. Se algo não estiver bonito ou intuitivo, sugerem melhorias.
  - Dão feedback a devs: fornecem feedback a devs,isso ajuda a equipe de desenvolvimento a corrigir problemas e aprimorar o software.

Qualidade

- Analista de qualidade pode variar, mas geralmente envolve:
  - Testar diferentes partes do software;
  - Documentar problemas;
  - Trabalhar em estreita colaboração com pessoas desenvolvedoras para solucionar os problemas encontrados;
  - Garantir que o software esteja pronto para ser entregue aos clientes.

Qualidade


BUG FREE
WHAT ARE YOU DOING?
I REMOVED ALL THE BAD CODE FROM THE APP
WHAT'S LEFT OF IT?
Hello world
Press any key to continue...
MONKEYUSER.COM

Qualidade

- Como aplicar?
  - A aplicação da qualidade de software envolve uma série de práticas e processos ao longo do ciclo de vida do desenvolvimento de software.
  - Podemos considerar como etapas fundamentais para aplicar a qualidade de software:
    - Definir requisitos claros: o primeiro passo é estabelecer requisitos de software claros e bem definidos. Isso envolve uma comunicação eficaz com os stakeholders para entender suas necessidades e expectativas. Os requisitos devem ser documentados de maneira detalhada e compreensível.

Qualidade

- Como aplicar?
  - Podemos considerar como etapas fundamentais para aplicar a qualidade de software:
    - Planejamento de qualidade: crie um plano de qualidade de software que descreva como a qualidade será gerenciada ao longo do projeto. Isso inclui a definição de métricas de qualidade, processos de teste e critérios de aceitação.
    - Padrões e conformidade: estabeleça padrões de desenvolvimento de software e assegure-se de que a equipe os siga. Isso inclui diretrizes de codificação, boas práticas de desenvolvimento e conformidade com normas relevantes, como ISO 9001 ou outras normas específicas da indústria.

Qualidade

- Como aplicar?
  - Podemos considerar como etapas fundamentais para aplicar a qualidade de software:
    - Testes e revisões: realize testes de qualidade em diferentes fases do projeto, desde testes unitários até testes de sistema e testes de aceitação. Além disso, conduza revisões de código para identificar problemas antes que eles se tornem críticos.
    - Automação de testes: use ferramentas de automação de testes para aumentar a eficiência e a cobertura dos testes. Isso ajuda a identificar problemas de qualidade de forma mais rápida e consistente.

Qualidade

- Como aplicar?
  - Podemos considerar como etapas fundamentais para aplicar a qualidade de software:
    - Gerenciamento de defeitos: registre, rastreie e priorize os defeitos encontrados durante os testes e revisões. Certifique-se de que os problemas sejam corrigidos e verificados antes do lançamento.
    - Melhoria contínua: estabeleça um ciclo de melhoria contínua, em que você analisa os resultados dos testes, identifica áreas de fraqueza e implementa melhorias nos processos de desenvolvimento e teste.

Qualidade

- Como aplicar?
  - Podemos considerar como etapas fundamentais para aplicar a qualidade de software:
    - Treinamento e desenvolvimento: mantenha sua equipe atualizada com treinamento regular em práticas de desenvolvimento de qualidade e novas tecnologias.
    - Documentação: mantenha uma documentação adequada de todos os processos, testes e mudanças realizadas ao longo do ciclo de vida do software. Isso ajuda na rastreabilidade e na auditoria de qualidade.

Qualidade

- Como aplicar?
  - Podemos considerar como etapas fundamentais para aplicar a qualidade de software:
    - Avaliação externa: considere a realização de auditorias de qualidade externas ou revisões independentes para garantir uma avaliação imparcial da qualidade do software.
    - Envolvimento do cliente: mantenha uma comunicação constante com os clientes para garantir que o software atenda às suas necessidades e expectativas.

Qualidade

- Como aplicar?
  - Podemos considerar como etapas fundamentais para aplicar a qualidade de software:
    - Monitoramento em produção: após o lançamento, monitore continuamente o software em produção para identificar e resolver problemas em tempo real.

Qualidade

- Modelos de qualidade
  - Modelos de qualidade são estruturas e abordagens que ajudam a medir, avaliar e melhorar a qualidade de software;
  - Existem três modelos principais de qualidade de software:
    - Qualidade interna;
    - Qualidade externa;
    - Qualidade em uso.

Qualidade

- Qualidade interna
  - Concentra-se na qualidade do código-fonte e nos processos internos de desenvolvimento do software. Ele avalia como o software é construído, observando fatores como a estrutura do código, a consistência, a modularidade, a legibilidade e a manutenibilidade. Os modelos de qualidade interna ajudam a criação de um software mais fácil de gerenciar, melhorar e expandir. Exemplos de modelos de qualidade interna incluem o ISO/IEC 9126 e o modelo de qualidade de McCall.

Qualidade

- Qualidade externa
  - Concentra-se na qualidade percebida pelos usuários finais do software. Ele avalia como o software se comporta em termos de desempenho, confiabilidade, usabilidade e eficácia quando usado em situações reais. Os modelos de qualidade externa medem a qualidade com base em critérios externos e objetivos. Um exemplo é a conformidade com padrões de usabilidade ou a ausência de erros críticos em produção.

Qualidade

- Qualidade em uso
  - Concentra-se na qualidade do software durante seu uso real no ambiente do cliente. Ele leva em consideração a satisfação do usuário, a eficiência, a eficácia e a produtividade alcançada ao usar o software em cenários do mundo real. Os modelos de qualidade em uso ajudam a entender como o software atende às necessidades e expectativas dos usuários e como ele pode ser aprimorado para oferecer uma experiência melhor. Um exemplo é o modelo de qualidade de produto de Garvin.

Qualidade

- Analista de qualidade pode variar, mas geralmente envolve:
  - Testar diferentes partes do software;
  - Documentar problemas;
  - Trabalhar em estreita colaboração com pessoas desenvolvedoras para solucionar os problemas encontrados;
  - Garantir que o software esteja pronto para ser entregue aos clientes.

Métricas de processos

Métricas de processos

- Métricas de qualidade de software são como os termômetros que usamos para medir a saúde do nosso corpo, mas, neste caso, medem a saúde do software.
- Principais métricas:
  - Taxa de bugs: imagine que o software é como um carro e os bugs são como pequenos problemas nele, por exemplo, faróis quebrados ou pneus murchos. A taxa de bugs mede quantos desses problemas existem no software. Quanto menos bugs, melhor!

Métricas de processos

- Principais métricas:
  - Tempo de resposta: Mede quanto tempo leva para uma ação acontecer quando clicamos em um botão. Menos tempo de resposta é melhor, porque as coisas acontecem mais rapidamente.
  - Confiabilidade: Mede quão confiável ele é. Um software confiável não trava ou apresenta erros com frequência.
  - Usabilidade: refere-se à facilidade de usar o software. Um software fácil de usar é mais agradável.

Métricas de processos

- Principais métricas:
  - Eficiência: É medir quão bem ele utiliza os recursos do computador, como memória e processador. Um software eficiente é rápido e não consome muitos recursos.
  - Segurança: Mede quão bem ele protege suas informações pessoais e se impede o acesso não autorizado.
  - Facilidade de manutenção: Mede a simplicidade de fazer atualizações e correções sem causar mais problemas.