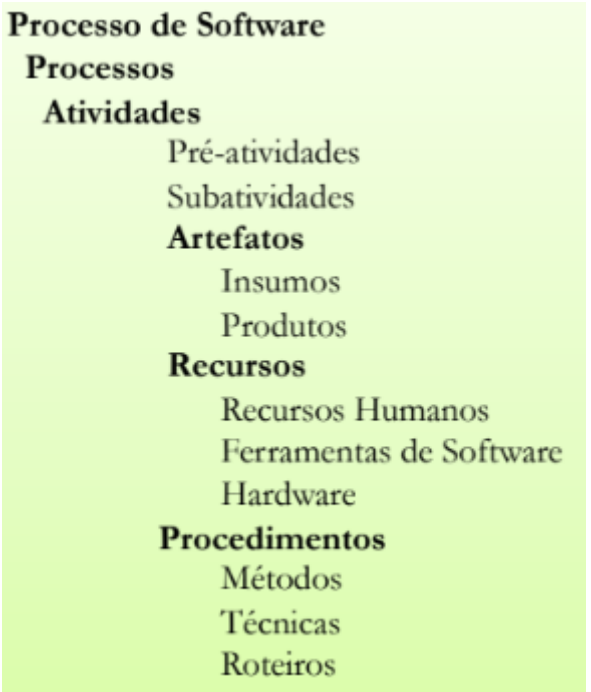
Introdução à Engenharia de Software

Engenharia de Software trata de aspectos relacionados ao estabelecimento de processos, métodos, técnicas, ferramentas e ambientes de suporte ao desenvolvimento de software.
Propõe a divisão do problema em problemas menores, cujas soluções devem ser integradas por uma arquitetura e devem ser obtidas utilizando-se procedimentos (métodos, técnicas, roteiros etc.), bem como ferramentas que automatizam o trabalho (ou parte dele).

Qualidade do Produto de Software x Qualidade do Processo do Software
Processos bem estabelecidos, que incorporam mecanismos sistemáticos para acompanhar o desenvolvimento e avaliar a qualidade, no geral, conduzem a produtos de qualidade.

Processo
É um conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software, devendo considerar as relações entre as atividades,os artefatos produzidos no desenvolvimento, as ferramentas, os procedimentos necessários, a habilidade, o treinamento e a motivação do pessoal envolvido.

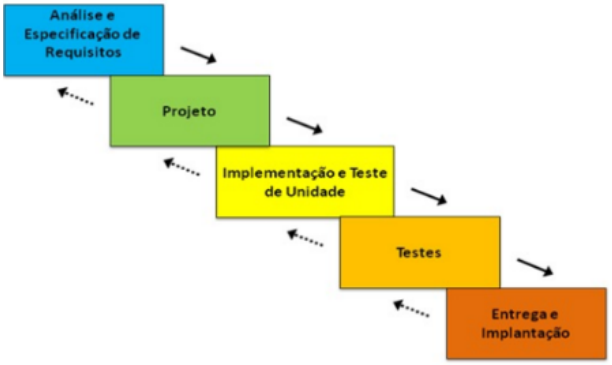


Modelo de Ciclo de Vida de Software ou Modelo de Processo de Software
É a representação abstrata de um esqueleto de processo, incluindo tipicamente algumas atividades principais, a ordem de precedência entre elas e, opcionalmente, artefatos requeridos e produzidos.
Envolve as seguintes fases do ciclo de vida de um software:

- Análise e Especificação de Requisitos (O que é o produto e seus requisitos)
- Projeto (Como definir a arquitetura e linguagem, tecnologia e modelos)
- Implementação (Codificação pura e materialização do modelo abstrato do software)
- Testes (Testes de unidade, integração, sistemas e aceitação)
- Entrega e Implantação (Capacitar as pessoas e entregar manuais de usuários)
- Operação
- Manutenção (Corretiva, Adaptativa e Preventiva)

Podem ser agrupados em três categorias principais: modelos sequenciais, modelos incrementais e modelos evolutivos.

Modelos Sequenciais – Modelo Cascata (Modelo de Ciclo de Vida Clássico)
É o modelo de ciclo de vida mais antigo e mais amplamente usado. Indicado para problemas bastante pequenos e bem definidos, onde os desenvolvedores conhecem bem o domínio do problema e os requisitos podem ser claramente estabelecidos.



Características:

- Uma fase só deve ser iniciada após a conclusão daquela que a precede e de maneira sequencial.
- Uma vez que, essas fases se sobrepõem de alguma forma, permite-se um retorno à fase anterior para a correção de erros encontrados.
- A entrega do sistema completo ocorre somente ao final da fase de Entrega e Implantação.
- O uso de revisões ao fim de cada fase permite o envolvimento do usuário.
- Cada fase serve como uma base aprovada e documentada para o passo seguinte, facilitando bastante a gerência de configuração.

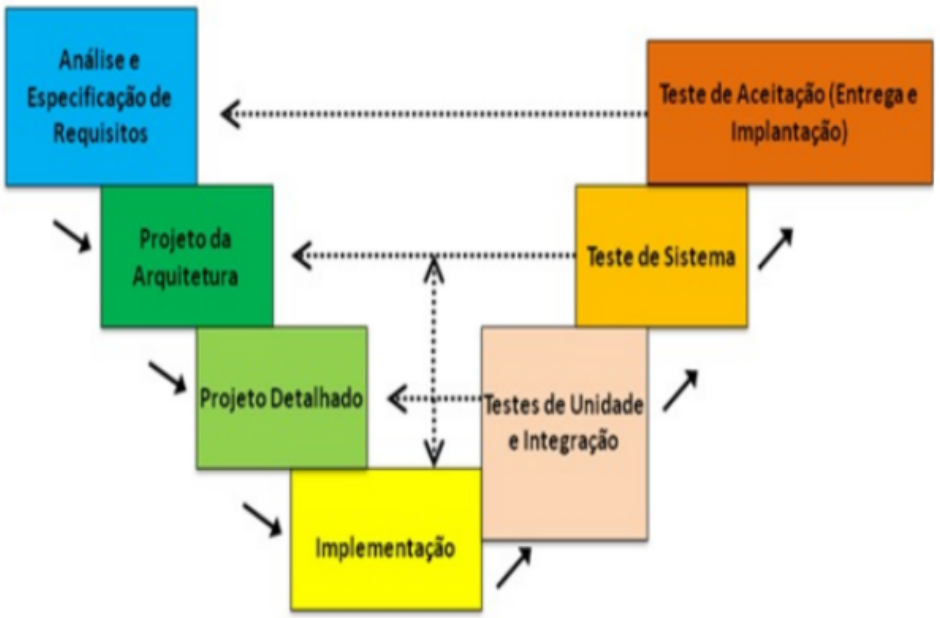
Problemas enfrentados:

- Projetos reais muitas vezes não seguem o fluxo sequencial que o modelo propõe.
- A natureza linear do ciclo de vida clássico leva a “estados de bloqueio” nos quais alguns membros da equipe do projeto precisam esperar que outros membros da equipe completem tarefas dependentes.

Modelos Sequenciais – Modelo em V

É a variação do Cascata que procura enfatizar a estreita relação entre as atividades de teste (teste de unidade, teste de integração, teste de sistema e teste de aceitação) e as demais fases do processo.

O projeto é dividido em dois : Projeto de Arquitetura (Componentes) e Projeto Detalhado (Como funciona esses componentes).



A conexão entre os lados direito e esquerdo do modelo em V implica que, caso sejam encontrados problemas em uma atividade de teste, a correspondente fase do lado esquerdo e suas fases subsequentes podem ter de ser executadas novamente para corrigir ou melhorar esses problemas, ou seja, em caso de erro diferentemente do modelo de cascata, ele consegue voltar em um dos testes para resolver.

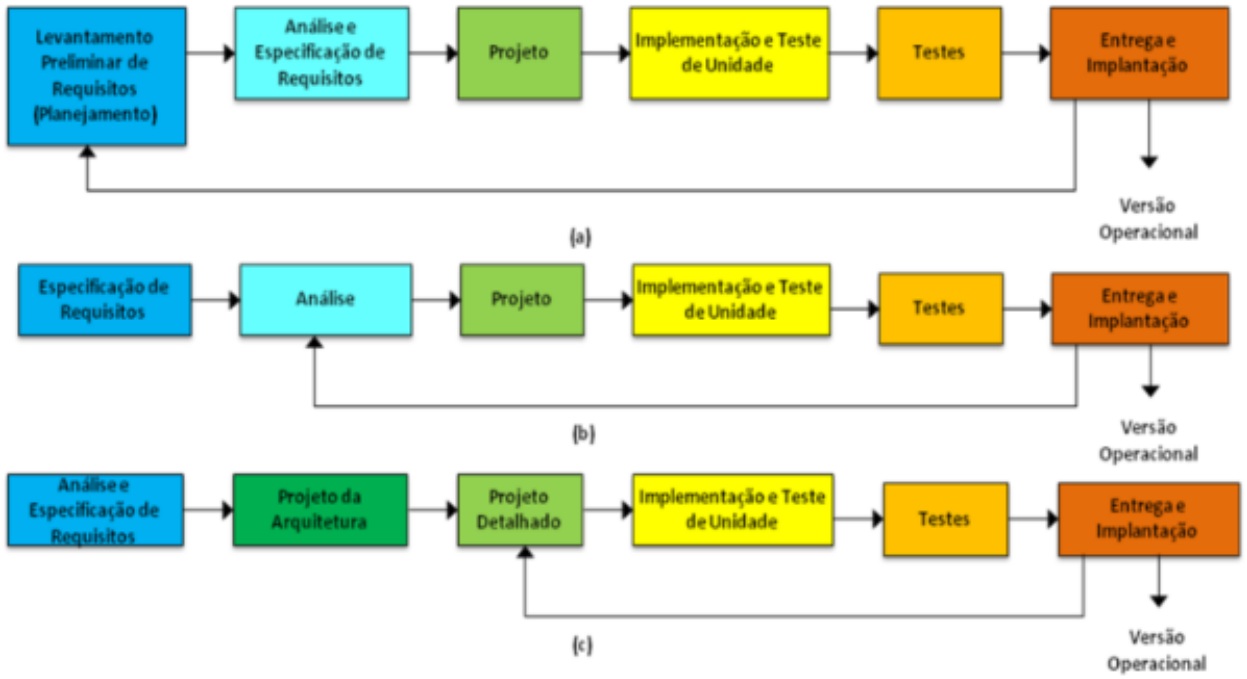
Testes:

- Testes de unidade e integração devem garantir que todos os aspectos do projeto do sistema foram implementados corretamente no código.
- Teste de sistema busca verificar se o sistema atende aos requisitos definidos na especificação.
- Testes de aceitação, conduzidos tipicamente pelos usuários e clientes, validam os requisitos, confirmando que os requisitos corretos foram implementados no sistema .

Modelos Incrementais – Modelo Incremental

Desde o início do projeto é conhecido o escopo/visão total do sistema sendo necessário um retorno rápido.

Seu princípio fundamental é que, a cada ciclo ou iteração, uma versão operacional do sistema será produzida e entregue para uso ou avaliação detalhada do cliente.



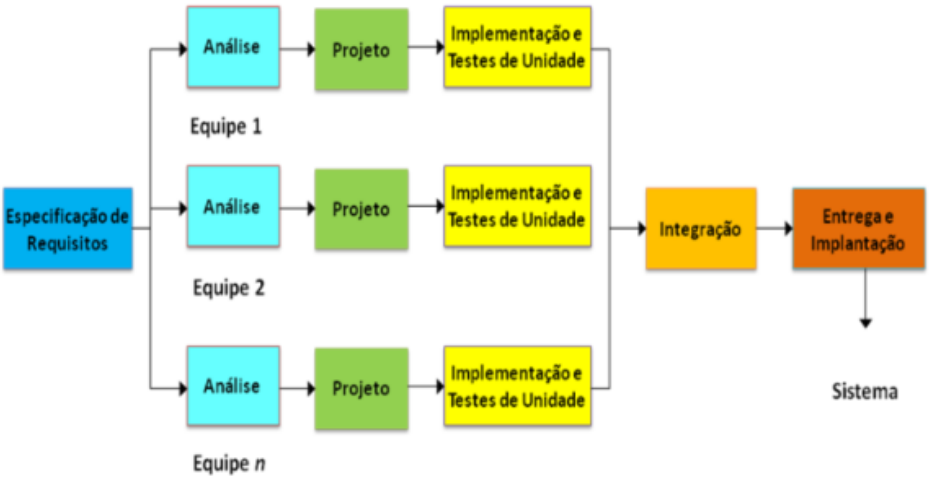
Vantagens:

- Menor custo e menos tempo são necessários para se entregar a primeira versão;
- Os riscos associados ao desenvolvimento de um incremento são menores, devido ao seu tamanho reduzido;
- O número de mudanças nos requisitos pode diminuir devido ao curto tempo de desenvolvimento de um incremento.

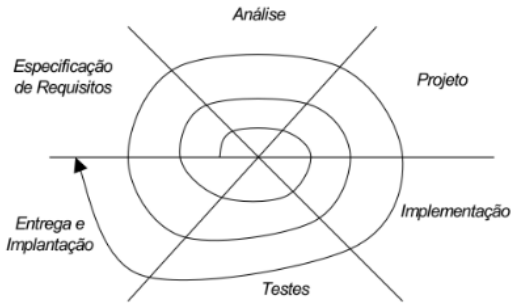
Desvantagens:

- Se os requisitos não são tão estáveis ou completos quanto se esperava, alguns incrementos podem ter de ser bastante alterados;
- A gerência do projeto é mais complexa, sobretudo quando a divisão em subsistemas inicialmente feita não se mostrar boa.

Modelos Incrementais – RAD (Rapid Application Development)
Possui um ciclo de desenvolvimento curto (tipicamente de até 90 dias) , incrementos são desenvolvidos em paralelo por equipes distintas (como vários modelos em cascata ao mesmo tempo) e apenas uma única entrega é feita e os requisitos têm de ser bem definidos com o escopo do projeto estrito e o
sistema modular.



Modelos Evolutivos – Modelo Espiral
O sistema é desenvolvido em ciclos, sendo que nos primeiros ciclos nem sempre todas as atividades são realizadas (por exemplo, o produto resultante do primeiro ciclo pode ser uma especificação do produto ou um estudo de viabilidade).
As passadas subsequentes ao longo da espiral podem ser usadas para desenvolver protótipos, chegando progressivamente a versões operacionais do software, até se obter o produto completo.



Características:

- Utilizado quando não é conhecido o escopo/requisitos total, se inicia desenvolvendo um mini projeto (ao completar a linha espiral) .
- A cada ciclo, o planejamento deve ser revisto com base no feedback do cliente, ajustando, inclusive, onúmero de iterações planejadas.
- Pode ser difícil convencer clientes, especialmente em situações envolvendo contrato, que a abordagem evolutiva é gerenciável.

Prototipação

A prototipação é uma técnica para ajudar engenheiros de software e clientes a entender o que está sendo construído quando os requisitos não estão claros, podendo ser aplicado no contexto de qualquer modelo de processo para identificar claramente as funcionalidades ou informações (requisitos) que o sistema terá de prover ou tratar.

Essa técnica só existe plugadas em outros modelos, utilizado para apoiar o levantamento de pré requisitos , tendo um protótipo como base e evoluindo até se tornar o sistema desejado.

O Processo Unificado - RUP (Rational Unified Process)

É um modelo evolutivo que preconiza o desenvolvimento em ciclos, de modo a permitir uma melhor compreensão dos requisitos, além de um modelo de processo de software, a definição detalhada de responsabilidades(papéis), atividades, artefatos e fluxos de trabalho, dentre outros.

Contém uma dimensão com 4 grandes fases e com dados em formato de “baleias” referente a quantidade de atividades presente neles.