

Ciclo de vida



Ciclo de Vida

- O ciclo de vida de um software descreve as fases pelas quais o software passa desde a sua concepção até ficar sem uso algum;
 - O SDLC (*Software Development Life Cycle*) simplesmente descreve todas as tarefas necessárias para criar e implantar um aplicativo de software
-

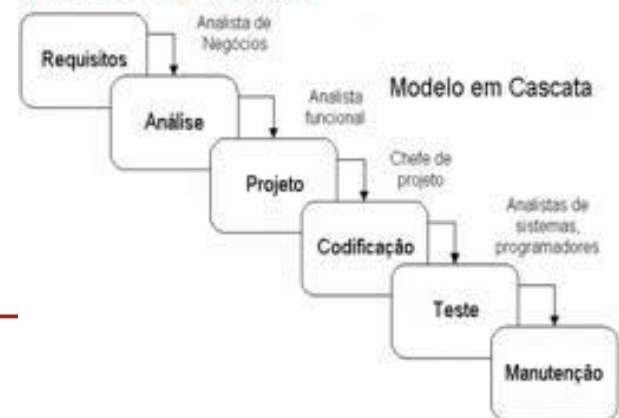
Ciclo de Vida



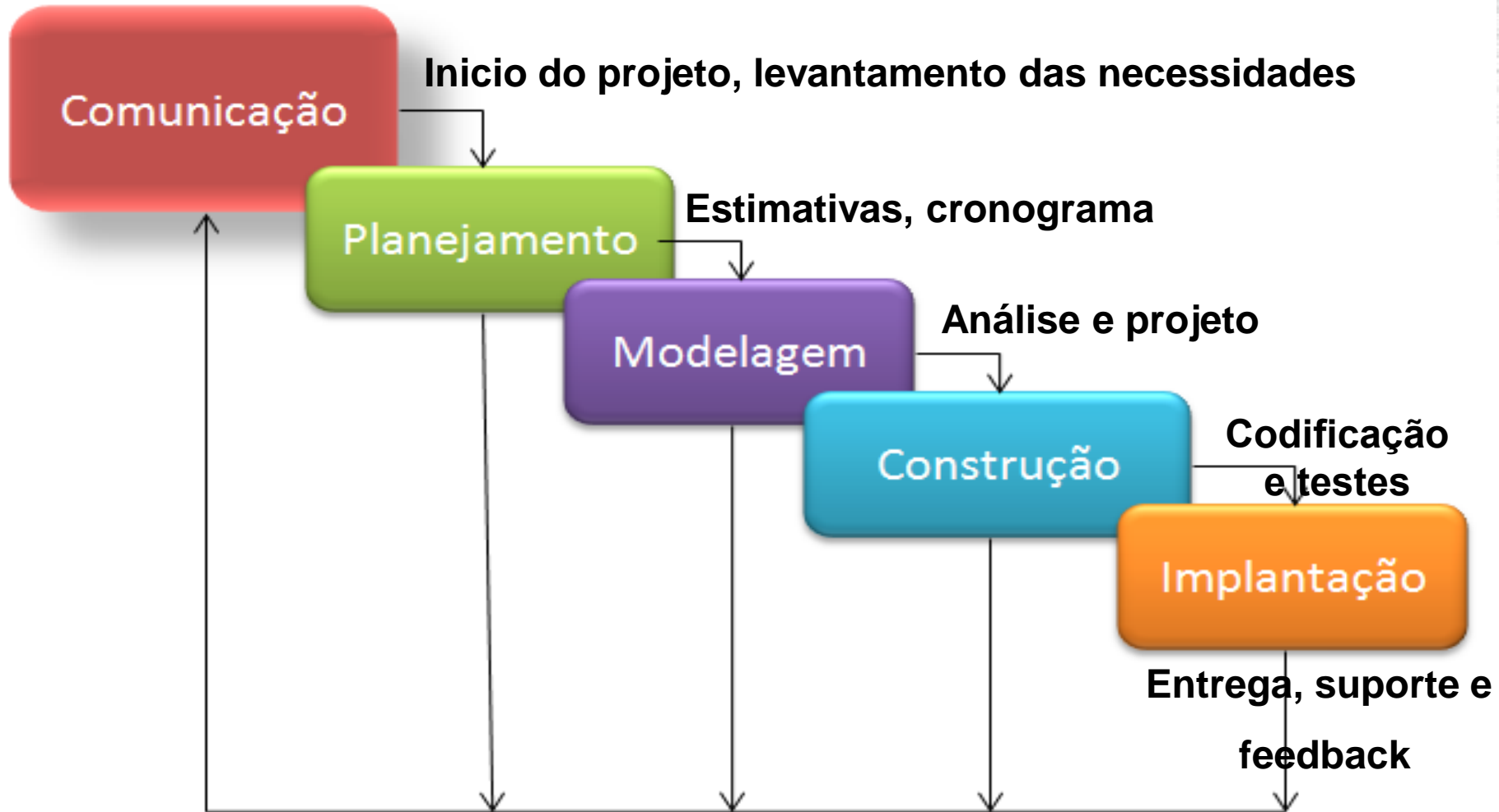
Cascata

- O modelo cascata, algumas vezes chamado de ciclo de vida clássico, sugere uma abordagem sequencial e sistemática para o desenvolvimento de software, começando com levantamento de necessidades por parte do cliente, avançando pelas fases de **planejamento**, **modelagem**, **construção**, **implantação** e culminando no suporte contínuo do software concluído.

Modelo Cascata



Cascata





Falhas – Modelo Cascata

- Frequentemente é difícil para o cliente estabelecer explicitamente todas as necessidades. O modelo cascata requer isso e tem dificuldade para adequar a incerteza natural que existe no início de muitos projetos;
 - O cliente deve ter paciência. Uma versão operacional do(s) software(s) não estará disponível antes de estarmos próximos do final do projeto. Um erro grave, se não detectado até o programa operacional ser revisto, pode ser desastroso.
-



Modelo RAD

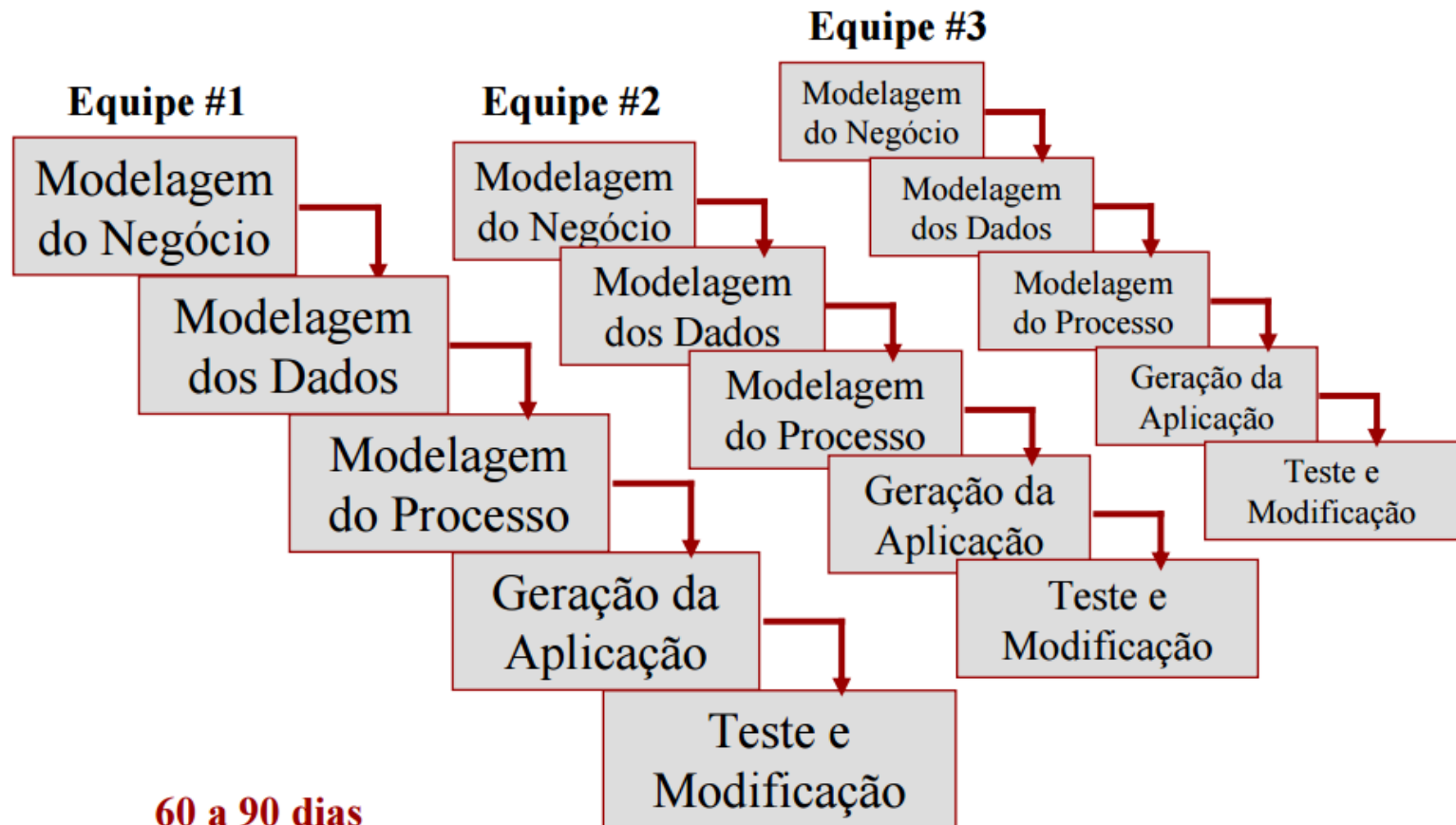
- RAD (Rapid Application Development) é um modelo sequencial linear que enfatiza um ciclo de desenvolvimento extremamente curto;
 - O desenvolvimento rápido é obtido usando uma abordagem de construção baseada em **componentes**;
-



Modelo RAD

- O modelo RAD é usado principalmente para aplicações de sistema de informação;
 - Cada função principal pode ser direcionada para uma equipe RAD separada e então integrada para formar o todo.
-

Modelo RAD





Modelo RAD

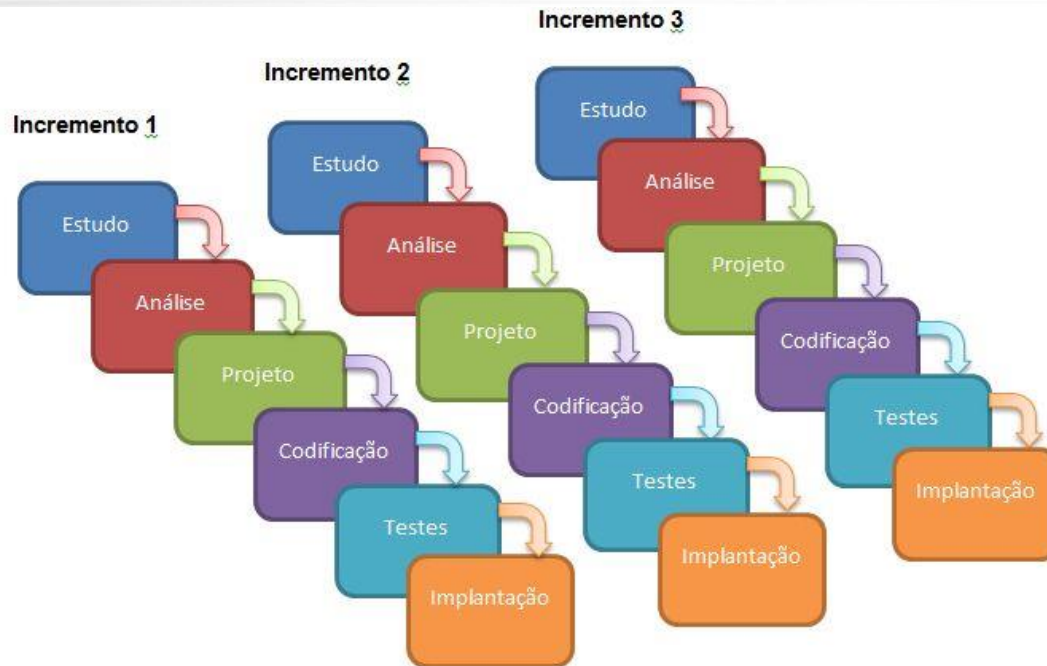
- Desvantagens:
 - Exige recursos humanos suficientes para todas as equipes;
 - Exige que desenvolvedores e clientes estejam comprometidos com as atividades de “fogo-rápido” a fim de terminar o projeto em um prazo curto.
-



Incremental

- O modelo incremental combina elementos os fluxos de processos lineares e paralelos;
 - O modelo incremental aplica sequencias lineares, de forma escalonada, à medida que o tempo vai avançando. Cada sequencia linear gera incrementos (entregáveis/aprovados/liberados) do software;
 - Quando se utiliza um modelo incremental, frequentemente, o primeiro incremento é um produto essencial, isto é, os requisitos básicos são atendidos.
-

Modelo Incremental



Prototipação



Prototipação

- Uma iteração de prototipação é planejada rapidamente e ocorre a modelagem (projeto rápido). Na sua forma ideal, o protótipo atua como um **mecanismo para identificar os requisitos do software**;
- Caso seja necessário desenvolver um protótipo operacional, pode-se utilizar partes de programas existentes ou aplicar ferramentas (por exemplo, geradores de relatórios) que possibilitem gerar rapidamente tais programas operacionais;

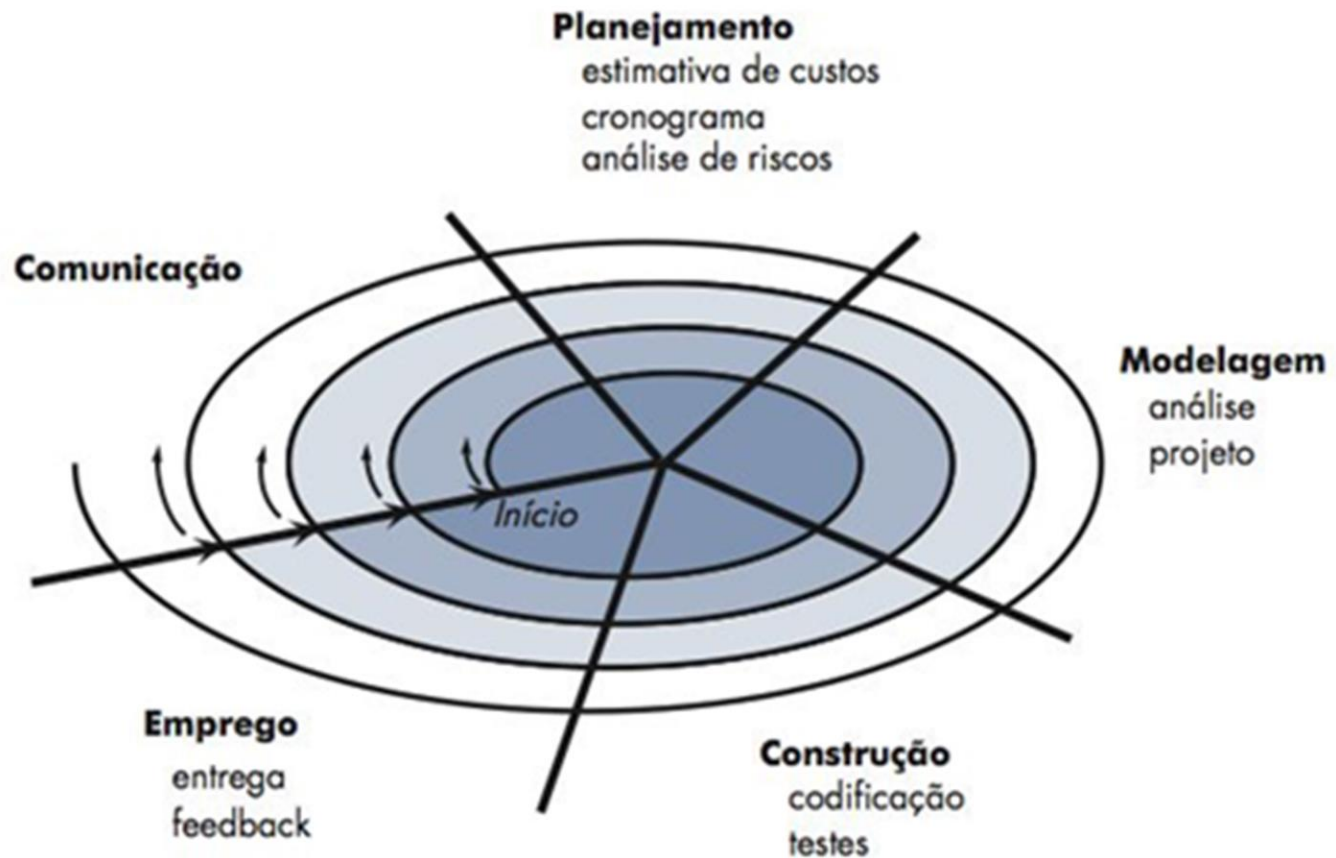
O protótipo pode servir como “o primeiro sistema”. Embora alguns protótipos sejam construídos como “descartáveis”, outros são evolucionários, no sentido de que evoluem lentamente até se transformar no sistema real.



Espiral

- Usando-se o modelo espiral, o software será desenvolvido em uma série de versões evolucionárias. Nas primeiras iterações, a versão pode consistir em um modelo ou em um protótipo. Já nas iterações posteriores, são produzidas versões cada vez mais completas do sistema que passa pelo processo de engenharia;
 - O modelo espiral é uma abordagem realista para o desenvolvimento de sistemas e de software em larga escala. Pelo fato de o software evoluir à medida que o processo avança, o desenvolvedor e o cliente compreendem e reagem melhor aos **riscos** em cada nível evolucionário.
-

Espiral

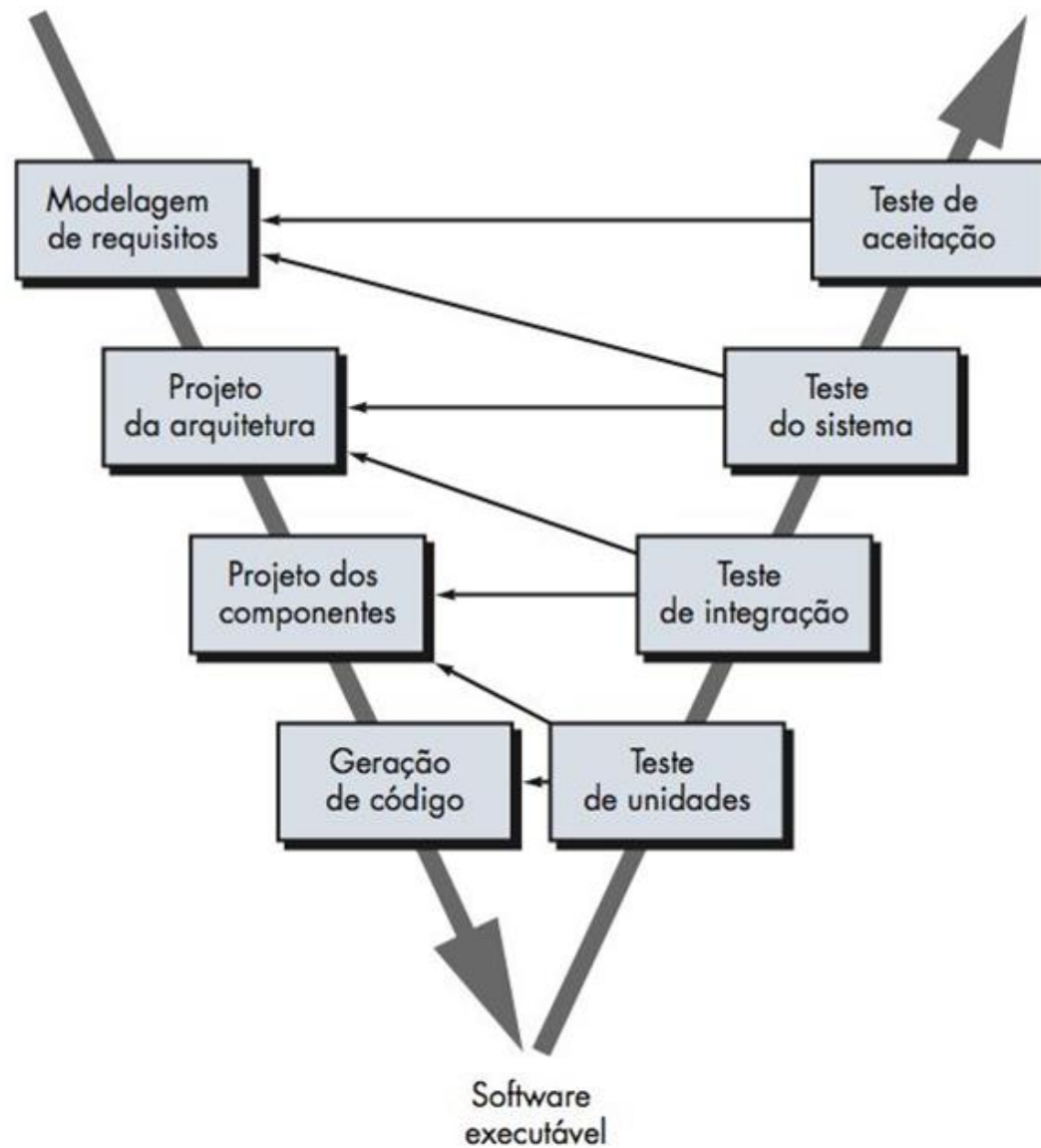




Modelo V

- O modelo V descreve a relação entre ações de garantia da **qualidade** e as ações associadas à comunicação, modelagem e atividades de construção iniciais. À medida que a equipe de software desce em direção ao lado esquerdo do V, os requisitos básicos do problema são refinados em representações progressivamente cada vez mais detalhadas e técnicas do problema e de sua solução
 - O modelo V fornece uma forma para visualizar como a verificação e as ações de validação são aplicadas ao trabalho de engenharia anterior.
-

Modelo V



Vantagens x Desvantagens

Modelo	Vantagens	Desvantagens
Cascata	<ul style="list-style-type: none">• Eficiente quando o domínio da aplicação é bem entendido;• Minimiza tempo de planejamento;• Funciona bem para equipes tecnicamente fracas;• Possui etapas bem definidas.	<ul style="list-style-type: none">• Apenas na última fase é possível entregar um produto;• Difícil e caro para voltar e corrigir erros;• Muitas vezes o cliente não consegue definir todos os requisitos inicialmente;

Vantagens x Desvantagens

Modelo	Vantagens	Desvantagens
Prototipação	<ul style="list-style-type: none">• Os clientes conseguem ver os processos;• Serviços esquecidos podem ser detectados e adicionados no próximo protótipo;• Um sistema com telas está funcionando nos primeiros estágios;• O protótipo pode ser usado como treinamento de usuário;	<ul style="list-style-type: none">• É muito difícil prever a duração da elaboração do sistema;• É extremamente difícil prever quantas iterações serão necessárias para concluir o sistema.

Vantagens x Desvantagens

Modelo	Vantagens	Desvantagens
Incremental	<ul style="list-style-type: none">• Feedback do cliente a cada incremento;• Disponibilidade de partes prontas do sistema mais cedo;• Facilidade nos testes, uma vez que testar cada incremento (módulo) é mais fácil que testar o sistema inteiro.	<ul style="list-style-type: none">• Podemos encontrar dificuldade em integrar cada módulo (incremento);• Nem todo sistema é possível dividir em partes, o que inviabiliza esse modelo.

Vantagens x Desvantagens

Modelo	Vantagens	Desvantagens
Espiral	<ul style="list-style-type: none">• Ajuda a aumentar a qualidade pelo planejamento e análise dos riscos em cada fase;• Cada iteração da espiral pode ser customizada para as necessidades específicas do projeto.	<ul style="list-style-type: none">• É complexo e requer atenção especial e conhecimentos específicos da equipe;• Exige experiência e esforço extra, o que aumenta os custos.



Processo Unificado

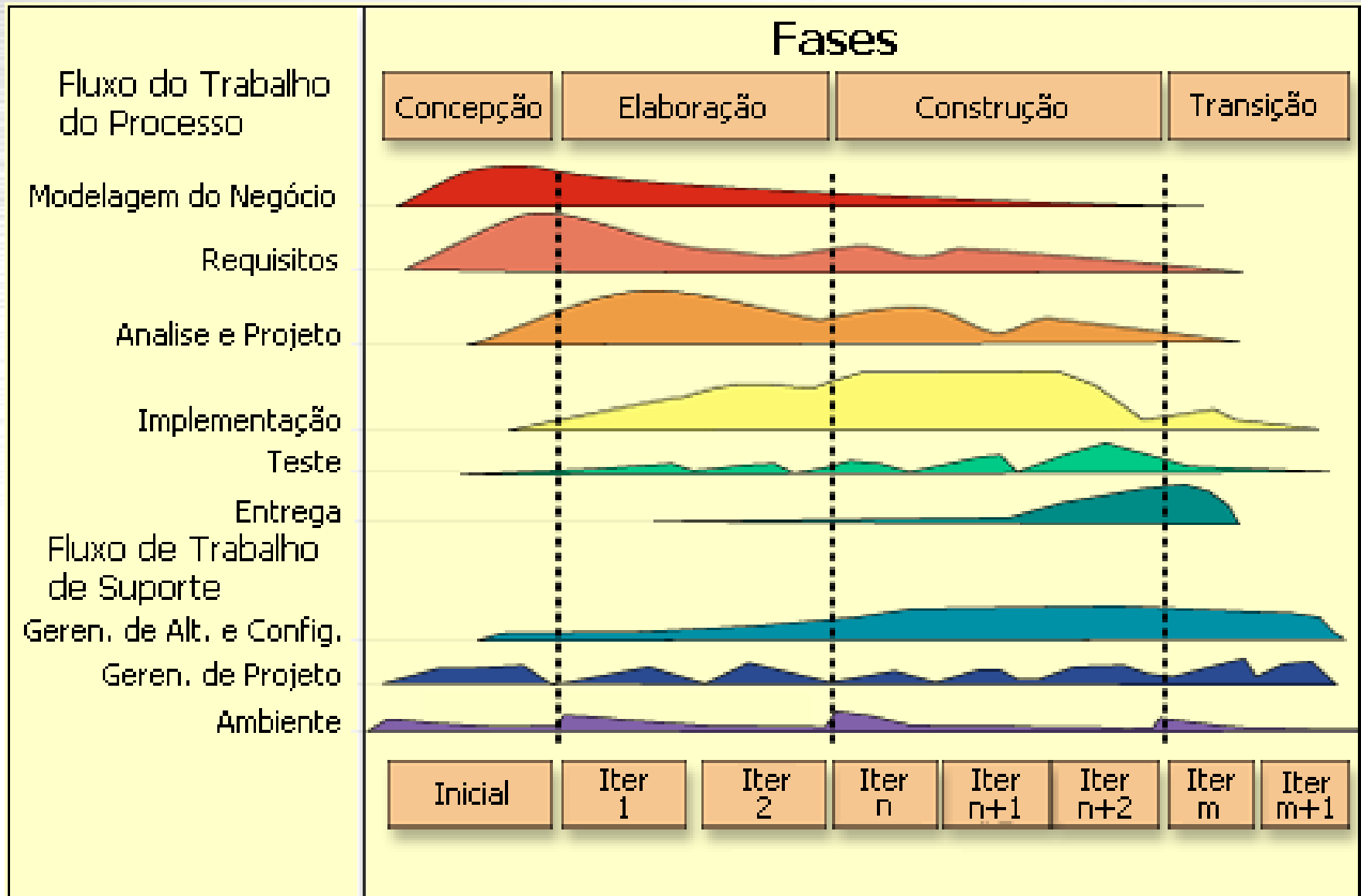
- O processo unificado é uma tentativa de aproveitar os melhores recursos e características dos modelos tradicionais de processo de software, mas caracterizando-os de modo a implementar muitos dos melhores princípios do desenvolvimento ágil de software;
-



Processo Unificado

- O processo unificado possui 4 fases:
 - Concepção;
 - Elaboração;
 - Construção;
 - Transição;
-

RUP





Concepção

- Envolve tanto a atividade de comunicação com o cliente como a de planejamento. Colaborando com os interessados, identificam-se as necessidades de negócio para o software;
 - Desenvolve um planejamento para a natureza iterativa e incremental do projeto decorrente;
 - Nesta fase os requisitos de negócio fundamentais são descritos por meio de um conjunto de casos práticos preliminares descrevendo quais recursos e funções cada categoria principal de usuário deseja.
-

Elaboração

- Envolve atividades de comunicação e modelagem do modelo de processo genérico. A elaboração refina e expande os casos práticos preliminares, desenvolvidos como parte da fase de concepção, e amplia a representação da arquitetura, incluindo cinco visões diferentes do software: modelo de caso prático, modelo de requisitos, modelo de projeto, modelo de implementação e modelo de emprego.
-



Construção

- Tendo como entrada o modelo de arquitetura, a fase de construção desenvolve ou adquire componentes de software; esses componentes farão com que cada caso prático (de uso) se torne operacional para os usuários finais. Os modelos de requisitos e de projeto, iniciados durante a fase de elaboração, são completados para refletir a versão final do incremento de software.
-



Transição

- Envolve os últimos estágios da atividade da construção genérica e a primeira parte da atividade de emprego genérico: entrega e realimentação;
 - Entrega-se o software aos usuários finais para testes beta e o feedback dos usuários relata defeitos e mudanças necessárias.
-