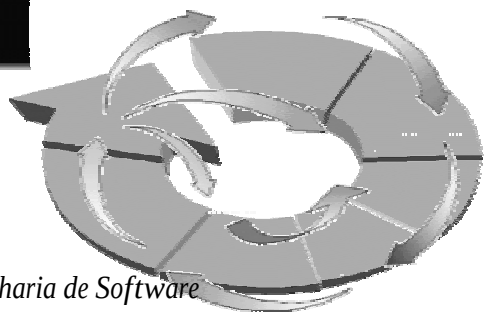
PUC CAMPINAS



Engenharia de Software

Tema da Aula

Modelos de Ciclos de Vida de Software

1 – Modelo em Cascata

Prof. Cristiano R R Portella
portella@widesoft.com.br

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software

O conceito de “Ciclo de Vida de Software” é um paradigma da Eng.Software.

Existem vários modelos de ciclo de vida de software, alguns cobrindo apenas da concepção ao desenvolvimento, enquanto outros cobrem concepção, desenvolvimento, implantação e manutenção.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Modelo didático que divide o ciclo de vida de 5 a 12 fases (cf. o autor).

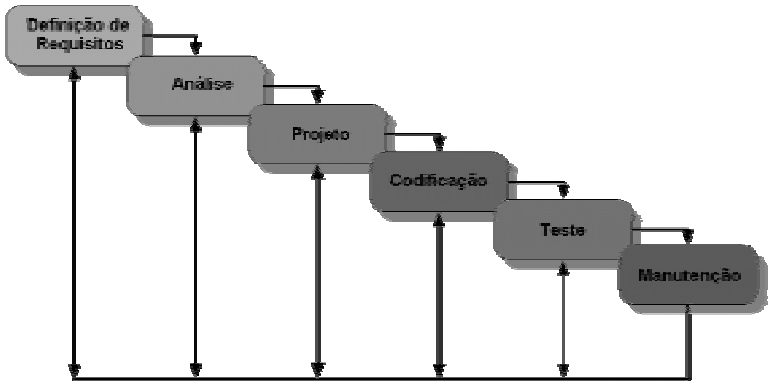
Criado por Winston W. Royce (1970) e aperfeiçoado por Barry Boehm (1976).

Adequação:

Projetos grandes (cobre todas as fases), quando os requisitos estão claramente definidos no início do desenvolvimento, com complexidade baixa e riscos técnicos e de projeto bem entendidos.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)



PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Definição de Requisitos:

Requisito: (adj.) O que se requisitou ou requereu. Condição necessária para obtenção de certo objetivo. Quesito.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Definição de Requisitos:

Foco:

No usuário e no processo (voz do cliente e voz do processo).

Tarefas:

Extrair os requisitos, especificar cada um deles, redigir uma Definição de Requisitos e valida-los junto ao usuário.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Definição de Requisitos:

Através de consulta ao usuário e observação do processo (existem outras possibilidades, por exemplo analisar os documentos em uso no sistema convencional), extrair os serviços e as metas a serem atingidas, bem como as restrições a serem respeitadas.

- Qual a qualidade desejada para o sistema em termos de funcionalidade, desempenho, flexibilidade de uso etc.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Definição de Requisitos:

Especificar quais os requisitos e não como eles deverão ser obtidos (detalhes de implementação). Esse refinamento será feito na fase de análise e/ou projeto.

Diferença entre:

- Informar clientes classificados em ordem alfabética crescente.
- Informar clientes classificados em ordem alfabética crescente (Tabela TC_Cliente, campo TC_Nome. Montar visão para a tabela ordenada).

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Definição de Requisitos:

Praticamente todo requisito identificado precisa de uma especificação.

Especificar: (v.t.d.) Indicar espécie de; explicar detalhadamente; Descrição rigorosa e minuciosa das características que um material, uma obra ou um serviço deverão apresentar.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Definição de Requisitos:

Por exemplo:

Requisito:

O sistema deve aceitar multi-empresas.

Especificação:

Cada empresa tem CNPJ, endereço e Diretoria própria. Os clientes são cadastrados para a Holding e não por empresa. Atualmente existem 4 empresas, mas esse número poderá chegar a 10 unidades.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Definição de Requisitos:

Gerar um Documento de Especificação, redigido em linguagem inteligível para o usuário.

Finalidade:

1. O usuário deverá analisar e confirmar se a descrição está correta e se atende suas necessidades e expectativa.
2. Será usados pelos desenvolvedores, durante o processo de construção do produto.
3. Quando o produto for entregue, será usado pelo usuário para valida-lo (ver se está conforme os requisitos).

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Definição de Requisitos:

Características do Documento de Especificação:

1. Linguagem de domínio do usuário;
2. Preciso;
3. Completo;
4. Consistente;
5. Sem redundâncias;
6. Sem ambigüidades.

Por exemplo:

O sistema deve ser preciso. (qual o significado de precisão nesse contexto ?)

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Definição de Requisitos: (Doc.de Especificação):

- Requisitos Funcionais:
  O que o produtos de software deve fazer (funcionalidade).
- Requisitos Não Funcionais:
  1. Confiabilidade — Disponibilidade, Integridade, Segurança

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Definição de Requisitos: (Doc. de Especificação):

- Requisitos Não Funcionais (cont.):
  2. Acurácia dos resultados (exatidão).
  3. Desempenho
  4. Problemas da interface homem-máquina
  5. Restrições físicas e operacionais
  6. Portabilidade
  7. Etc.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Definição de Requisitos:

Aplicar os "5 Princípios da Engenharia de Software", especialmente: Abstração, Decomposição e Generalização.

Abstração: Fixar-se nos aspectos importantes, ignorando os detalhes.

Decomposição: Dividir em partes para lidar com a complexidade.

Generalização: Buscar características comuns relegando as características específicas (é uma forma de abstração).

Não usamos: Formalidade e Flexibilização.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Análise de Requisitos:

Foco:

Nos objetivos, restrições e alternativas.

Ferramentas:

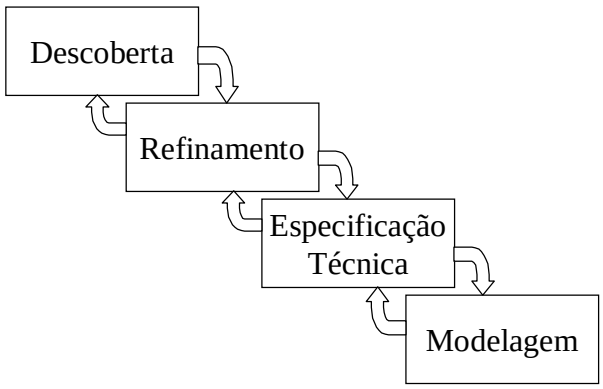
Metodologias/técnicas de modelagem e análise, ferramentas (editores gráficos, CASE's etc).

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Análise de Requisitos:

Obter uma compreensão completa dos requisitos de software, através de:



PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Análise de Requisitos:

Definição detalhada do domínio das informações e do domínio da funcionalidade requerida para o software.

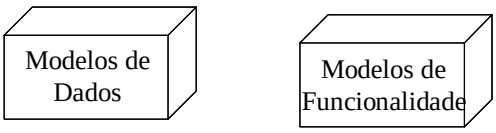
O desenvolvedor deve definir as estruturas de dados, conhecendo cada uma delas (tipo, tamanho, volume, consistências, inter-relação, se disponível em bases de dados, forma de coleta etc).

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Análise de Requisitos:

Também deve definir detalhes da funcionalidade (detalhes de como o sistema deve se comportar, tanto em funcionalidade como em performance, segurança etc.)



PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Projeto de Software:

Foco:

Nos dados, componentes de software e no produto final de software.

Ferramentas:

Metodologias/técnicas de modelagem e análise, ferramentas (editores gráficos, ferramentas CASE etc).

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Projeto de Software:

Representação das funções do sistema, em uma forma que possa ser transformada em programas executáveis.

Decompor o produto de software desejado em partes (programas, módulos, componentes etc).

Recompor, pensando no produto final (monolítico)

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Projeto de Software:

Criar documento de especificação de cada parte do produto:

- O que ela deve fazer (entradas, comportamento, saídas).
- Definir a relação entre componentes

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Projeto de Software:

Relação entre componentes:

- Forte coesão (interna)
- Fraco acoplamento (externo)

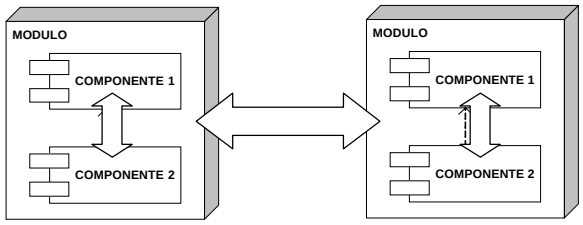


DIAGRAMA DE COMPONENTE

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Projeto de Software:

Projeto Preliminar:

Transformação dos requisitos numa arquitetura de dados e de funcionalidade.

Projeto Detalhado:

Refinamento das representações estruturais, obtendo-se assim representações detalhada dos algoritmos (nos casos em que for necessário) e dos dados.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

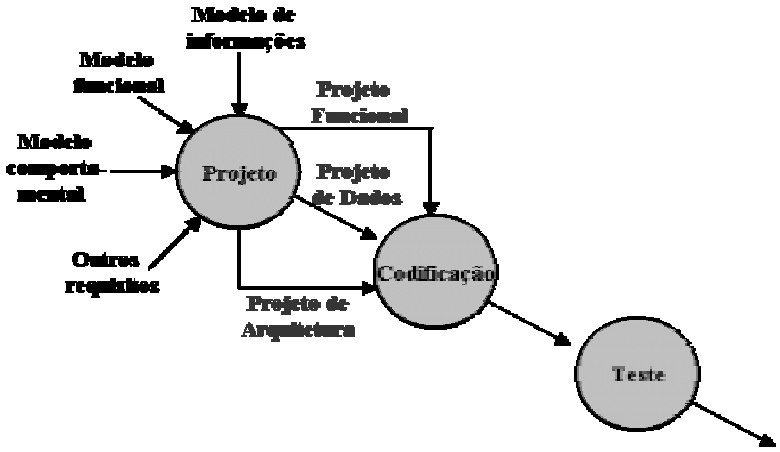
Projeto de Software:

Detalhe: O ciclo de vida clássico tem um foco acentuado no produto de software e não o processo, portanto está faltando um projeto operacional, visando detalhes do processo de construção (esforços, responsabilidades, milestones etc).

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Projeto de Software:



PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Codificação:

Foco:

Nos algoritmos e nas linguagens de programação (sintaxe, limitações, funcionalidade disponível,

Ferramentas:

Linguagens de programação, geradores de código fonte, CASE de amplo espectro etc.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Teste:

Foco:

Nas especificações e nas saídas do produto de software.

Ferramentas:

Técnicas de testagem, procedimentos da Qualidade, ferramentas de testagem.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Teste:

- Testar contra especificações de requisitos
- Testar contra padrões da instalação
  - Nomenclatura de campos, tabelas
  - Padrões de interfaces
  - Padrões de qualidade

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Teste:

- Unitário (cada unidade de software)
- De Integração do sistema (todos os componentes, a partir de uma estratégia de aglutinação progressiva).
- De integração entre sistemas (sistema gerado com outros sistemas com os quais haverá troca de informações).

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Teste

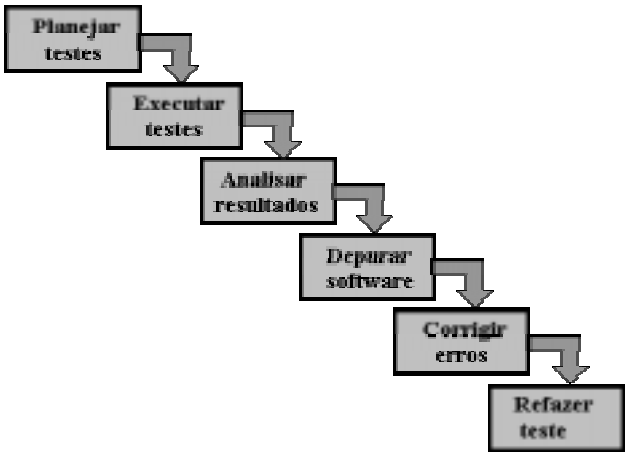
Elementos necessários:

- Padrões da instalação e Def. Requisitos
- Ferramentas de teste
- Plano de teste
- Critérios de teste
- Critérios de completude (quando parar ?)
- Gerenciamento dos casos de teste

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Teste Passos para “um caso de teste”:



PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Teste:

Recomendação IEEE (Computer Dictionary). Não usar o termo “bug”; usar:

- Defeito (Fault)
  Instrução ou definição incorreta.
- Falha (Failure)
  Resultados incorretos
- Erro (Mistake)
  Falha resultante de ação humana

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Teste:

Testes especiais:

- Performance
- Segurança
- Stress etc

Teste na área de produção:

- Teste inicial (alfa)
- Beta teste

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

(Operação e) Manutenção:

Foco:

Depende do tipo de manutenção (algoritmo, usuário, processo, tecnologia etc)

Ferramentas:

Linguagens de programação, ferramentas CASE etc . A rigor deveria ser o mesmo foco e as mesmas ferramentas da fase de Definição de Requisitos (volta às origens).

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

(Operação e) Manutenção:

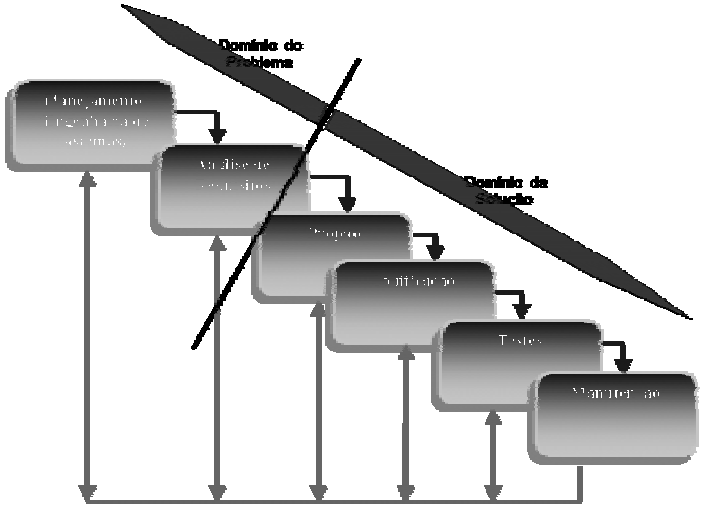
Fase mais longa do ciclo.

Tipos de manutenção:

- Corrigir erros remanescentes
- Adaptar a novas situações e necessidades
- Preparar para futuras alterações

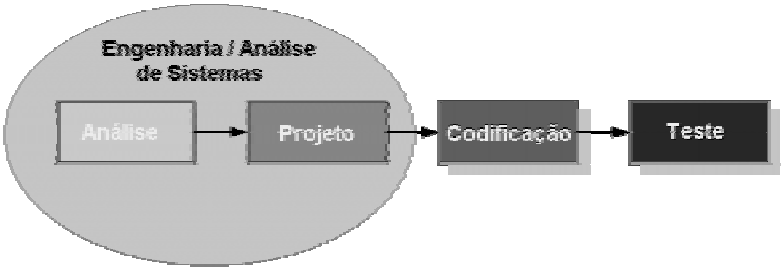
PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico Atualizado



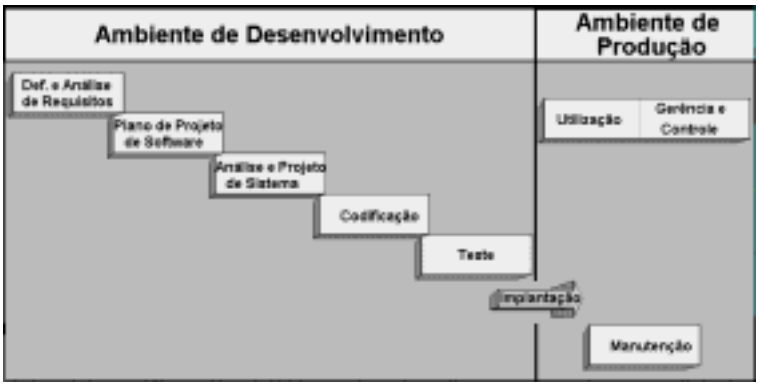
PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (visão realista)



PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (ambientes)



PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Implantação:

Transferir da área de desenvolvimento para a área de produção.

- Preparar área de rede (grupo, usuários, diretórios, direitos)
- Conversão de arquivos ou
- Geração dos arquivos (caso Biblioteca)
- Treinamento dos usuários
- Integrar tecnologias

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Falhas do modelo:

Todo modelo tem suas limitações (abstração, estático, foco (visão) do modelo, etc.

1) Imagina o processo como sendo seqüencial e progressivo, onde cada fase é estanque, mesmo com as setas indicando retorno a fases anteriores, cada fase é vista isoladamente. Tenta manter a linearidade para manter o processo previsível e de fácil gerenciamento.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Falhas do modelo:

2) A fase de Análise só se inicia após obtenção dos requisitos, que devem ser:

- Completos
- Corretos
- Não ambíguos
- Não redundantes
- Sem detalhes de implementação

Quando isso é possível ?

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Falhas do modelo:

3) Não estimula o desenvolvimento conjunto. O trabalho com o usuário está restrito à fase de Definição de Requisitos.

4) A entrega do produto só ocorre depois de terminado. Quando o usuário tem a chance de ver se o produto atende suas necessidades e expectativa, ele já está pronto.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Falhas do modelo:

5) Não prevê um "Estudo de Viabilidade", que deveria iniciar ao término da Definição de Requisitos e, no máximo prolongar-se até o início da Análise.

Sua finalidade é evitar que recursos sejam gastos na tentativa de solucionar o problema de maneira errada, além de verificar se a solução é viável do ponto de vista econômico (investimos poucos recursos para termos certeza de que o projeto é viável, evitando perder muitos recursos).

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

O “Estudo de Viabilidade” dever avaliar:

- Se o problema é passível de solução via software
- As possíveis soluções (desenvolver, comprar)
- Estudo de custos x benefícios

Deve conter:

- Definição do problema
- Possíveis soluções (inclusive alternativas)
- Benefício de cada uma delas
- Recursos necessários, custo e prazo de cada uma delas

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Falhas do modelo:

O “Estudo de Viabilidade” dever avaliar:

- Se o problema é passível de solução via software
- As possíveis soluções (desenvolver, comprar)
- Estudo de custos x benefícios

No Estudo de Viabilidades investimos poucos recursos para termos certeza de que o projeto é viável, evitando perder muitos recursos.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

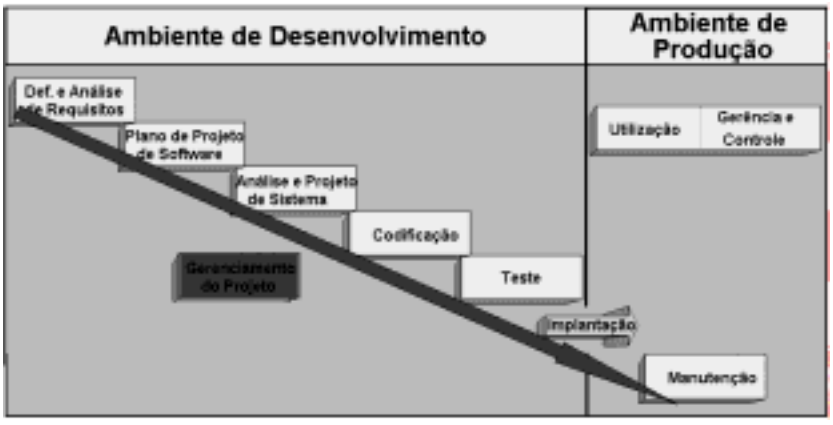
Falhas do modelo:

6) Não menciona o Gerenciamento do Processo de Software, que ocorre simultaneamente a construção do produto e é altamente influenciado pela complexidade deste último.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Falhas do modelo:



PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Gerenciar o processo:

Objetivo: Controlar o processo.

1. Adaptar o modelo de gerenciamento do processo ao tipo de ciclo de vida e tipo de produto.
2. Definir Políticas (autorização de acessos, períodos de backup, responsabilidades, documentação obrigatória etc).
3. Obter recursos
4. Gerenciar recursos
5. Corrigir desvios do projeto e monitorar prazos e custos.

PUC CAMPINAS
Engenharia de Software

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Falhas do modelo:

7) Não permite Engenharia Reversa (reconstrução de sistema legados).

Ciclos de Vida de Software
Modelo Clássico (Cascata)

Na prática:

- Pressões sobre o tempo
- Receber ordens para fazer (mandatório)
- DSI fazendo “política de boa vizinhança”
- Diretoria fazendo “política de boa vizinhança”
- O desenvolvimento já estava previsto em P.D.I.