

Evolução de Software

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

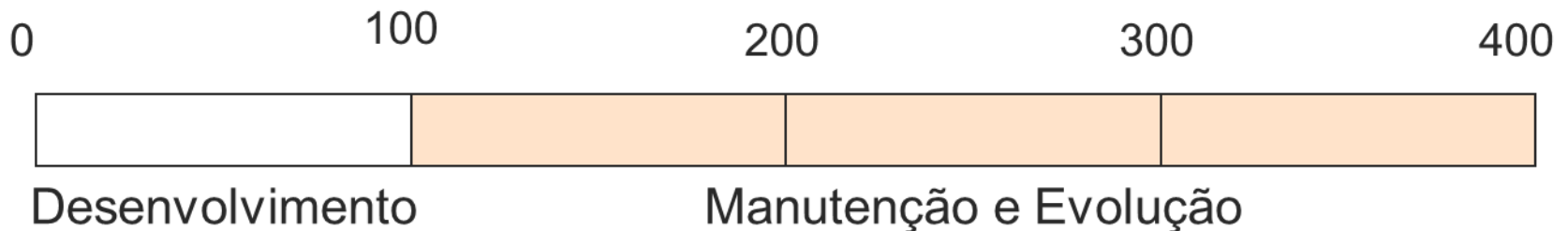
[Evolução de Software]

- Sistemas de software são caros e geralmente têm vida útil longa
 - Empresas precisam usá-los por anos para ter retorno do investimento
- Depois de implantados, sistemas devem inevitavelmente mudar para permanecerem úteis



Custo de Evolução

- O custo de manutenção é muito maior que o custo de desenvolvimento
 - 60% a 90% dos custos do software são relativos à manutenção
 - 75% dos profissionais de software estão envolvidos com atividades de evolução

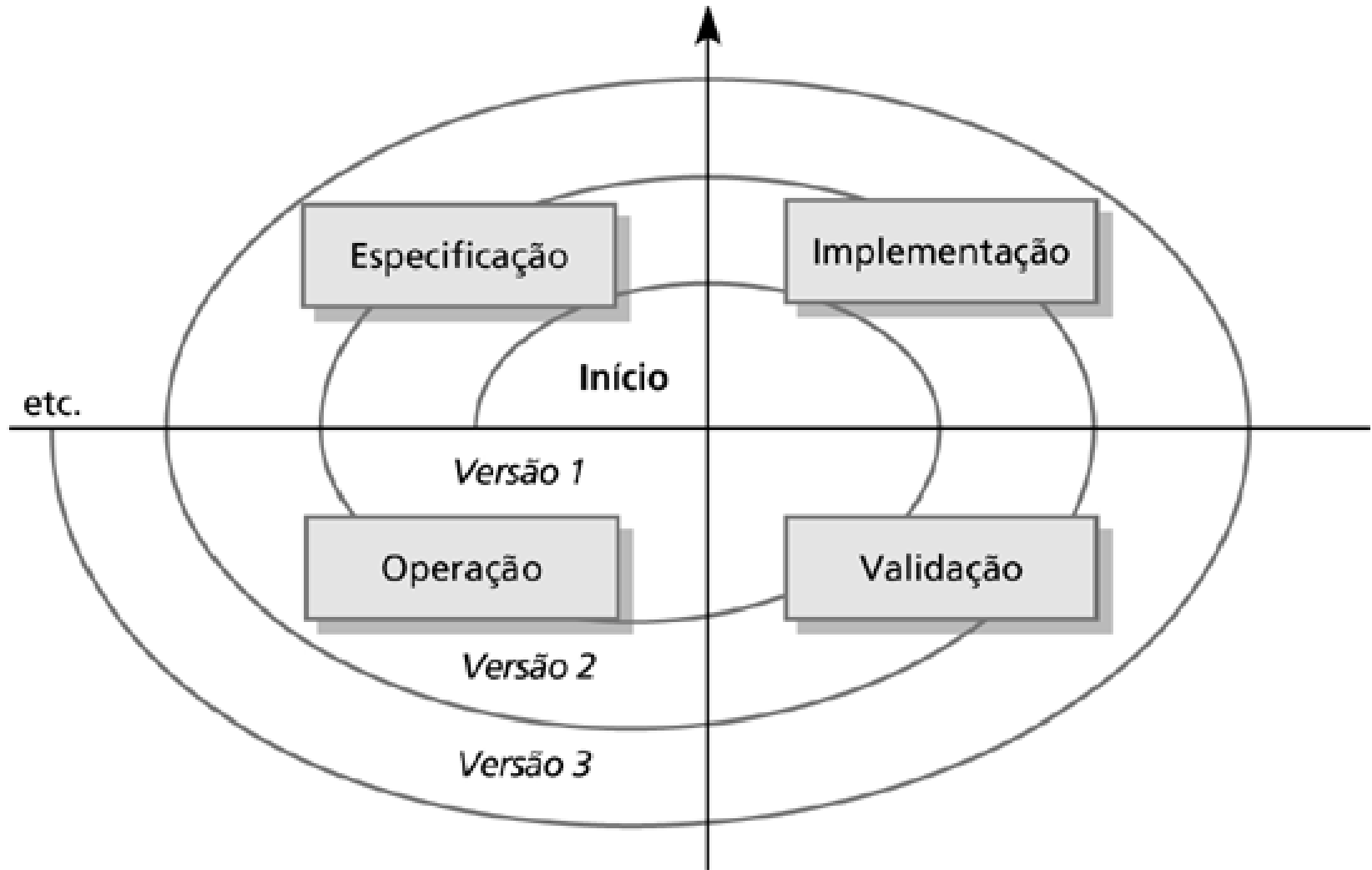


[Mudança Contínua]

- Mudanças no ambiente requerem atualizações do sistema
 - Ao implantar um sistema modificado, este sistema causa uma mudança no ambiente
- Ciclos contínuos e ininterruptos de desenvolvimento e evolução podem ser representados por uma espiral



[Modelo Espiral de Evolução]



Fases do Sistema

- Desenvolvimento de Software
 - Primeira solicitação do cliente
- Evolução
 - Sistema é intensamente usado
- Em Serviço
 - Sistema é pouco usado
- Desativação
 - Empresa considera a substituição do sistema



[Evolução vs. Em Serviço]

Desenvolvimento

Evolução

Em Serviço

Desativação

■ Evolução

- Mudanças significativas são feitas tanto na arquitetura quanto nas funcionalidades
- A estrutura tende a gradativamente se degradar

■ Em Serviço

- Apenas mudanças pequenas e essenciais ocorrem (software é pouco usado)



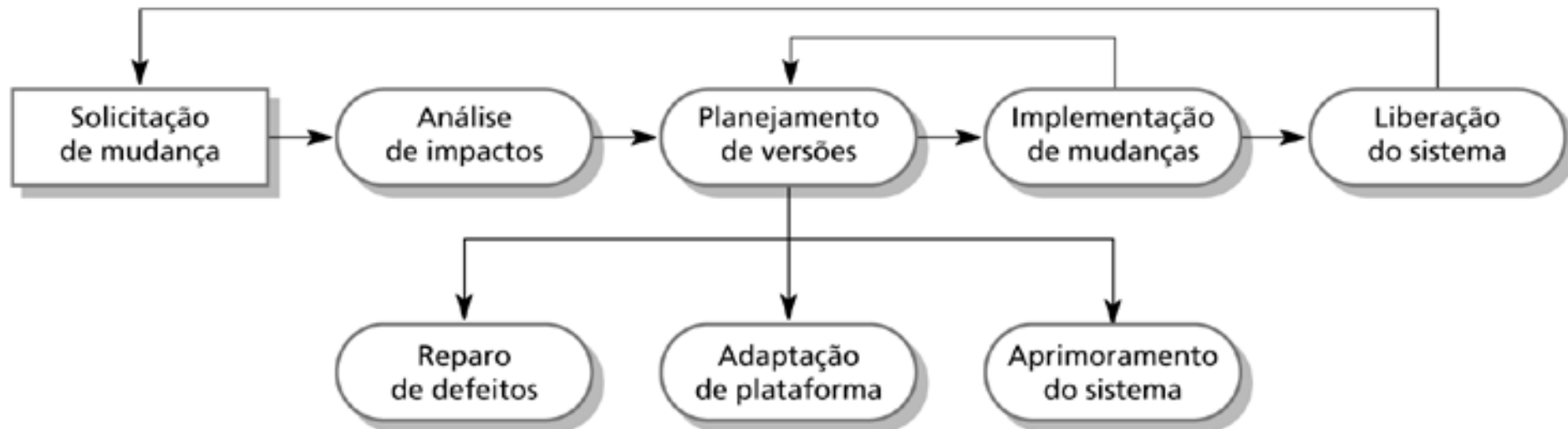
Processos de Evolução

[Processo de Evolução]

- A evolução pode variar entre organizações e tipos de sistemas
 - **Processo informal:** solicitação de mudanças provém de conversas entre usuários e desenvolvedores
 - **Processo formalizado:** documentação produzida e avaliada em cada atividade do processo

Modelo de Processo

- Cada solicitação de mudança é avaliada, planejada e implementada
 - O processo se repete em novas solicitações



[Análise e Planejamento]

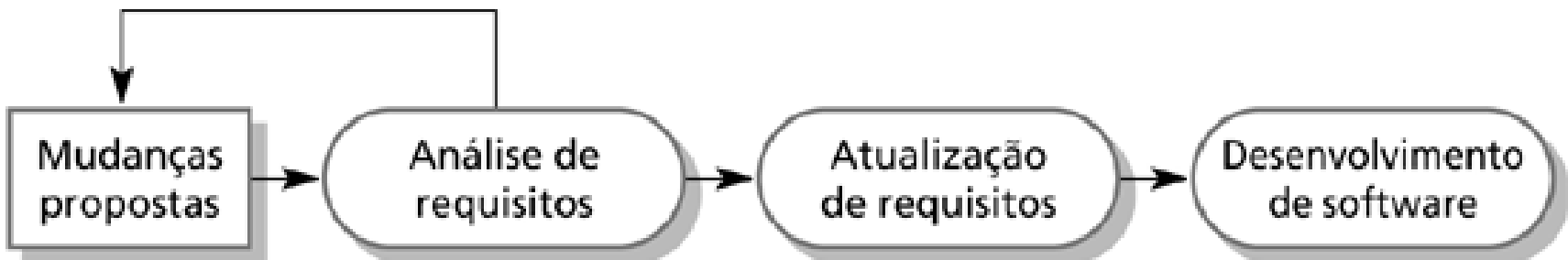
- Análise de Impactos
 - Avalia o custo e a porção do sistema afetado pela mudança solicitada
 - Decide se a mudança será aceita
- Planejamento de Versões
 - Caso sejam aceitas, diferentes tipos de mudanças podem ser agrupados para a próxima versão
 - Correção de defeitos, adaptações de plataforma e aperfeiçoamentos

[Implementação e Liberação]

- Implementação de Mudanças
 - Geralmente, uma nova versão inclui um conjunto com várias mudanças solicitadas
- Liberação do Sistema
 - Uma nova versão é liberada após validação com o cliente
- O processo se repete com um novo conjunto de solicitação de mudanças

Implementação de Mudanças

- Lembrando que a implementação de mudanças deve atualizar a documentação correspondente
 - Especificação, modelos de projeto, implementação, testes, etc.



[Compreensão de Programa]

- A implementação da mudança envolve atividades semelhante ao desenvolvimento inicial do sistema
 - Uma diferença fundamental é ser necessário compreender o programa
- Compreensão inclui
 - Entender como a funcionalidade a ser alterada encontra-se implementada
 - Avaliar o impacto da mudança em outras partes do programa



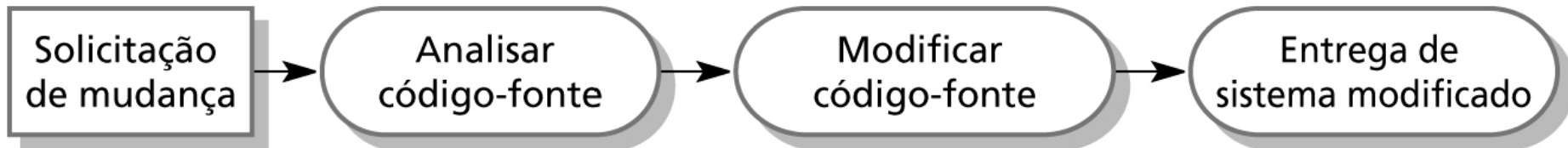
Situações Emergenciais

- Algumas solicitações de mudanças podem ser urgente, exemplo:
 - Se ocorrer um defeito grave que impede o funcionamento normal do sistema
 - Alterações do ambiente impedem o funcionamento do sistema
 - Mudanças no negócio do cliente ou uma nova legislação
 - Entrada de um novo concorrente no mercado



[Reparo Emergencial]

- Pode não haver tempo hábil para atualização da documentação
 - Ao invés de seguir as atividades “normais” de desenvolvimento, a alteração é feita diretamente no código



[Refazer a Correção]

- Código para correção de emergência geralmente tem baixa qualidade
- Portanto, a correção de emergência deveria ser idealmente re-implementada
 - E a documentação correspondente atualizada
- Na prática, este re-trabalho tem baixa prioridade e acaba esquecido

[Bibliografia da Aula]

- Ian Sommerville. **Engenharia de Software**, 10ª Edição. Pearson Education, 2019.
 - Capítulo 9 Evolução de Software (até Seção 9.1)