



**UNIVERSIDADE PRESBITERIANA MACKENZIE**  
**Engenharia de Software 3ºH**



# **Modelo Sequencial**

## **Ciclo de Vida Clássico**

Alexandre Acea Tessarotto – 311.3337-1  
Bruno Braz Alves Berchielli – 309.75591  
Guilherme Pires Berrettini – 310.8217-3  
Gustavo M. Abussamra - 31170927  
Willer G. Santana – 310.14-7

# Introdução

Um modelo de ciclo de vida ou modelo de processo pode ser visto como uma representação abstrata de um esqueleto de processo, incluindo tipicamente algumas atividades principais, a ordem de precedência entre elas e, opcionalmente, artefatos requeridos e produzidos. De maneira geral, um modelo de processo descreve uma filosofia de organização de atividades, estruturando as atividades do processo em fases e definindo como essas fases estão relacionadas. Entretanto, ele não descreve um curso de ações preciso, recursos, procedimentos e restrições. Ou seja, ele é um importante ponto de partida para definir como o projeto deve ser conduzido, mas a sua adoção não é o suficiente para guiar e controlar um projeto de software na prática. Ainda que os processos tenham de ser definidos caso a caso, de maneira geral, o ciclo de vida de um software envolve, pelo menos, as seguintes fases:

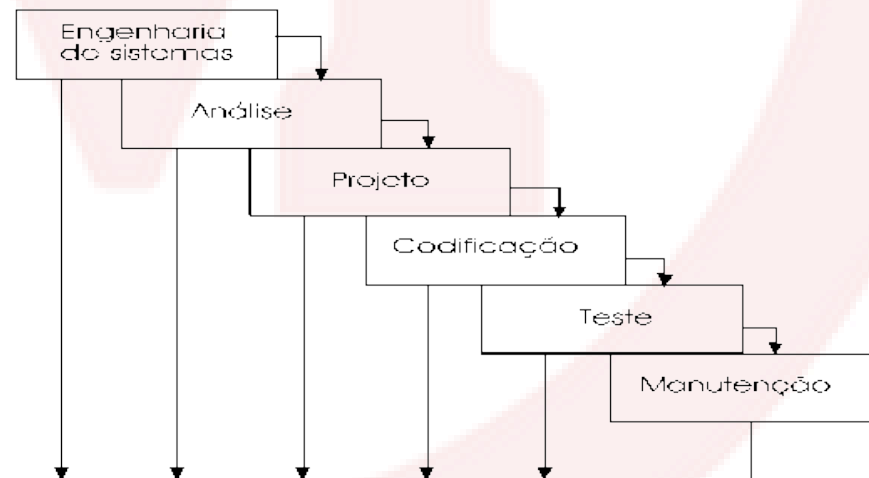


Figura 1 – O ciclo de vida clássico

# Engenharia de Sistemas

Uma vez que o software sempre faz parte de um sistema mais amplo, o trabalho inicia-se com o estabelecimento dos requisitos para todos os elementos do sistema. Essa visão é essencial quando o software deve fazer interface com outros elementos, tais como hardware, pessoas e banco de dados

# Análise de requisitos de Software

O processo de coleta dos requisitos é intensificado e concentrado especificamente no software. O Engenheiro (“Analista”) de software deve compreender o domínio (escopo) da informação. Os requisitos são documentados e revistos com o cliente.

# Projeto

O projeto de software é, de fato, um processo de múltiplos passos que se concentra em quatro atributos distintos: Estrutura de dados, Arquitetura de Software, Detalhes Procedimentais e Caracterização de Interface. Como os requisitos, o projeto é documentado e torna-se parte da configuração do software.

# Codificação

A etapa de codificação executa a tarefa de traduzir o projeto em uma forma legível por máquina. Se o projeto estiver bem detalhado a codificação pode ser executada mecanicamente.

# Testes

Tão logo finalizada a fase de codificação inicia-se os testes. O processo de testes concentra-se nos aspectos lógicos internos do software, garantindo que todas as rotinas tenham sido testadas. Concentra-se também nos aspectos funcionais externos, verificando se as entradas externas produzem resultados reais que reflitam o exigido.

# Manutenção

Indubitavelmente, o software sofrerá mudanças depois que for entregue ao cliente. Mudanças essas por erros que foram encontrados ou porque o cliente exigiu acréscimos funcionais ou de desempenho. A manutenção reaplica cada uma das etapas precedentes do ciclo de vida clássico, e não a um novo.



# Principais Problemas

- **1.** Os projetos reais raramente seguem o fluxo sequencial que o modelo propõe.
- **2.** Muitas vezes é difícil para o cliente declarar todas as exigências explicitamente logo no início.
- **3.** Exige paciência do cliente. Uma versão do software não estará disponível até um ponto tardio do cronograma do projeto. Um erro grosseiro se não detectado pode ser desastroso.