

# **Introdução a Engenharia de Software / Análise de Sistemas**

Disciplina: Análise e Projeto de Sistemas I  
Curso: Computação – Sistemas de Informação  
Prof. Marcelo Balbino

Para explicar o porquê da necessidade e surgimento da Engenharia de Software, gosto sempre de fazer um paralelo com a Engenharia Civil. Não sou um conhecedor da estória da Engenharia Civil, mas certamente as primeiras construções por sua simplicidade foram feitas sem uma planta detalhada e ainda sim alcançavam razoavelmente seus objetivos. No entanto, com o passar do tempo as construções se tornaram cada vez maiores e mais complexas, chegando hoje ao ponto de vermos prédio com dezenas de andares, viadutos, pontes incrivelmente extensas e etc. Para tanto foi necessário o surgimento e evolução da Engenharia Civil para desenvolvimento de técnicas, métodos, estruturas, enfim recursos para possibilitar o desenvolvimento seguro das edificações que vemos hoje.

Situação semelhante viveu a construção dos softwares. Durante muito tempo os sistemas foram pequenos, simples e usados para resolver problemas muito específicos. No entanto, com a evolução do hardware e o aumento da complexidade dos negócios, aumentou junto a complexidade dos sistemas. Daí a necessidade da Engenharia de Software para estudo e desenvolvimento de técnicas, metodologias, padrões e outros para nos municiar de recursos que nos permita desenvolver sistemas grandes e complexos com qualidade.

Historicamente a Engenharia de Software surgiu no final da década de 60 em resposta a um fato importante conhecido como a “Crise do Software”. Nesta época, os sistemas começaram a tomar uma dimensão cada vez maior e seu desenvolvimento ficava cada vez mais complexo. Entretanto, as técnicas usadas no seu desenvolvimento continuavam a ser as mesmas usadas anteriormente para desenvolver pequenos programas. Assim esta combinação, sistemas complexos e metodologias ultrapassadas culminaram com esta crise. Estávamos tentando construir prédios de 20 andares sem técnicas e planejamento, as conseqüências não são difíceis de prever: os software eram entregues com atrasos, que as vezes chegava na casa dos anos, conseqüentemente os custos também extrapolavam. Além disso, os produtos frequentemente saíam com baixa qualidade, gerando altos custos de manutenção e deixando os clientes extremamente insatisfeitos.

Em resposta a crise surgiu a Engenharia de Software que vem evoluindo até os dias de hoje para cada vez nos proporcionar mais recursos para o desenvolvimento de sistemas, mesmo que complexos, com qualidade.

## **O que é software?**

Muita gente associa a palavra software apenas ao programa em si. No entanto, o software é não só o programa mais toda a documentação associada (documentação do sistema e documentação do usuário) e os arquivos de configuração necessários para o total funcionamento do sistema.

Há dois tipos de software:

- Produtos genéricos: são sistemas do tipo *stand-alone*, também conhecidos como software de prateleira ou pacotes de software. As organizações os desenvolvem na tentativa de se vender em larga escala. Para isto estes precisam ser genéricos o suficiente para atender a todos os clientes.
- Produtos sob encomenda (específicos / personalizados): são sistemas desenvolvidos para um cliente em particular seguindo suas especificações.

Evidentemente o preço unitário de venda do sistema genérico é muito mais baixo do que do software personalizado, pois a empresa espera ter o retorno financeiro vendendo uma grande quantidade de cópias. No que diz respeito ao software sob encomenda a empresa desenvolvedora precisa ter muito cuidado no estabelecimento das regras para alteração do software após sua conclusão. Caso estas regras não sejam claramente estabelecidas este certamente será um ponto de atrito permanente entre o cliente e a empresa desenvolvedora.

## **O que é a Engenharia de Software?**

Para entendermos o que é a Engenharia de Software é interessante antes ter claro o que é a Engenharia de uma forma geral. A Engenharia é a arte de aplicar conhecimentos científicos e empíricos e certas habilitações específicas à criação de estruturas, dispositivos e processos para converter recursos naturais em formas adequadas ao atendimento das necessidades

humanas. É particularmente importante nesta definição ressaltar o ponto onde se coloca que a Engenharia aplica conhecimentos científicos e empíricos uma vez que a Engenharia de Software usa muitas vezes de conhecimentos empíricos para desenvolvimento de seus recursos, ou seja, boa parte dos métodos desenvolvidos e aplicados pela Engenharia de Software não são cientificamente comprovados, mas vem sendo usados com sucesso ao longo dos anos e por isso são sugeridos pela mesma.

Assim, a Engenharia de Software propriamente dita é ramo da engenharia que busca desenvolver metodologias, processos, ferramentas, padrões, técnicas, enfim recursos para auxiliar o desenvolvimento de softwares. A Engenharia de Software se ocupa de todos os aspectos que cercam a produção de software, desde a entrevista com o cliente para levantamento dos requisitos do sistema, até as atividades de manutenção depois do sistema pronto e entregue. Vale frisar que a Engenharia de Software se dedica não só aos aspectos que envolvem a produção do software, mas também as atividades de gerenciamento dos projetos e construção de ferramentas, teorias e métodos, ou seja, tudo o mais que possa dar apoio ao desenvolvimento deste software.

O intuito principal da Engenharia de Software é que os sistemas sejam desenvolvidos verdadeiramente como uma atividade de Engenharia, ou seja, com disciplina, seguindo métodos testados e definidos. No entanto, o que ainda se vê nos dias de hoje são que muitos desenvolvedores produzem sistemas como se fosse uma atividade artística seguindo seus próprios métodos, criatividade e intuição. Trabalhando assim a probabilidade de se desenvolver um sistema com erros, de difícil utilização por parte do usuário e difícil manutenção para os desenvolvedores é muito grande, principalmente em se tratando de um sistema complexo.

## **Processos**

Um processo é um conjunto de passos parcialmente ordenados, constituído por atividades, métodos, práticas e transformações usadas para atingir uma meta (documentos, modelos, produtos, etc). Todo processo deve detalhar:

- O que deve ser feito
- Quando

- Por quem
- O que usa
- O que deve ser produzido

Um dos intuitos da Engenharia de Software é o desenvolvimento de processos de auxílio às atividades que envolvem sistemas. Hoje temos disponíveis processos para aquisição, manutenção, desenvolvimento de softwares entre outros. No que diz respeito a processos de desenvolvimento, a utilização destes vem de encontro a idéia de se desenvolver softwares seguindo diretrizes, de forma sistemática.

Muitas vezes empresas diferentes usam processos diferentes para chegar a um mesmo produto. No entanto, alguns processos são mais adequados para certas situações que outros. Por isso, é importante saber adotar o processo adequado ou adaptar o processo em uso quando necessário.

## **Projeto de software**

O desenvolvimento de um sistema é feito dentro de um projeto. O termo “Projeto de Software” diz respeito a um sistema a se desenvolver dentro de um determinado esquema, com um planejamento e objetivos bem definidos. Todo projeto tem: data de início e fim, uma equipe de desenvolvimento, um responsável (gerente de projetos) e outros recursos.

As decisões de níveis mais altos são tomadas pelo gerente de projetos, ele é o profissional responsável por supervisionar o projeto e tomar as devidas decisões para que este segua ao fim dentro dos prazos e custos estimados e com o nível de qualidade exigido.

Um projeto bem definidos possui subdivisões (marcos do projeto) que permitem avaliar seu progresso para corrigir possíveis desvios. Esta avaliação normalmente é desenvolvida pelo gerente de projetos, embora apoiado por toda equipe de desenvolvimento.

## **Etapas de desenvolvimento um sistema**

Muitos relacionam a atividade de desenvolvimento de sistemas à programação, mas desenvolver sistemas está longe de ser apenas programar. Existe uma série de etapas que antecedem e sucedem a codificação e que são fundamentais para produção de sistemas de qualidade. Basicamente temos as seguintes etapas:

- Antes de iniciar o desenvolvimento: elaborar uma proposta de desenvolvimento.
- No início do desenvolvimento: elaborar o planejamento do projeto.
- Desenvolvimento:
  - Especificação de requisitos;
  - Análise de requisitos;
  - Projeto;
  - Implementação ou codificação;
  - Testes.
- Após o desenvolvimento: manutenção.

**Elaboração de propostas:** uma das primeiras atividades de um projeto é desenvolver uma proposta do que o projeto se propõe a fazer. Esta precisa fundamentalmente mostrar os objetivos que o projeto pretende atingir e estimativas de prazos e custos. Para elaboração desta proposta é preciso que a empresa tenha feito um levantamento mínimo do que o cliente precisa. Este levantamento precisa ser bem dosado, pois a proposta pode não satisfazer o cliente e o projeto não ser levado a frente. No entanto, é preciso conhecer satisfatoriamente o problema para não subestimar as necessidades do cliente. Encontrar este “meio-termo” do quanto investir na elaboração de uma proposta para um projeto que precisa ser bem entendido, mas pode não ser efetivado é uma das maiores dificuldades do gerente de projetos.

**Planejamento do projeto:** este deve contemplar os seguintes itens:

- Processo: como dito anteriormente alguns processos são mais adequados a uma dada situação do que outros. Logo, deve-se definir os processos a serem usados dentro de um projeto mesmo antes do seu início. Mesmo quando a empresa já possui um processo pré-definido, muitas vezes algumas adaptações são necessárias;

- Documentação: até mesmo em virtude e juntamente com a definição do processo decide-se também o que será documentado no projeto e como esta será atualizada;
- Pessoas: faz parte ainda deste planejamento definir as pessoas que estarão envolvidas no projeto;
- Cronograma: deve-se definir as atividades do projeto e quem ficará responsável por cada uma destas atividades;
- Recursos computacionais: hardware e software. É necessário confirmar se a empresa dispõe de todos os recursos de software e hardware necessários para a condução do projeto para não correr o risco de ter se interromper o projeto em algum momento por falta de algum recurso;
- Riscos: o levantamento e análise de riscos de um projeto é muito importante para que os riscos mais graves e possíveis de ocorrer possam ser vigiados de perto;
- Controle de qualidade: é importante definir como será feito o controle de qualidade do projeto.

Note que todos os aspectos acima citados são questões de decisão fundamental para um bom início do projeto e por isso constam no planejamento. Outra observação importante é que vários dos pontos citados envolvem decisões gerenciais e por isso são, na sua maioria, incumbências do gerente de projetos.

**Desenvolvimento:** planejado o projeto parte-se para o desenvolvimento propriamente dito. Este é constituído das seguintes etapas:

- Especificação de Requisitos: é a etapa onde se procura levantar e documentar todas as necessidades do cliente. Nesta etapa é que a equipe de desenvolvimento toma conhecimento do problema a se resolver;
- Análise: nesta etapa se estuda o problema levantado na etapa anterior. Este estudo é desenvolvido com o auxílio de alguns diagramas, logo esta é uma etapa de muita análise e modelagem;
- Projeto (Desenho): se as duas etapas anteriores são etapas de entendimento e estudo do problema, a etapa de Desenho visa projetar a solução do problema. Nesta são definidas questões tecnológicas, como será a arquitetura do sistema, projeta-se o banco de dados, as interfaces do sistema e etc;

- Codificação: neste momento basicamente se traduz (implementa) a solução projetada em uma linguagem de programação;
- Testes: não se pode delegar apenas aos testes a responsabilidade de encontrar erros no projeto. Atividades de revisão das etapas anteriores são fundamentais para detecção de erros no projeto. No entanto, os testes é a última oportunidade de se encontra problemas no sistema, a partir de então um erro apenas é percebido pelo próprio usuário, situação bastante indesejável e que pode ter conseqüências bastante desagradáveis para ambas as partes (empresa e cliente).

**Manutenção:** tudo o que se faz nas etapas anteriores busca evitar a manutenção. Deve-se tentar minimizar ao máximo o volume de manutenções em um sistema. Digo minimizar porque é quase impossível em um grande projeto não termos manutenção, seja para correção de defeitos ou para acréscimos de funcionalidades.

## **Ferramentas CASE**

CASE significa *Computer-Aided Software Engineering* (Engenharia de Software com auxílio de Computador), ou seja, são ferramentas para auxiliar nas etapas de desenvolvimento de sistemas. Estas incluem sistemas para ajudar na gerência do projeto (planejamento, estimativas, controle de versões), na modelagem do sistema (Ex.: Rational Rose, Poseidon, Jude, Microsoft Visio, Umbrello), na modelagem do banco de dados (Ex.: Case Studio, ERWin), nos testes do sistema, para prototipação, etc.

Estas ferramentas dispõem de recursos que facilitam sobremaneira várias tarefas que o desenvolvedor teria de fazer manualmente. Dentre estes recursos podemos citar geração de código, de relatórios, gráficos, scripts de criação de banco de dados, entre outros.

Referências:

- SOMMERVILLE, I. Engenharia de Software, Addison-Wesley, 2003.
- PAULA Filho, Wilson de Pádua. Engenharia de Software: Fundamentos, Métodos e Padrões, (2ª ed.). São Paulo, LTC, 2003.