1

TEXTO BASE

ANÁLISE E MODELAGEM DE SISTEMAS

Faculdade IMPACTA

A análise

Sua Importância e as vantagens de Modelar

Prof. Renato de Tarso Silva

Resumo

Neste texto serão abordados temas introdutórios sobre o conteúdo das matérias de Análise e Modelagens de Sistemas, como a importância da Análise enquanto objeto de estudo, e as vantagens de praticar no cotidiano do processo de desenvolvimento de sistemas de software o desenvolvimento de modelos que o apoiem.

1.1. Início

A competência mais necessária para melhor aproveitar-se da "mágica" da computação é "*saber instruir*". Computadores realizam instruções cada vez mais rápida e eficazmente, e fazem exatamente o que lhes instruem, mas instruí-los bem é essencial.

Existe uma frase interessante criada por uma uma das maiores mentes que se conhece: "Os computadores são incrivelmente rápidos, precisos e burros; os homens são incrivelmente lentos, imprecisos e brilhantes; juntos, seus poderes ultrapassam os limites da imaginação." (Albert Einstein).

Tenho dito a alunos que perguntam: "Professor, qual linguagem de programação devo aprender primeiro?", e respondo: "Tanto faz. Mas essencialmente, aprenda a especificar boas soluções". Esta provocação é legítima. O mercado atualmente tem alta oferta de desenvolvedores; porém, demasiadamente, o mercado anseia por profissionais com capacidades - em *mindset* analítico - que permitam a descoberta, a especificação, e a solução de problemas. E isso é independente de ferramentas ou tecnologias.

Assim, iniciarei o passar do aprendizado com que dá o poder de conceber soluções para posteriormente melhor construí-las, em projetos mais assertivos.

1.2. A Análise

Quando se fala de análise de sistemas, estão englobados uma série de conceitos e ferramentas consagradas no mercado de desenvolvimento de software. Elas permitem a maior elucidação de problemas e necessidades, para melhor se especificar soluções de software através de artefatos bem construídos e modelos que apoiem o desenvolvimento de sistemas, atingindo maior eficiência e menor custo.

Realizar a análise de um sistema é o que permite a descoberta e a visualização das essenciais necessidades que devem ser sanadas pela solução de software a ser criado. De acordo com McMenamin (2011):

> A finalidade da análise de sistemas é fornecer uma declaração dos requerimentos verdadeiros do sistema que será construído. um requerimento, verdadeiro é uma característica que o sistema deve ter, qualquer que seja a tecnologia utilizada para implementá-lo.

A fase de análise de sistemas permite aprofundar entendimentos dos problemas do cliente. É a que apoia a realização e a validação dos requisitos do sistema, bem como a licitação de requisitos de software. McMenamin (2011) afirma que "(...) na essencial atividade de análise de sistemas, este problema é enorme: descobrir a verdadeira essência de um sistema, provavelmente, é a tarefa mais importante que um analista tem a cumprir, e também a mais difícil".

A linguagem utilizada para desenvolver tais artefatos, e diagramar modelos de sistemas e de software, é a linguagem de modelagem oficial mantida pelo grupo OMG (*Object Management Group*). A Linguagem de Modelagem Unificada, ou UML (*Unified Modeling Language*), que vem sendo aprimorada há décadas, é a mais reconhecida pela comunidade mundial de análise de sistemas.

1.3. A Falta de Análise

Analisar um sistema e programá-lo é como arquitetar um prédio e construí-lo. Se o engenheiro responsável não planejar sua construção, a chance do prédio ruir, ou nem ser finalizado, é altíssima - um crime. Com sistemas de software, é similar.

Quase inevitavelmente, o que se vê é: um bom time de desenvolvedores criando más soluções, por mal conhecer o problema que deveria solucionar. Ou descobre-se a verdadeira essência do sistema que se planeja construir, ou há 100% de chances de erros.

Na figura 1.1, aparece um balanço na árvore - tão desejado no verão! - que foi entregue no inverno; daí, é irreconhecível e fora da janela de oportunidade. Esta imagem ilustra os impactos da falta de análise, e como o projeto pode ser prejudicado e distorcido.

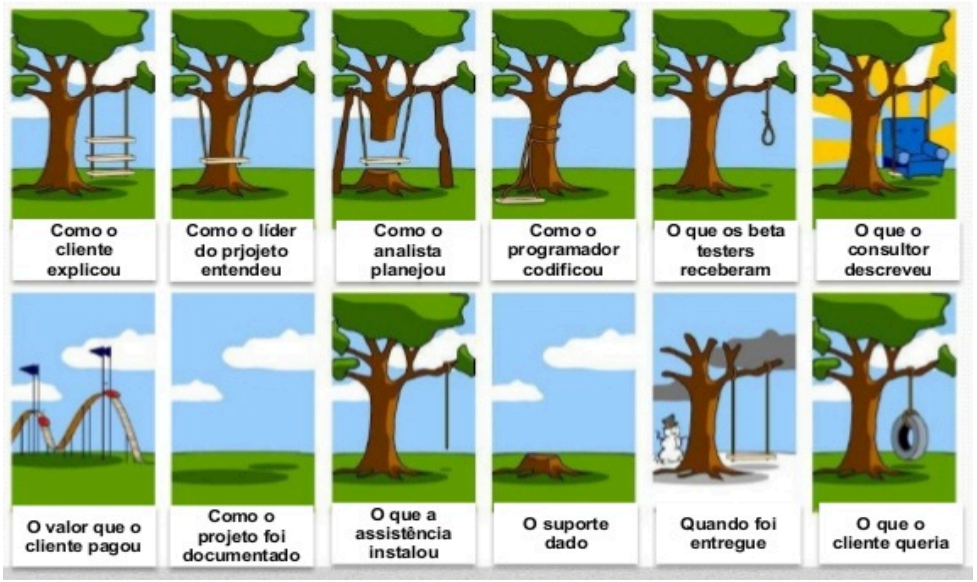

Figura 1.1. Impactos da falta de análise

Apenas após se analisar o sistema, e bem entendê-lo, as projeções do sistema são iniciadas. Se denota comportamentos, se prevê funcionalidades e estruturas dos componentes e dos objetos do sistema.

Com a finalidade de manifestar em notações gráficas estas necessidades de implementações, cria-se modelos que expressem o sistema.

1.4. Necessidade e importância da Modelagem

A criação de modelos de sistemas aumenta exponencialmente entendimentos de reutilização de conceitos e de funcionalidades de um sistema, também eleva sua manutenibilidade, agilizando o acesso ao conhecimento da equipe de desenvolvimento.

A figura 1.2 abaixo reflete um exemplo de como modelos podem ser utilizados para criar objetos que obedecem um padrão, mas mantêm características particulares.

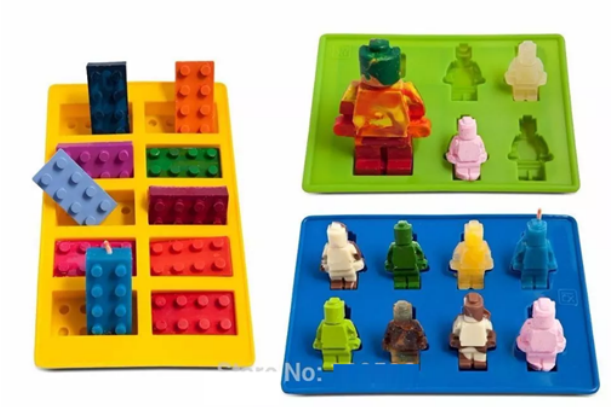

Figura 1.2. Modelos e reutilização

A modelagem de sistemas também permite que vários fatores cumpram-se antes do início da construção de um sistema de software, como:

- **Decisões documentadas.** Artefatos explicam como as decisões do projeto foram tomadas por *stakeholders*;
- **Avaliar diferentes perspectivas**. Obtém-se mais pontos de vista sobre o sistema e duas propriedades;
- **Reduzir custos.** A medida que se analisa, se alavanca aproveitamento de tempo e performance da equipe;
- **Aferir coerência e coesão.** Conceitos e elementos do projeto que estejam alinhados antes mesmo de iniciar codificação.

Um sistema bem desenvolvido oferece boa escalabilidade e alta adaptabilidade, características que um sistema mal concebido terá dificuldades de entregar. Após a fase de análise, consegue-se visualizar se existe coesão nos objetos especificados, e o prejuízo da incoerência entre componentes sistêmicos são, por exemplo, uma aplicação pouco escalável ou inflexível.

De acordo com Pressman (2011), "conforme aumenta a importância do software, a comunidade da área tenta desenvolver tecnologias que tornam mais fácil, mais rápido e mais barato de desenvolver um programa de computador de alta qualidade."

Mercado afora, se nota um desconhecimento de conceitos consagrados. Assim, cria-se soluções de software mal planejadas, análogas a favelas - enquanto provedores de funcionalidade e segurança - pois frequentemente "puxadinhos" são desenvolvidos e prejudicam a integridade das aplicações.

O auto-convencimento de que se deve ser "prático", e por logo a "mão na massa", agrada a ânsia que se tem por uma solução funcionando. Não há problema nesse pensamento, por sinal, pratica-se muito esta postura, mas deve-se conhecer os riscos e os custos disso que, inevitavelmente, serão assumidos.

Determinadas afirmações vão ao encontro à responsabilidade que devem ser minimamente assumidas em projetos de software, como: "não dá para ser ágil e o sistema", ou "vou logo para o código, desenhar é perda de tempo". Na realidade, como já citado anteriormente, ganha-se tempo, segurança e estabilidade na solução a medida em que se analisa e se modela o sistema. Aqui se fala principalmente em sistemas

complexos e/ou críticos, pois há de se concordar que desenvolver aplicações extremamente simples ou um site *single page* não cobra muita análise.

Modelar um sistema tem a mesma importância, e ganhos similares, de arquitetar um prédio industrial, por exemplo. Não é concebível, nos dias atuais, iniciar um projetos de alto risco e/ou complexidade sem o mínimo de planejamento. Caso contrário, existe um risco total de ocorrerem desastres funcionais, ou fatais, neste projeto. Em um empreendimento como o citado, é essencial que se faça representações de sistemas como: hidráulicos, elétricos, incêndio, etc.

1.5. Recapitulando a Computação

É curioso perceber como em meio século as formas como os sistemas são construídos evoluíram, especialmente no que tange análise e modelagem de sistemas, principalmente no processo de desenvolvimento de software.

Nos anos 50, lidamos com a construção de sistemas de maneira intangível, com programação em baixíssimo nível, em linguagem de máquina, junto à ínfima modelagem e inexistência de convenções para se realizar análises e criar modelos de sistemas. Pouco tempo depois, nos anos 70, depara-se com a popularização do computador pessoal, ou PC, com isso o desenvolvimento complexo de sistemas de software foi inevitável, mas isso ainda acontecia de maneira estruturada, em um paradigma procedural de codificação. Nos anos 80 apareceram as interfaces de interação entre humano e computador (IHC) popularizaram acentuadamente o uso de computadores em vastos tipos de indústria, trazendo um surgimento de soluções de software muito mais sofisticadas.

Nos anos 90 o início do paradigma orientado a objetos revolucionou o mercado de desenvolvimento de software - junto ao surgimento das primeiras iniciativas da criação de uma linguagem unificada de modelagem de sistemas. E nos anos 2000 já percebe-se o amadurecimento do paradigma orientado a objetos, com surgimento de Frameworks que amplamente viabilizam o desenvolvimento e implantação sistemas de software, bem como nota-se o aumento exponencial do uso da UML.

Esta evolução permite o vislumbre de quão maior é a capacidade de se desenvolver software na contemporaneidade, vistas as poderosas ferramentas amadurecidas ao longo do tempo e a consagração de métodos nas comunidades de software, suportando melhores práticas no processo de construção de software.

Referências

McMenamin, S. and Palmer, S. (1991) “Análise Essencial de Sistemas”. Ed. McGraw Hill.

Pressman, R. (2011) “Engenharia de Software”. Ed. McGraw Hill.