

Capítulo 1

Victor Wichmann Raposo - 9298020

1.1. A partir da comparação do desempenho dos sites *Amazon.com* e *HealthCare.gov*, o livro mostra que o método de desenvolvimento de um software é importante. Além disso, introduz o método que será ensinado nos próximos capítulos, o desenvolvimento ágil de software. Esse método permite equipes pequenas desenvolverem software competitivos, além de fazer os programadores trabalharem com código já existente.

1.2. Baseando no planejamento de projetos de engenharia civil, criou-se o método de desenvolvimento de software *Planeje-e-Documente*, que, a partir do planejamento e documentação extensiva, visa ser previsível em relação aos custos, tempo e qualidade. Então, apareceram diferentes ciclos de vida com essa metodologia, como o ciclo de vida em Cascata, caracterizado por um intenso planejamento antes da implementação e várias etapas que devem ser seguidas em sequência. Há também o ciclo de vida em Espiral, que consiste em etapas que serão executadas em quatro iterações, produzindo, assim, quatro protótipos. Outro ciclo de vida é o RUP, que tenta aproximar mais o cliente fazendo com que cada etapa tenha múltiplas iterações.

1.3. É apresentada uma metodologia baseada no Manifesto Ágil, que diferente do Planeje-e-Documente há um ênfase em iterações e feedback do cliente. A ideia desse ciclo de vida é ter um protótipo incompleto, porém funcionando que será aperfeiçoado conforme as iterações, levando em conta a história dos usuários e o resultado dos testes. No início esse método foi recebido com críticas, no entanto, pesquisas mostraram que ele é bom em casos de projetos pequenos.

1.4. Há o esclarecimento do que Arquitetura orientada a serviços (SOA) significa, uma abordagem para o desenvolvimento de software na qual há vários componentes interoperáveis, que podem ser usados independentemente ou recombinações de diferentes formas. As desvantagens da SOA são: os custos por invocação são maiores, afetando o desempenho, além de ser possível haver falhas parciais. No entanto, a divisão em pequenos serviços permite que cada um deles seja tratado com o desenvolvimento ágil.

1.5. O SaaS é uma combinação da SOA com a Internet, ou seja, o software e os dados são um serviço na Internet. Esse tipo de software gera muitas vantagens para ambos os

clientes no uso dos serviços e os provedores no desenvolvimento deles. Para desenvolver um SaaS é muito popular usar o método ágil, então existem muitos frameworks que dão suporte a esse desenvolvimento, esse livro utiliza o Ruby on Rails, pois tem uma comunidade que adotou o ciclo de vida ágil, então possui várias ferramentas úteis para esse tipo de desenvolvimento.

1.6. A computação em nuvem é composta por conjuntos de computadores de pequena escala conectados, chamados de *clusters*. Um SaaS precisa de um meio de comunicação, fornecido pela internet, mas também precisa ter escalabilidade e disponibilidade para os usuários. A computação em nuvem é um bom meio de fornecer essa escalabilidade e disponibilidade que o SaaS precisa, ainda mais que atualmente existem provedores desse tipo de infraestrutura.

1.7. Um software pode ser muito duradouro, mas para isso ele deve ser atualizado e melhorado. Há algumas classificações de códigos: *código legado*, que é um código antigo, porém ainda atende as necessidades dos clientes, mas ele tem uma dificuldade de evoluir; *código belo*, tipo de código duradouro e fácil de evoluir; e *código de curta duração inesperada* que são brevemente descartados.

1.8 Analisamos a qualidade de um software por meio de testes, eles nos permitem verificar se o software atende a especificação e validar se ele faz o que o cliente deseja. Como é inviável fazer testes de maneira exaustiva, divide-se em diferentes processos: teste de unidade, verifica se os procedimentos ou métodos funcionam como esperado; teste de módulo, testa várias unidades de um mesmo módulo; teste de integração, garante a comunicação entre as interfaces; e teste de sistema, que verifica o programa integrado.

1.9. Quatro mecanismos fundamentais para melhorar a produtividade de programadores são: clareza via concisão, que é fazer o código ser facilmente compreendido; síntese de implementação; reúso de designs, como o uso de bibliotecas e programação orientada a objeto; e ferramentas que auxiliam a produtividade como compiladores, interpretadores, Makefile, entre outras.

1.10. O livro é dividido em duas partes, uma que explica o SaaS e outra o desenvolvimento ágil de software. O capítulo 2 explica a arquitetura de SaaS, o 3º ensina Ruby, o 4º e 5º ensinam Rails, o 6º ensina JavaScript, JQuery e introduz a ferramenta Jasmine; o capítulo 7 é sobre Projeto guiado por comportamento (BDD), introduz o conceito de Velocidade e mostra as ferramentas Cucumber e Pivotal Tracker; o 8º capítulo fala sobre desenvolvimento guiado por testes (TDD) e ensina as ferramentas RSpec, Autotest e SimpleCov; o 9º é sobre trabalhar com código existente, o 10º sobre

organização e trabalho em equipe, com a ferramenta Git; o 11º sobre padrões de projetos e introduz a notação UML; o 12º sobre implantação e melhora de desempenho e escalabilidade na nuvem.

1.11. O livro possui screencasts, que os autores dizem que são muito importantes. Uma prática boa para a aprendizagem é tentar os exemplos de código. No início de cada capítulo há a descrição dos conceitos mais importantes, no fim há a sessão de falácias e armadilhas. No fim de cada subseção tem questões e projetos.

1.12. O ciclo de vida ágil não é o melhor método de desenvolver software sempre e para todos os casos. Deve-se levar em conta os custos de design e testes, pois são substanciais no custo do projeto.

1.13. A base da programação extrema é, iterações curtas, fazer os testes antes do código e revisar o código constantemente ao trabalhar em pares. A nuvem, os frameworks e ferramentas e o desenvolvimento ágil são o que formam a engenharia de uma SaaS.

O que eu gostei mais

Gostei muito da contextualização a partir de exemplos reais e estatísticas apresentadas pelos autores, isso mostrou o poder que o ciclo de vida ágil pode ter e que é realmente usado e dá resultados. Dessa forma fiquei ansioso para apreender a construir um SaaS com esse método.

O que eu não concordo

O único ponto de discordância foi o fato de não ter sido apresentado alguns pontos negativos da SaaS comparado com outros modelos de distribuição de software. E seria interessante uma seção dedicada ao crescimento de um SaaS já estabelecido.