BCC
Engenharia de Software
2018.2

Professor Rodrigo Andrade

Aula passada

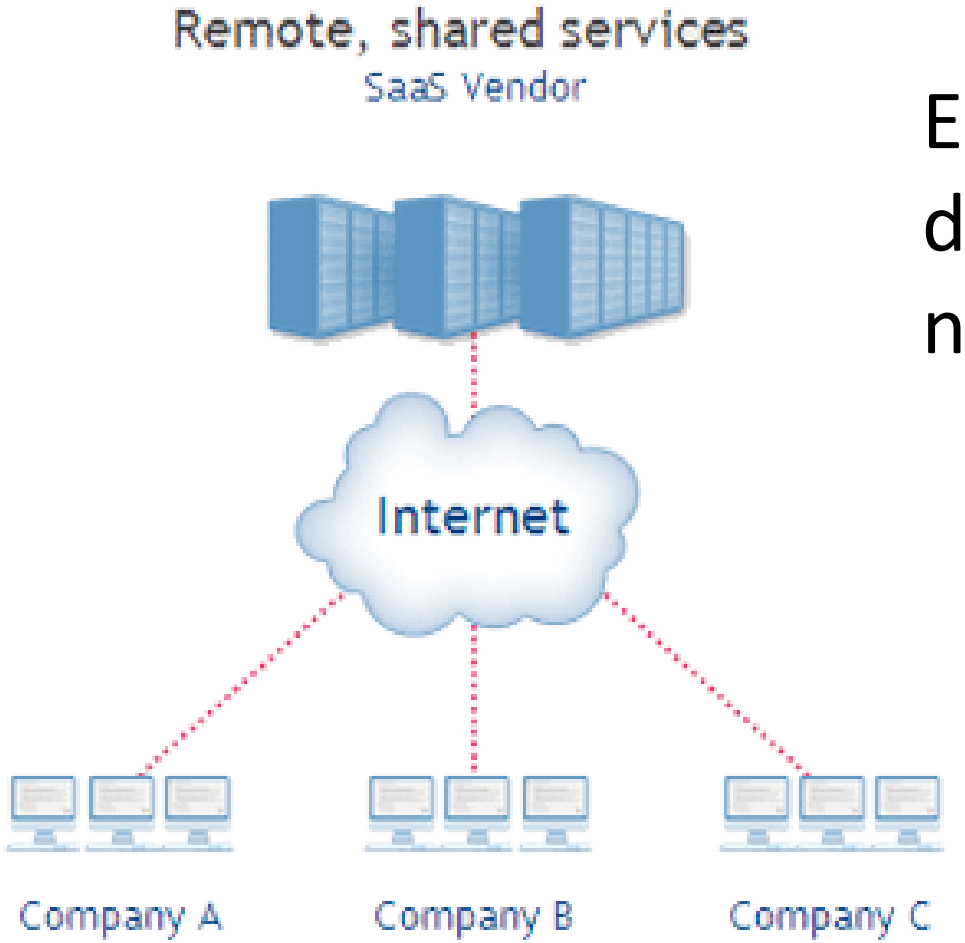
Quais os dois pilares da Engenharia de Software?

Quais os nomes dos três processos de desenvolvimento que vimos?

Citem algumas características do manifesto ágil

Software como Serviço (SaaS)

Software as a Service (SaaS) Model

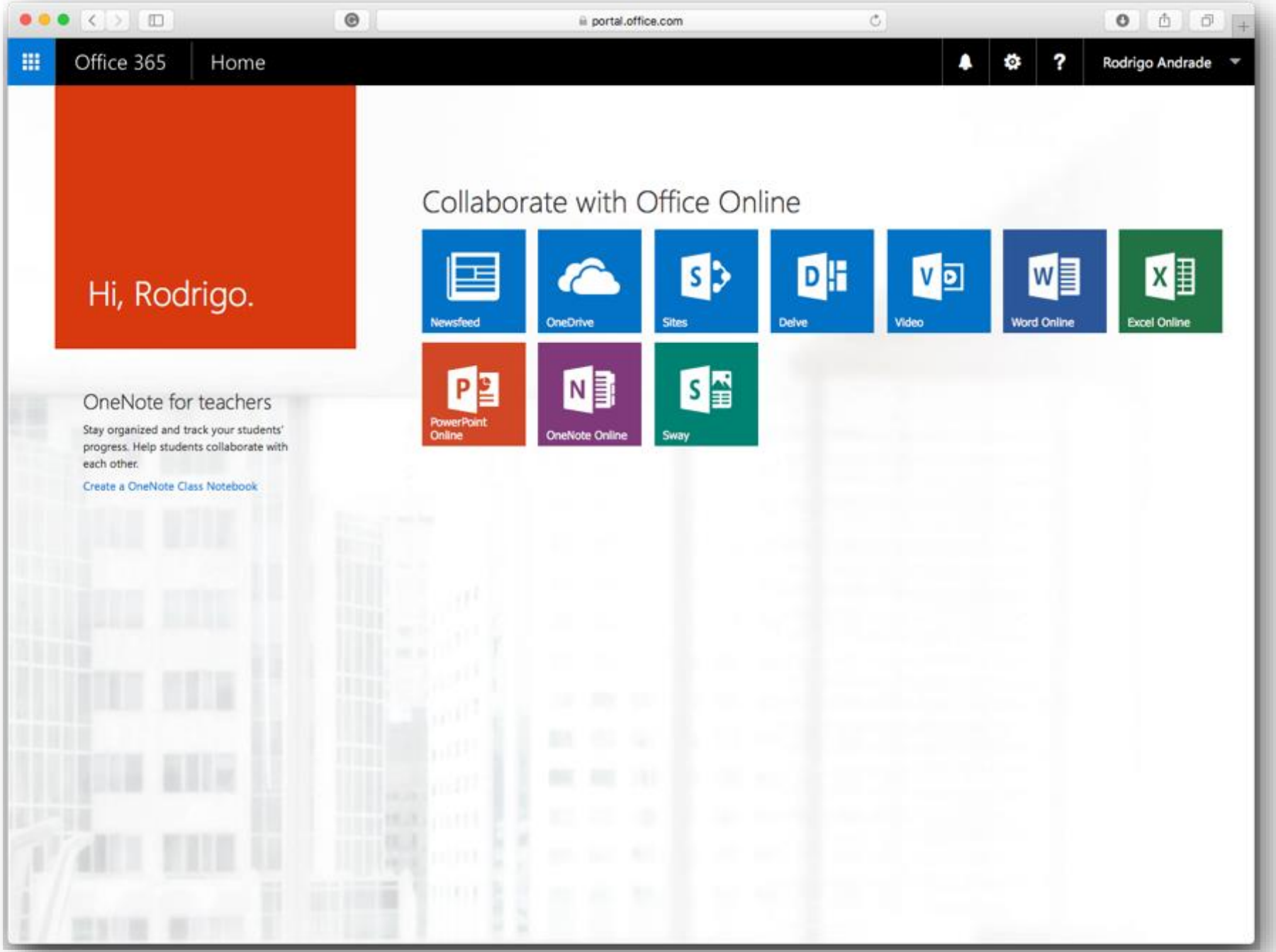


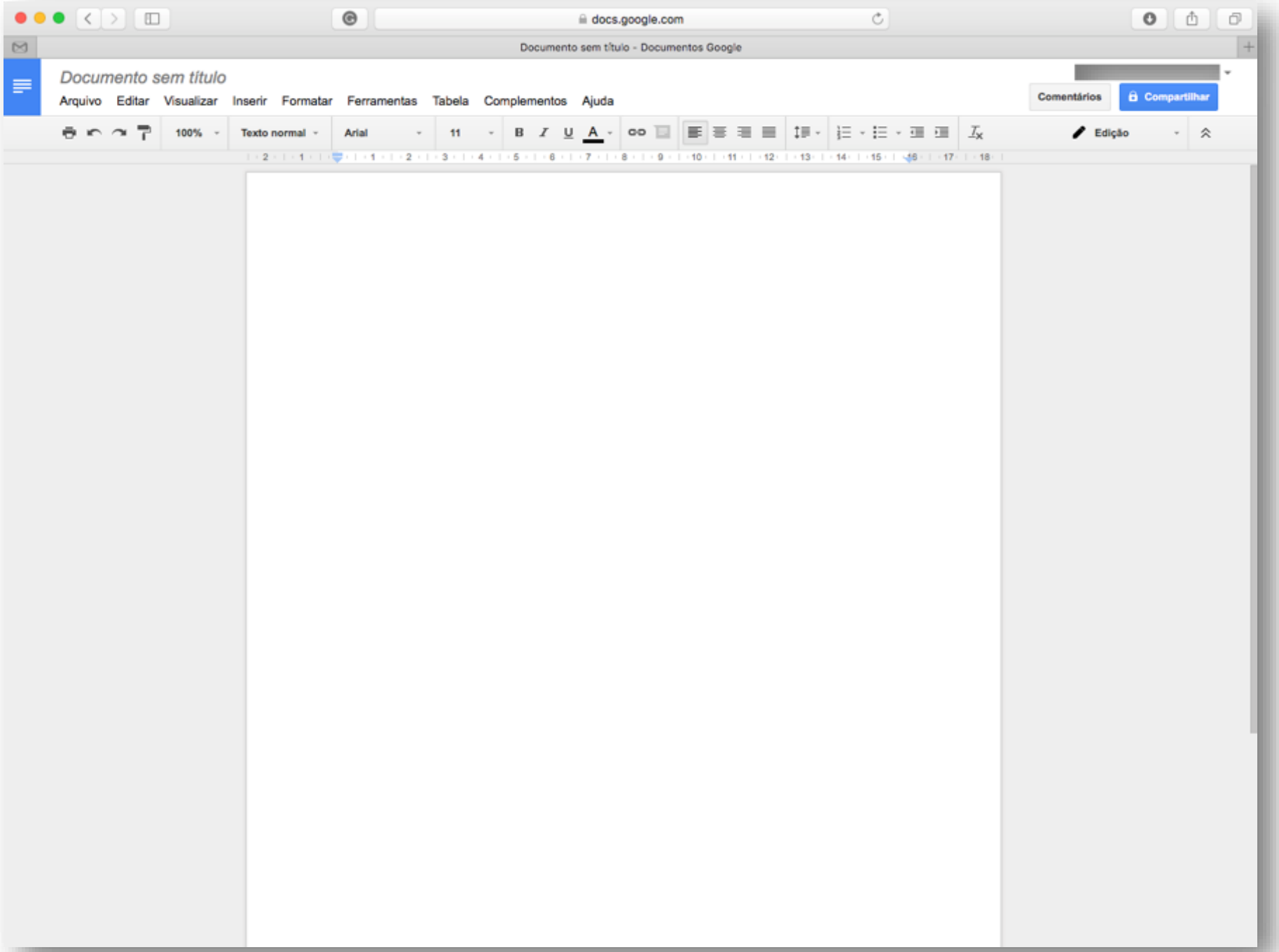
Entrega o software e os dados como um serviço na Internet

Não precisa ser instalado no dispositivo do cliente

Vantagens

- Clientes não precisam se preocupar se o hardware ou sistema operacional é compatível
- Clientes não precisam se preocupar com backup de dados
- Fornecimento de novas funcionalidades a qualquer momento
- Atualizações do software são transparentes aos usuários


portal.office.com
Office 365
Home
Rodrigo Andrade
Hi, Rodrigo.
Collaborate with Office Online
Newsfeed
OneDrive
Sites
Delve
Video
Word Online
Excel Online
PowerPoint Online
OneNote Online
Sway
OneNote for teachers
Stay organized and track your students' progress. Help students collaborate with each other.
Create a OneNote Class Notebook


docs.google.com
Documento sem título - Documentos Google
Documento sem título
Arquivo Editar Visualizar Inserir Formatar Ferramentas Tabela Complementos Ajuda
Comentários
Compartilhar
100%
Texto normal
Arial
11
Edição

youtube.com

Filmes - YouTube

YouTube BR

Enviar | Fazer login

- Início
- Em alta

O MELHOR DO YOUTUBE

- Música
- Esportes
- Jogos
- Filmes
- Notícias
- Ao vivo
- Destaque
- Vídeo em 360°

Procurar canais

Faça login agora para ver seus canais e recomendações!

Faça login

Filmes

Inscrever-se 44.682.001

Início | Vídeos | Playlists | Canais | Sobre

Novos lançamentos

The Beach Boys - Uma História De Sucesso (Legendado)

$ A partir de R$6,90

CC

Talento e Fé (Legendado)

$ A partir de R$6,90

CC

Refém De Um Criminoso (Legendado)

$ A partir de R$6,90

CC

Peter Pan (Dublado)

$ A partir de R$29,90

CC

Top de vendas

Como programar SaaS?

Arcabouço de programação	Linguagem de programação
Rails	Ruby
Grails	Groovy
Django	Python

E várias outras...

Ruby on Rails

Vantagens

- Produtividade
  - Reúso
  - Síntese
- Qualidade
  - Testes

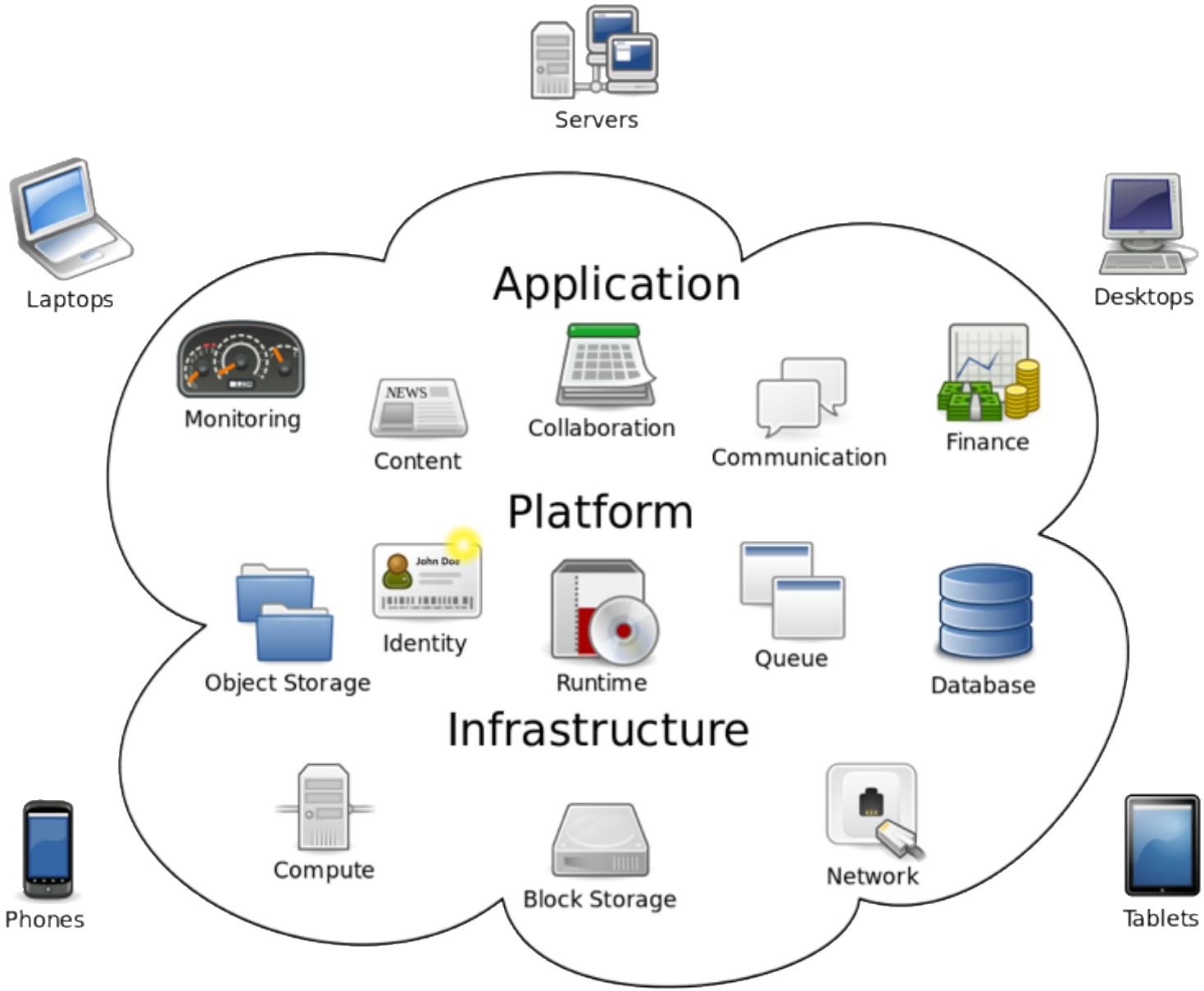
Computação em Nuvem (Cloud computing)

Características

- Tipo de computação baseada na Internet
- Recursos compartilhados, dados e informação são fornecidos a dispositivos sob-demanda
- Fornece hardware e armazenamento escalável e seguro para o SaaS
- Composta por clusters (aglomerados) de servidores

Critérios de SaaS

- Comunicação
  - Permite que clientes interajam com o serviço
- Escalabilidade
  - Lida com flutuações de demanda
- Disponibilidade
  - Serviços continuamente disponíveis



Cloud Computing


Dropbox


Google Drive


OneDrive


github
SOCIAL CODING


CloudBees

Escalabilidade

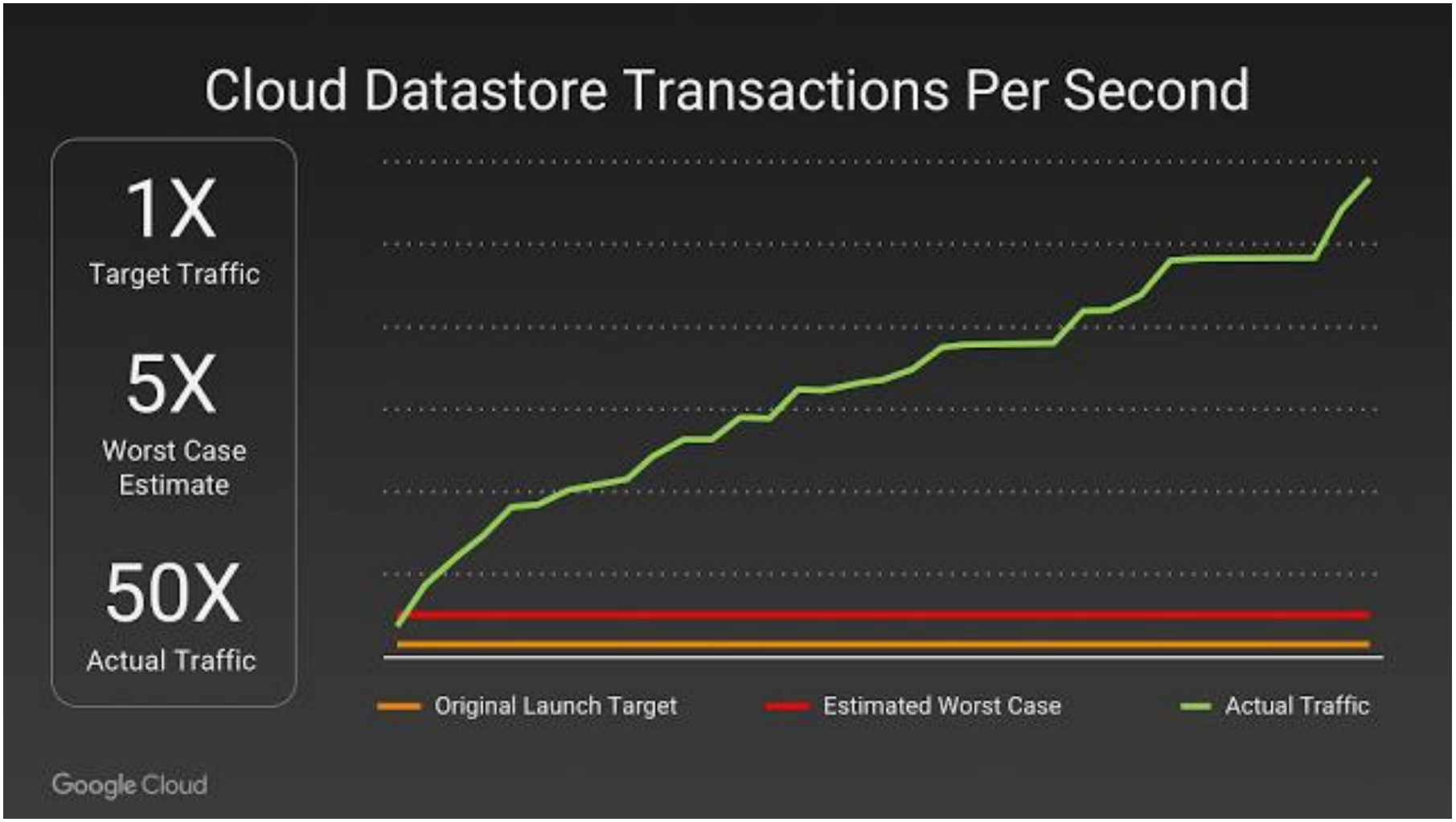
Se eu criar um SaaS e disponibilizá-lo, como lidar com uma eventual alta demanda?

Serviços públicos de nuvem

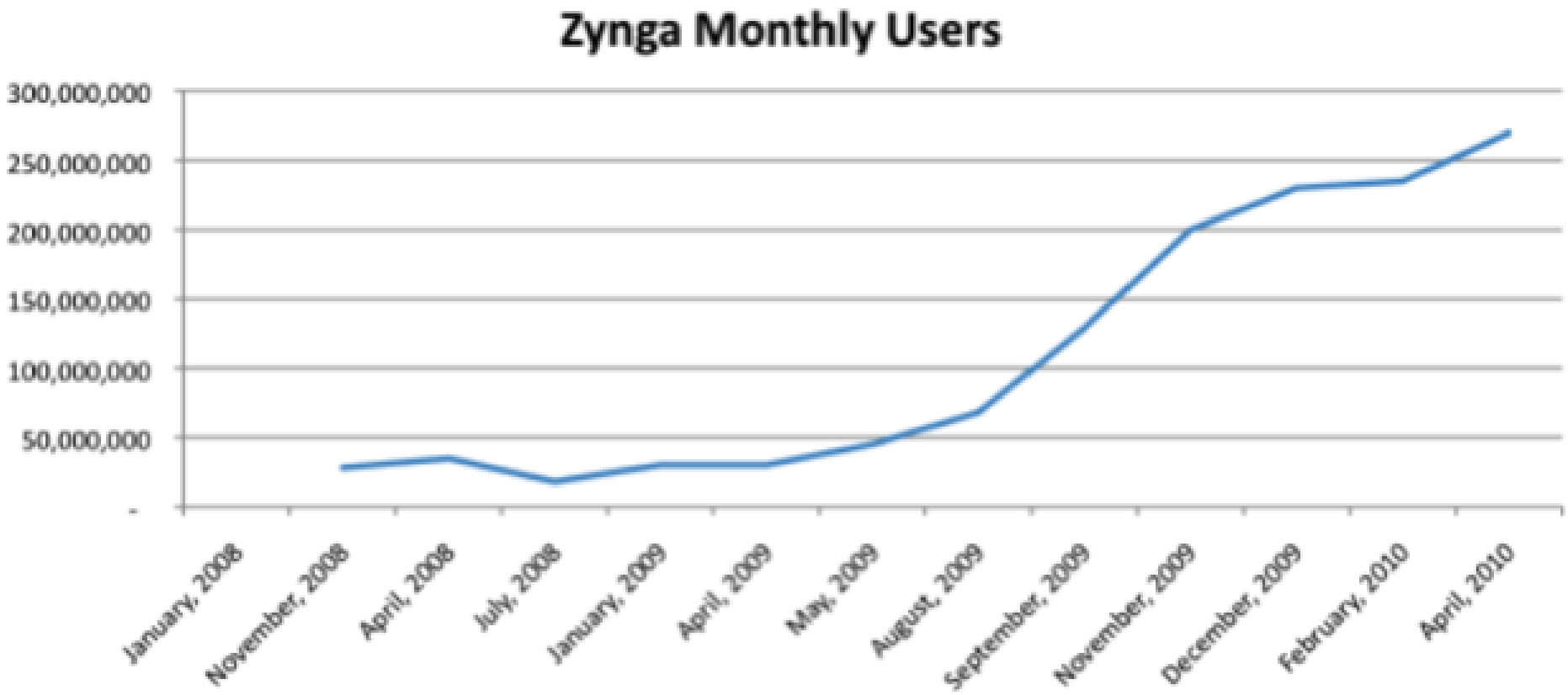
Google Cloud Platform Live

Microsoft Azure

Crescimento do Pokemon GO



Crescimento do Farmville



Disponibilidade

99,999%

Sistema de telefone público dos EUA

5 minutos de indisponibilidade por ano

Clusters

100
1000
10.000…

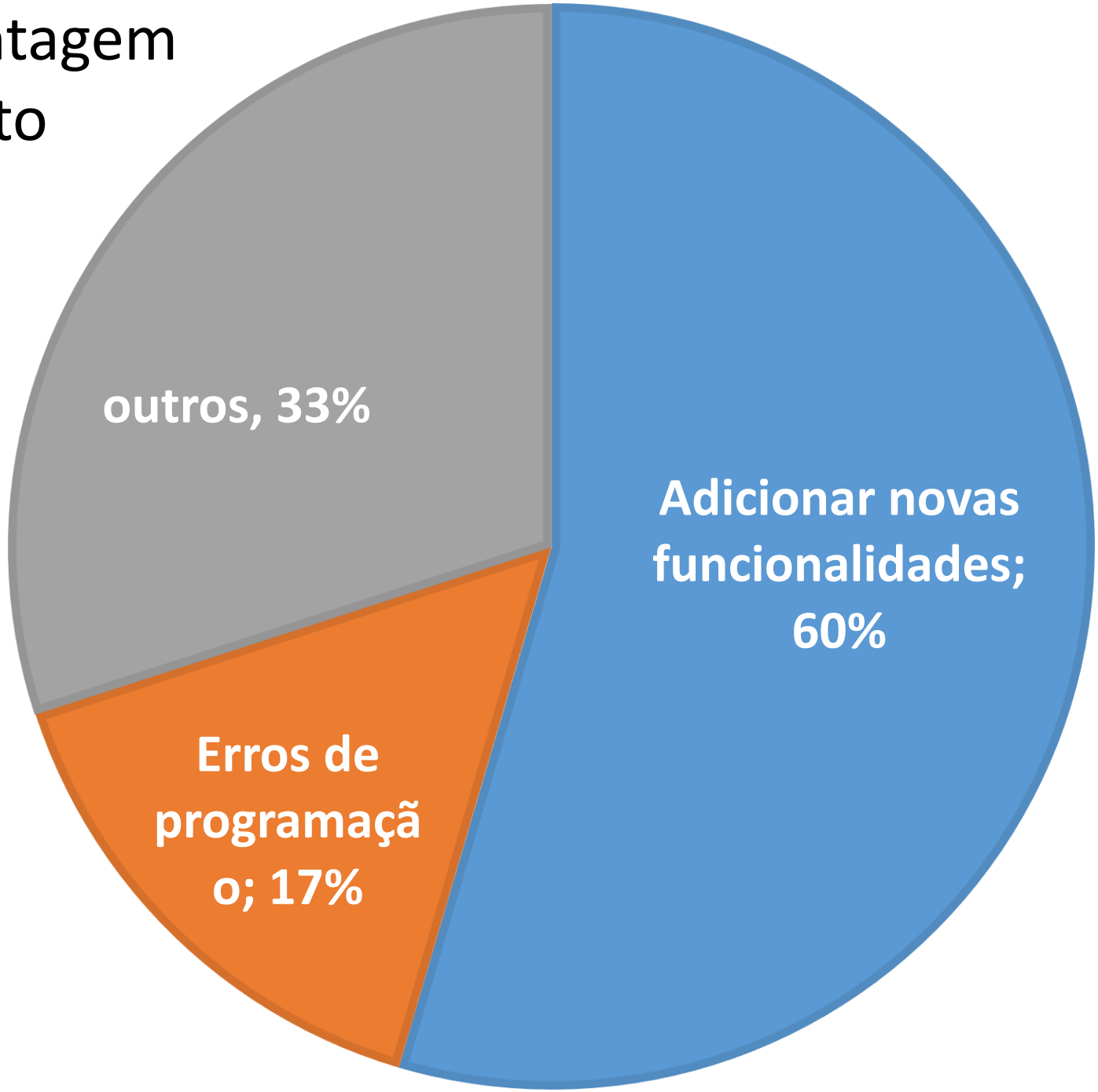
Resultando em serviços públicos de nuvem

Código belo x Código legado

Código legado

- Software que continua sendo usado pois atende as necessidades dos clientes
  - Software pode ser antigo
- Software bem sucedido

Porcentagem de custo



Código belo

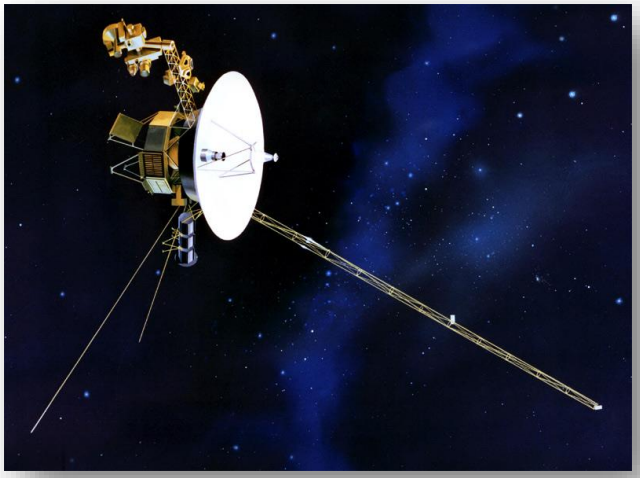
- Código duradouro e fácil de evoluir
- Usa padrões de projeto
  - Solução geral e reutilizável para um problema frequente
- Foca no reuso de componentes

Código mais antigo ainda em uso

- Mechanization of Contract Administration Services (MOCA)
  - Escrito em COBOL
  - 1958
  - Mantido até hoje
  - Integrado com vários outros softwares hoje em dia

There have been efforts in the past to build a full replacement for MOCAS, and they've sputtered due to cost, complexity, and transition planning.

Outros exemplos



Voyager

Lançada em 1977

Está a 20,8 bilhões de quilômetros de distância do Sol

Software ainda funciona!

Testes de software: Garantia de qualidade

Qualidade

- O quão "apropriado para o uso" o produto é
- Software
  - Importante para o cliente e para o desenvolvedor

Qualidade para o cliente

- Satisfazer as necessidades do cliente
  - Uso fácil
  - Respostas corretas
  - Não trava
  - Bom desempenho
  - ...

Qualidade para o desenvolvedor

- Facilidade para resolver erros e melhorar o software
- Modularidade
- Reúso
- Código bem escrito e fácil de ler

Controle de Qualidade (QA)

- Processos que levam à manufatura de produtos de alta qualidade
- Introdução de processos que melhorem a manufatura

Determinação da qualidade do software

- **Verificação**
  - O software foi construído corretamente?
  - A especificação foi seguida?
- **Validação**
  - O software construído é o certo?
  - É o que os clientes querem?

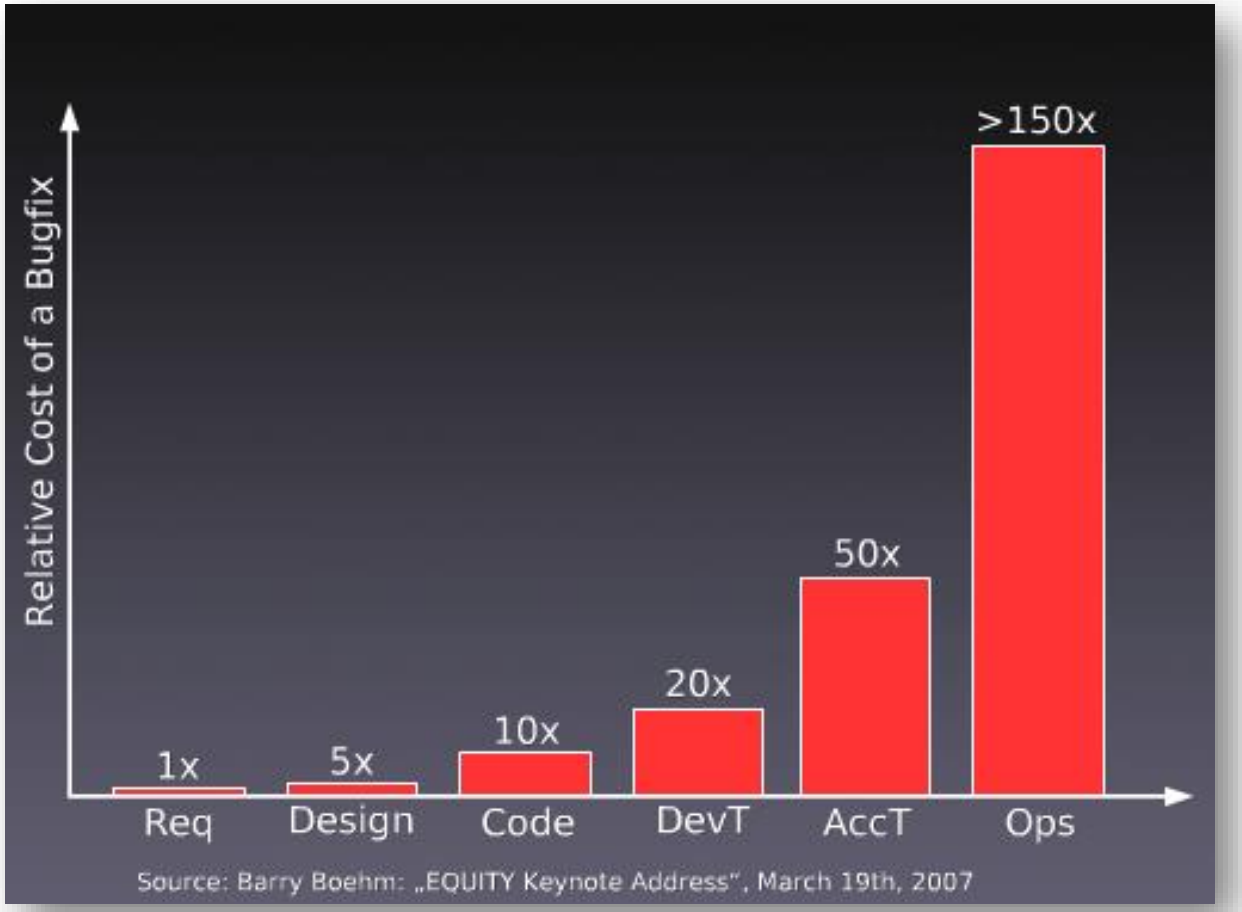
Protótipos de software auxiliam mais na verificação ou validação?

Qual a abordagem para verificação e validação?

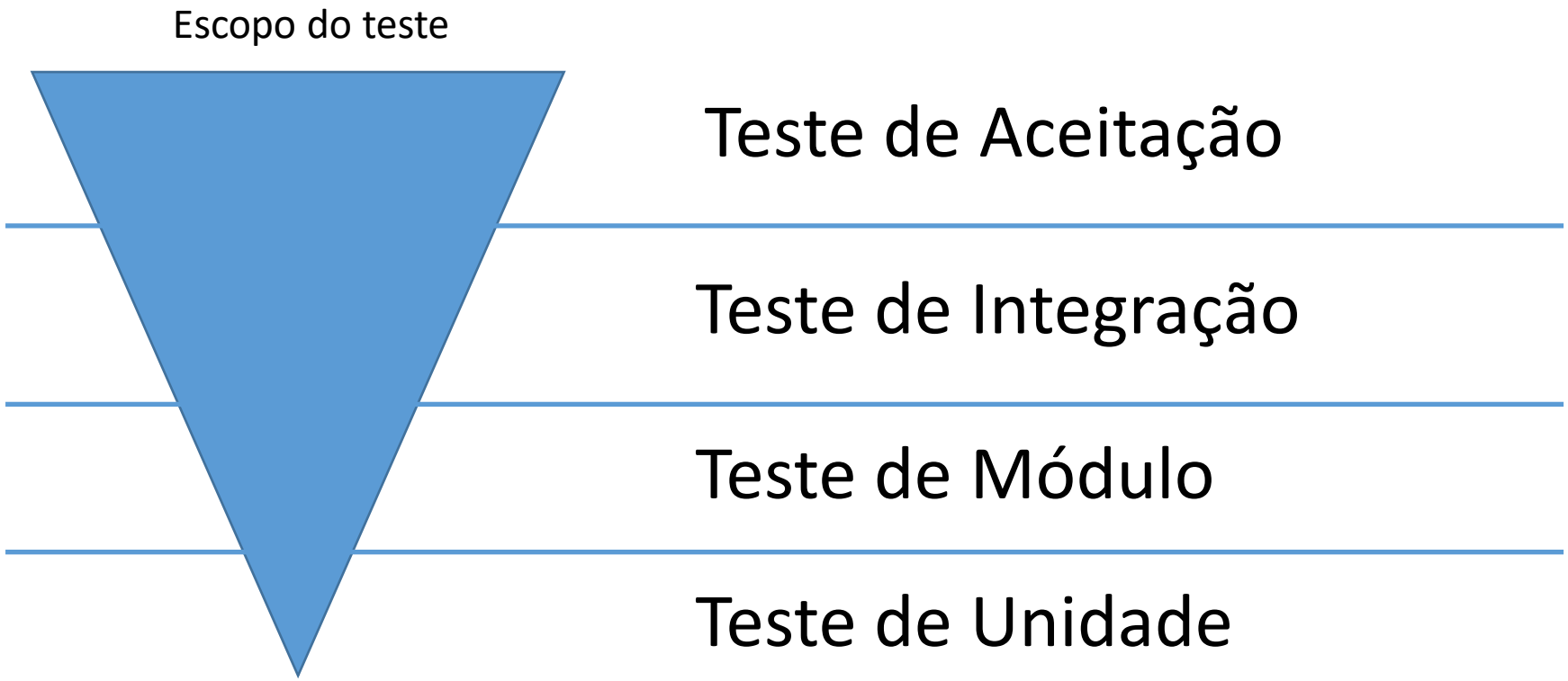
Testes de software

Testes de software

- Achar erros o mais cedo possível



Tipos de testes



Teste de unidade

- Visa assegurar que determinado método faça o esperado
- Menor unidade de teste
- Foca na lógica interna de processamento

Teste de módulo

- Testa vários componentes de um mesmo módulo
- Visa achar erros entre esses componentes
- Exemplo: testa várias classes de um mesmo pacote

Teste de integração

- Visa assegurar que interfaces entre componentes se comuniquem corretamente
- Exemplo: camada de dados se comunica corretamente com camada de negócio
- Bottom-up ou Top-down

Teste de sistema (aceitação)

- Verifica se o programa integrado atende às suas especificações
- Verifica se o comportamento do sistema está de acordo com o que foi solicitado pelos clientes
- Permite ao cliente aceitar ou rejeitar o sistema

Qual dos tipos de testes é mais adequado para Validação?

Técnica alternativa: Métodos formais

- Abordagem adicional para garantir qualidade
- Visa provar que o comportamento do código segue a especificação
- Tende a ser muito custoso
  - Exemplo 500 dólares por linha de Código
- Não são adequados para software com funcionalidades que mudam frequentemente

Produtividade

Mecanismos fundamentais

1. Clareza via concisão
2. Síntese
3. Reúso
4. Automação por meio de ferramentas

1- Clareza via concisão

- Linguagens de programação de alto nível
- Elevar o nível de abstração da linguagem
- Programas facilmente compreendidos
- Linguagens de programação
  - Expressar ideias com poucos caracteres

```
assert_greater_than_or_equal_to(a,7)
```

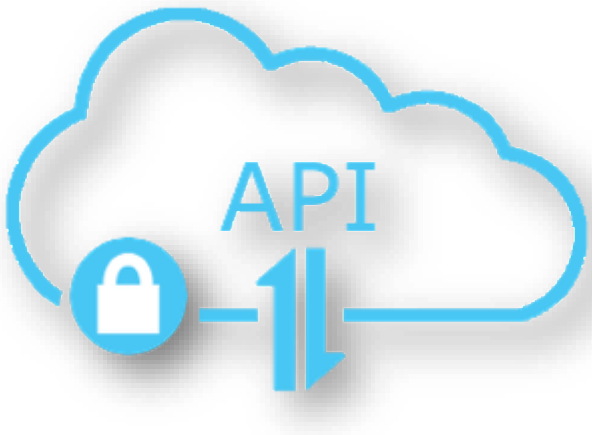
```
assert a <= 7
```

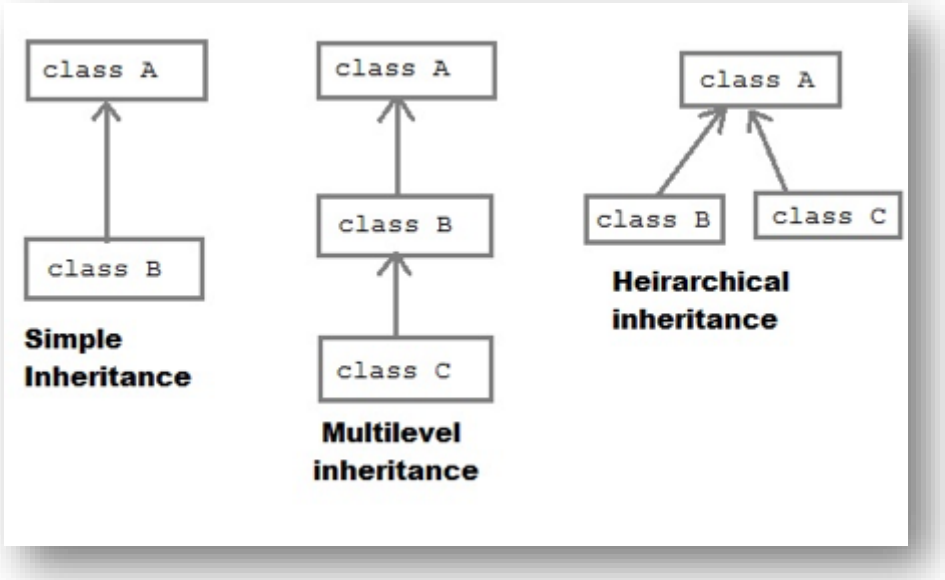
2- Síntese

- Implementação gerada ao invés de criada manualmente
- Geração de código
  - CRUD gerados automaticamente
  - Scaffold

3- Reúso

- Utilização de porções de projetos já prontos
- Evita escrever tudo a partir do zero





Reúso != Copiar e Colar

- Impossibilidade de alterar todas as cópias em todos os lugares
  - Dificuldade de corrigir erros
- Don’t repeat yourself (DRY)

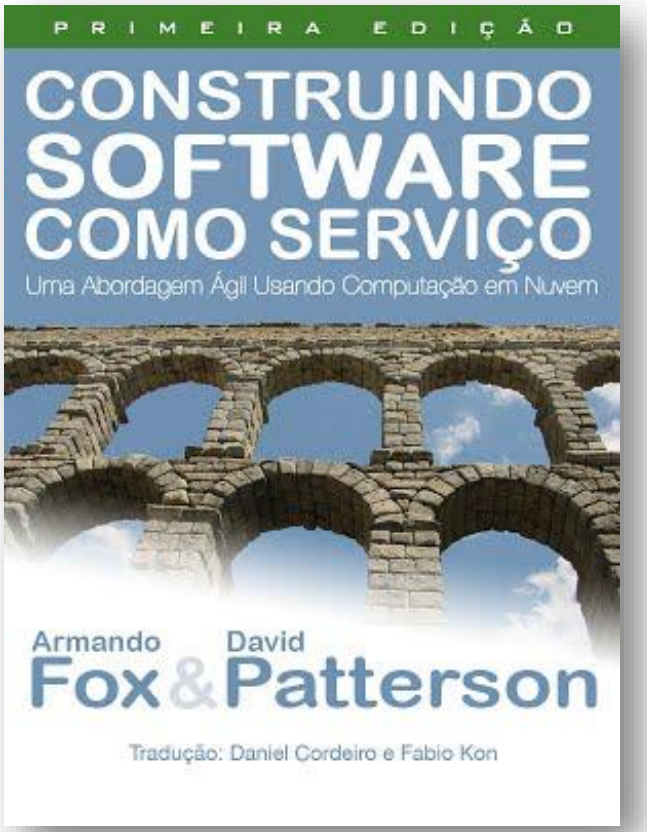
4- Automação

- Ferramentas para poupar tempo
  - Exemplo: Automatizar a transformação de histórias de usuário em testes de integração
- Necessário tempo para aprender a usar a ferramenta
  - Benefícios vêm depois

Resumo da aula

- SaaS
- Computação nas nuvens
- Tipos de testes (Qualidade)
- Produtividade

Referência



Capítulos 1.5 ao 1.15