RODRIGO AGUIAR ORDONIS DA SILVA

ABORDAGEM PARA O DESENVOLVIMENTO DE SOFTWARES ESCALÁVEIS FOCANDO EM DISPONIBILIDADE E DETECÇÃO DE FALHAS

São Paulo
2020

RODRIGO AGUIAR ORDONIS DA SILVA

ABORDAGEM PARA O DESENVOLVIMENTO DE SOFTWARES ESCALÁVEIS FOCANDO EM DISPONIBILIDADE E DETECÇÃO DE FALHAS

Trabalho apresentado à Escola Politécnica da Universidade de São Paulo para a conclusão do MBA de Transformações Digítais.

São Paulo
2020

RODRIGO AGUIAR ORDONIS DA SILVA

ABORDAGEM PARA O DESENVOLVIMENTO DE SOFTWARES ESCALÁVEIS FOCANDO EM DISPONIBILIDADE E DETECÇÃO DE FALHAS

Trabalho apresentado à Escola Politécnica da Universidade de São Paulo para a conclusão do MBA de Transformações Digítais.

Orientador:

Reginaldo Arakaki

São Paulo
2020

SUMÁRIO

RESUMO

ABSTRACT

1 INTRODUÇÃO

"A confiança perdida é difícil de recuperar. Ela não cresce como as unhas."

-- Brahms, Johannes

Com o passar do tempo, os *softwares* firacam cada vez mais importantes na sociedade e os negócios começaram a depender ainda mais deles. Alem dos computadores, há diversos outros dispositovos que nos permitem acesso a internet como por exemplo os celulares, os *tablets*, *video games*, as televisões, e entre vários outros dispositivos. O que facilitou o acesso a informação e consequntemente, a divulgação da informação. Com os sistemas ficando mais importantes, diversos serviços são realizados pela internet, como compras, negociações, comunicação, tranferências bancarias, entre outros.

Devido a isso, questões como quantidade de acesso, tempo de resposta e segurnaça da informação começaram a ficar cada vez mais importantes na concepção de um *software*. A queda de um sistema por alguns segundos, pode ocasionar em diversos problemas para uma empresa como, a perda de milhões de reais, a desvalorização da marca e causar uma reputação negativa para a empresa. Dependendo do motivo da queda, pode ocasionar o fim do sistema e por consequência o fim de um produto para a empresa. Outro ponto a ressaltar é que devemos entender se mesmo com o sistema funcionando corretamente, ele realmente está trazendo retornos positivos para a empresa? O usuário está gostando do que disponibilizamos para ele? Está falando bem ou mal do produto?

Para se preparar para estas situações inesperadas e manter o *software* funcionando, é importante manter o produto escalável. Não podemos controlar a quantidade de acessos no sistema, mas podemos controlar o quanto de recursos computacionais é disponibilizado para o *software*, em épocas em que os acessos aumentam, podemos aumentar a quantidade de processamento, em épocas que diminuem, podemos diminuir o processamento, assim controlamos o custo do sistema e garantimos sua disponibilidade. Não podemos tambem ter um *feedback* de todos os usuário de como está a experiência na utilização, muitas vezes nem da maioria deles, mas podemos analisar as ações dos usuários para entender como está sendo a sua experiência.

Outro ponto que não deve ser esquecido, é de que problemas vão acontecer, situações inesperadas que não vamos saber lidar no momento, brechas na segurança, um *bug* no

sistema, usuários perdidos na navegação, entre outros casos. Nestes casos devemos aprender com os problemas, descobrindo como identifica-los e corrigi-los. Uma vez descoberto, aplicamos testes automatizados para que eles não voltem a acontecer.

1.1 Motivações

Como podemos notar, criar um *software* se tornou uma tarefa complexa, garantir sua qualidade se tornou uma taréfa difícil, devemos realizar diversas ações para assegurar que nosso sistema vai gerar valor. Foi neste cenário que nos sentimos motivados.
Nossas motivações se definem em criar e manter produtos escaláveis, consequntemente, possibilitar o controle de seu custo com base na utilização dos usuários e na situação da empresa, demonstrando como utilizar a interação dos usuários para aprender como melhorar e engajando a utilização de testes automatizados como um processo de aprendizagem, detecção de falhas e de controle de qualidade.

2 METODOLOGIA DE PESQUISA

3 OBJETIVO

Este trabalho tem como objetivo apresentar uma aborgagem de como construir sistemas escaláveis, com foco em assegurar performance e detecção de falhas. Criando *softwares* com qualidade, gerando valor e que com base nos retornos do sistema, possibilitando novas visões de negócio e ocasionando no desenvolvimento de novos produtos.

4 FUNDAMENTOS CONCEITUAIS

4.1 Análise geral sobre projetos de software

4.1.1 Waterfall

4.1.2 Metodologias ágeis

4.1.3 DevOps

4.1.4 CI / CD

4.1.5 Testes automatizados

4.2 Qualidade de software

4.2.1 Requisitos funcionais

4.2.2 Requisitos não funcionais

4.3 Escalabilidade

4.3.1 Disponibilidade

4.3.2 Falhas no projeto

5 PROPOSTA

5.1 Como adquirir escalabilidade?

5.2 Como manter o sistema disponível?

5.3 Como identificar falhas?

5.3.1 O que fazer com as falhas identificadas?

6 RESULTADOS DA PROPOSTA

6.1 Um produto escalável

6.1.1 Um produto com custo dinâmico

6.2 Um produto disponível

6.3 Um produto com falhas planejadas

7 CONCLUSÃO

7.1 Resultados em relação ao objetivo

7.2 Trabalhos futuros

8 REFERÊNCIA BIBLIOGRÁFICA