

RODRIGO AGUIAR ORDONIS DA SILVA

TÍTULO

São Paulo
2020

SUMÁRIO

Resumo

1	Introdução	5
1.1	Motivações	5
2	Metodologia de pesquisa	7
3	Objetivo	8
4	Fundamentos conceituais	9
4.1	Análise geral sobre projetos de software	9
4.1.1	Waterfall	9
4.1.2	Metodologias ágeis	9
4.1.3	DevOps	9
4.1.4	CI / CD	9
4.1.5	Testes automatizados	9
4.2	Qualidade de software	9
4.2.1	Requisitos funcionais	9
4.2.2	Requisitos não funcionais	9
4.3	Escalabilidade	9
4.3.1	Performance	9
4.3.2	Falhas no projeto	9
5	Proposta	10
5.1	Como adquirir escalabilidade?	10
5.2	Como manter a o sistema performático?	10

5.3	Como identificar falhas?	10
5.3.1	O que fazer com as falhas identificadas?	10
6	Resultados da proposta	11
6.1	Um produto escalável	11
6.1.1	Um produto com custo dinâmico	11
6.2	Um produto disponível	11
6.3	Um produto com falhas planejadas	11
7	Conclusão	12
7.1	Resultados em relação ao objetivo	12
7.2	Trabalhos futuros	12
8	Referência Bibliográfica	13

RESUMO

Resumo...

1 INTRODUÇÃO

O ambiente de desenvolvimento de *software* passou por diversas mudanças durante o tempo. Assuntos como produtividade, qualidade, adaptação a mudanças e manutenção em tempo real, assuntos que é pesquisado até os dias de hoje.

Com o nascimento desses temas surgiu novas tarefas para serem realizadas, e com estas tarefas, novos cargos e papéis. Porém, para realizar estas novas práticas, muitas vezes é necessário diversas pessoas, possuindo diversos papéis diferentes, e com isso, a quantidade de membros em um time cresce, com membros de papéis variados tornando difícil manter a gestão e a comunicação entre o time.

[Buscar uma pesquisa para confirmar o aumento de pessoas com diminuição da produtividade]

Com a utilização das práticas de *devops*, ferramentas de automação e com a aplicação de inteligência artificial, buscamos possibilitar que a comunicação entre os membros do time sejam realizadas de forma mais fácil e mais rápida, sugerindo padrões a serem tomados pelo time e automatizando algumas tarefas.

1.1 Motivações

Nossa maior motivação, é possibilitar que os projetos de grande porte possa manter sua escalabilidade sem perder a produtividade. Garantir que os times sigam práticas para garantir a qualidade do produto que estão entregando e que um projeto que se inicie pequeno possa evoluir de forma natural.

[Buscar pesquisa sobre projetos que cresceram mal]

Através de nossas análises nas principais práticas de desenvolvimento e gestão de produtos de *software*, apresentamos formas para automatizar tarefas repetitivas com o intuito de manter os times concentrados na melhoria do produto e tornar a execução dessas práticas mais simples. Com isso, acreditamos que o desenvolvimento dos produtos serão mais produtivos e serão entregues com maior qualidade.

2 METODOLOGIA DE PESQUISA

3 OBJETIVO

Este trabalho se propõe à auxiliar no desenvolvimento de projetos de *software* ágeis focando em garantir a qualidade do código que está sendo produzido.

Em conjunto das práticas e ferramentas já utilizadas no mercado é apresentado uma especificação para garantir a qualidade de testes automatizados, focando em garantir que requisitos não funcionais como, performance, testes em massa, abstração dos testes, entre outros, estejam sendo garantidos na produção do *software*.

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 Performance

4.3.2 Falhas no projeto

5 PROPOSTA

5.1 Como adquirir escalabilidade?

5.2 Como manter a o sistema performático?

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