

Lucas Teixeira Boneto
2311197

1. Cost of Poor Quality Software in US

Esse artigo, construído enquanto o mundo vive a batalha da pandemia da Covid-19, tem como objetivo alertar para os gastos excessivos em softwares de má qualidade nos Estados Unidos e demonstrar as principais fraquezas e vulnerabilidades envolvidas nessas tecnologias, além de enfatizar a importância em construir bons softwares, a fim de abarcar o custo gasto pela ~~população~~ nação, que estaria estimado em 2.08 trilhões de dólares no ano de 2020.

Nesse sentido, o relator ~~mas~~ alerta que não existe uma quantidade suficiente de desenvolvedores de software para criar e modificar os softwares que os usuários necessitam, sendo 22% a menos do que é preciso, porém há uma previsão feita com estudos de que ~~em~~ na próxima década, essa necessidade será suprida, visto que mais pessoas estão se especializando na área. Por outro lado, a complexidade dos programas, aliada à dificuldade da profissão, contribuem para o problema. ~~Por fim~~, o autor conclui que serão necessárias novas ferramentas para automatizar o desenvolvimento, especialmente as relacionadas à manutenção.

Seguindo tal lógica, o aumento da digitalização criou uma cultura de entregar o maior valor de produto, em alta velocidade, sem sacrificar qualidade e segurança, fazendo um equilíbrio entre eles. Em contrapartida, a partir das análises feitas no artigo, tal equilíbrio parece não existir.

Referente ao tema, é citada as principais categorias de custo desses softwares, que são:

1. Projetos de TI mal sucedidos
2. Baixa qualidade em sistemas legados
3. Falhas em Software operacional
4. Cibersegurança e Deficiência Técnica

Além disso, as maiores falhas de software incluem ::

1. Ataques ransomware
2. Ataques de segurança
3. Interrupções em TI
4. Vazamento de dados

Portanto, após a ~~disse~~ análise descrita do autor, ele traz algumas estratégias para maior longevidade da aplicação :

1. Encapsular - ocultar/ isolar detalhes
2. Rehospedar - mover sistemas do mainframe
3. Replataforma - acelerar com novo hardware
4. Reparar - corrigir bugs
5. Refatorar - reduzir falhas futuras
6. Reestruturar - nova tecnologia
7. Reconstruir - ajustes finais
8. Substituir - solução nova ou SaaS