

Teste de Software - de Curso Engenharia de Software - 2020.1

Debate Estruturado 1 - Fase de Execução - Análise de Assertivas

Propositores: Gustavo Vieira, Ronan de Souza, Willer Santos | Contrapositores: Fernando Jean, Pedro Araujo

- Pode-se afirmar que o Engenheiro de Software deve -

	Proposição	Contraposição	Síntese
Ex:	<i>Priorizar a criação de casos de teste para todos os módulos igualmente</i>	<i>Não. Em criar casos de teste para “todos os módulos igualmente” é melhor priorizar os módulos mais complexos, onde geralmente há maior chances de defeitos.</i>	A/D
1	Averiguar testes para módulos que geraram poucos erros antes da release, pois geralmente surgem mais erros pós-release.	Sim, Não apresentam um aumento na atividade de teste antes das releases, mas sim em períodos de testes no desenvolvimento da história de usuário	A
2	Utilizar de sistemas de controle de versão para definir casos de teste mais acurados para os módulos	Não, pois realizando testes para módulos individuais em versões separadas não garante que todos os módulos funcionarão corretamente quanto integrados.	D
3	Realizar o processo de teste de forma mais próxima do processo de desenvolvimento, fazendo as duas code base evoluírem em conjunto.	Não. É extremamente difícil fazer com que as duas code bases evoluírem juntas pois os testes são despriorizados, a arquitetura utilizada pode dificultar o processo de criação de testes bem como as mudanças de negócio.	D
4	Aumentar quantidade de testes funcionais no início do projeto.	Não, aumentar a quantidade de testes funcionais no início do desenvolvimento, como o teste de aceitação, não garante nada.	D
5	Priorizar os testes em módulos que já apresentaram falha na atual fase de desenvolvimento.	Não, priorizar os testes em módulo que apresentaram falha no pos-release.	D

Referências

FENTON, Norman E.. ; OHLSSON, Niclas. Quantitative analysis of faults and failures in a complex software system. **IEEE Transactions on Software engineering**, v. 26, n. 8, p. 797-814, 2000. <https://doi.org/10.1109/32.879815>

ZAIDMAN, Andy et al. Studying the co-evolution of production and test code in open source and industrial developer test processes through repository mining. **Empirical Software Engineering**, v. 16, n. 3, p. 325-364, 2011. <https://doi.org/10.1007/s10664-010-9143-7>