

QuittingTime

Desenvolver casos de uso desnecessários é um desperdício de tempo, recursos e pode ainda atrasar o projeto. Os requisitos devem ser especializados para reduzir o risco de falha do projeto e devemos seguir esses procedimentos rigorosamente para correr tudo como o cliente quer.

Os casos de uso para serem bons tem de ser equilibrados e sobretudo, devem ter o essencial para o projeto funcionar. Por vezes é complicado deixar de escrever casos de uso porque temos de alternar e não dificultar o processo porque por vezes temos de escolher entre uns casos e outros para não acrescentar demasiada informação, muita dela inútil. Devemos evitar que um cliente queira uma alteração após os casos de uso estarem bem definidos, daí a necessidade de alguma experiência e delinear bem o projeto para evitar ter que voltar atrás no projeto e perder tempo.

A experiência ajuda a que não aconteçam este tipo de erros. Os requisitos não devem ser específicos de mais, pois os mesmos podem comprometer os pré-requisitos. A deteção inicial de um erro num projeto é muito melhor pois evitamos ter que recomeçar o projeto e perder tempo e dinheiro com isso, pois é mau para o cliente e para o programador. Não devemos adicionar casos de uso a mais do que aqueles que já foram aprovados pelo cliente. Estas são as três perguntas para que um cliente faz para saber se o projeto cumpre os requisitos ou não:

1. Foram identificados e documentados todos os objetivos e todos os atores?
2. O cliente ou o seu representante reconheceu e aprovou todos os casos de uso antes o desenvolvimento do projeto?
3. O designer pode implementar os casos de uso?

A resposta a estas perguntas deve ser sim avançar com o projeto, caso contrário ainda existe trabalho a fazer.

Exemplo:

Wings Over The World (Asas sobre o mundo).

Os casos de uso não estão completos até estar tudo bem detalhado e discutido. Não se deve perder tempo e avançar com o projeto sem quaisquer preocupações.

“Quando é que se sabe quando os casos de uso estão completos?”

O QuittingTime equilibra o risco do atraso do projeto com o risco da falta de requerimentos.

É necessário ter sempre em conta sempre cuidado o que se faz e implementa. A comunicação entre o cliente e o programador deve ser bastante clara e esclarecedora para que não haja erros e para que não haja falsas interpretações, para o produto final corresponder aos requisitos do cliente.