

## Entrevista 03 - Entrevistado parte1+2

Atualmente, eu trabalho como engenheiro de DevOps, entre engenheiro de DevOps e analista de sistema. Mas, eu fico transicionando entre esses dois papéis. Pela experiência que eu tenho, sei lá, atualmente, cerca de quatro projetos, dois a quatro projetos.

Tem, tem sim. Todo processo, todo mundo que eu trabalho, sempre tem essa alocação de projeto primário e projeto de secundária. Tem um que é o principal, que é relacionado ao processo judicial e eletrônico, e o resto é assessorio, secundário em relação a isso.

Mas, veja só, no principal, diretamente dentro do órgão onde eu trabalho, existem quatro desenvolvedores. Agora, como tem uma empresa que presta serviço no desenvolvimento de features do principal, aí esse número, ele é volátil. Ele fica alternando, assim.

No momento tem, se eu bem me lembro, tem cinco desenvolvedores lá. Mas, dependendo das demandas que a empresa tem, esse número pode ser alterado, pode aumentar, pode diminuir. Não tem um controle.

Mas, é por volta desse contativo. Em total, uns dez desenvolvedores. Tipicamente, o processo de desenvolvimento em si, ele segue aquela mecânica tradicional de levantamento de requisito.

O analista, ele vai, ele faz o levantamento de requisito, ele prepara uma documentação. Essa documentação é passada para a equipe de negócio e aí é feito um processo de validação. Feito isso aqui, o documento, ele é repassado para o desenvolvedor, é desenvolvido.

Pós-desenvolvimento, o app é colocado num ambiente de testes ou de homologação. Pós-homologação e validação no cliente, é feito o rollout para a produção. Tipicamente assim que acontece.

A gente usa aquela estratégia do GitFlow, que é um branch por feature e aí a gente desenvolve num branch, num feature branch, depois faz o processo de merge request para homologação. Dando ok à homologação, é feita a promoção para a produção. Então, é esse esquema que a gente utiliza.

Assim, não no órgão que eu trabalho atualmente. Em outro órgão, antes de entrar, a gente tinha um desenvolvimento tranquilo. Não tinha, tipicamente não trabalhava com feature branch, então era tudo feito num branch só.

No caso, os fatores que influenciam o uso desse... Acho que o principal é a questão de você paralisar o trabalho em features diferentes. E aí você consegue uma certa independência para trabalhar cada um desses features. E aí não tem uma concorrência,

nenhum problema, principalmente no merge das features.

Apesar de a gente usar o Git e fazer esse desenvolvimento descentralizado, mas ainda assim, se estivesse trabalhando várias pessoas dentro do mesmo branch, a possibilidade e a chance de ter problema com merge seria muito maior do que essa outra estratégia do feature branch. O que nos favorece, acho que vai depender do... Se é uma equipe muito pequena e a gente não tem um paralelismo, esse overhead e essa complexidade de você ter vários branches acaba se tornando desnecessário. Mas, como eu falei, vai depender do tamanho da equipe e da complexidade do projeto.

Acho que esses são dois fatores que pesam bastante. Eita, parece que a internet caiu aqui. Opa, você me escuta? Alô? Alô? Atualmente, não.

Acho que o modelo que a gente está de funciona bem. Mas, além da questão orgânica e anímica de funcionar bem, outros órgãos, como o próprio órgão superior vinculado ao trabalho, também utiliza essa mesma estratégia. Então, também tem essa questão da compliance.

Você seguir um regamento. Pode repetir, por favor? Startup ou... Está mais para um órgão estruturado. Rapaz, a internet aqui está dando uma falha aqui.

Cortou a tua fala. Repete aí, por favor. Olha, a pioria é assim.

Talvez você argumentar sobre o tipo de versionamento descentralizado. O uso de versionamento descentralizado ou não. Embora que, hoje em dia, o uso principal é o versionamento descentralizado com NIT ou Mercurial, pode ser que tenha algum local que trabalhe com um versionamento centralizado.

Talvez você faça essa verificação sobre o tipo de versionamento que cada indivíduo pode utilizar. Acho que é só isso.