

Desafio teórico: Liderança e Trabalho em Equipe

Matheus Rodrigues Borges

I. Início e planejamento

O desenvolvimento de quaisquer *features* pode se beneficiar imensamente caso haja uma difusão de conhecimento acerca de seu propósito e de suas regras de negócio. Inclusive uma das abordagens mais bem vistas atualmente é o desenvolvimento guiado pelo DDD (*Domain Driven Design*). Uma de suas premissas fundamentais reside no profundo entendimento do domínio da aplicação, criando-se uma comunicação comum entre as equipes técnicas e o setor de negócio, a chamada Linguagem Ubíqua. Dessa forma, o primeira etapa deveria ser uma imersão da equipe de desenvolvimento junto de um representante do time de produto, a fim de se criar esse consenso. Em situações onde a participação completa da equipe não seja viável nessa imersão, cabe ao líder do time a responsabilidade de apresentar o domínio à equipe durante a fase de planejamento, de modo a construir essa noção de consenso.

Como segunda etapa temos o planejamento de fato das tarefas de desenvolvimento. Ela é extremamente crítica e requer a colaboração de todos os participantes da equipe, e é papel do líder técnico trazer discussões pertinentes, chamando atenção para os pontos de gargalo, tarefas mal quebradas e fornecendo esclarecimento acerca de problemáticas que podem impactar a entrega dos objetivos da *sprint*, afinal um planejamento assertivo gera um bem estar no time aumentando o moral dos colaboradores. Nesse momento, é comum que haja diversas discussões ante aos diferentes pontos de vista dos integrantes do time. Uma abordagem interessante nessas situações é o uso de um "token de fala" — apenas a pessoa com o token pode falar, passando-o adiante ao término de sua intervenção. Isso evita interrupções desnecessárias e promove um melhor entendimento das ideias transmitidas. No entanto, em certos momentos, as discussões podem deixar de ser produtivas. Nesses casos, cabe ao líder técnico intervir, destacando pontos fracos e válidos nos argumentos, a fim de apontar um caminho viável e satisfatório para a equipe.

Por fim, com todas as tarefas quebradas e pontuadas, tem-se uma visão clara das tarefas gargalo e/ou com um desafio técnico maior. É comum atribuir essas responsabilidades aos desenvolvedores mais experientes, porém, essa prática pode ser contraproducente, já que os desenvolvedores menos experientes podem ter pouca exposição a tarefas desafiadoras. Uma abordagem que acredito muito eficaz para mitigar isso é o *pair*

RadarFit - Teste técnico

programming entre um desenvolvedor mais sênior (ou até mesmo o líder técnico) com desenvolvedores mais juniores. Essa prática estimula uma troca constante de conhecimento, beneficiando principalmente os menos experientes, e os capacitando mais rapidamente para lidar com tarefas desafiadoras de forma independente, além de prepará-los para orientar outros desenvolvedores menos experientes. Vale ressaltar, entretanto, que é muito importante que a técnica de *pair programming* seja respeitada para que sua eficácia, comprovada pela literatura, seja alcançada. Dessa forma, uma das ações que geralmente faço é marcar quais tarefas possuem essa característica.

II. Rituais

No decorrer da *sprint* pode-se empregar alguns rituais que facilitam a disseminação do conhecimento e contribuem para o desenvolvimento dos desenvolvedores menos experientes. O primeiro ritual a se considerar é o momento de delivery/retro review. Nele costumo abrir espaço para o compartilhamento dos maiores desafios que foram enfrentados e, principalmente, como eles foram solucionados. Não é necessário entrar em detalhes excessivos, mas destacar esses desafios pode alertar os demais desenvolvedores. Caso se deparem com desafios semelhantes no futuro, saberão que esses desafios já foram enfrentados antes e quem foi envolvido na sua solução, possibilitando que busquem orientação.

Um segundo ritual complementar a esse são as reuniões de capítulo (de conhecimento), as quais permitem sim um aprofundamento de alguns dos temas e desafios levantados em alguma reunião anterior. Essas reuniões criam um ambiente colaborativo e de diálogo, no qual os participantes não apenas esclarecem os passos necessários para a resolução dos problemas, mas também podem identificar soluções ainda mais eficazes. Durante esses encontros, também reservo tempo para a elaboração, discussão e refinamento dos critérios que serão considerados na aprovação dos pull requests, promovendo assim um consenso entre os membros da equipe..

Por fim, para equipes que contam com desenvolvedores do perfil júnior ou aprendiz, realizo um acompanhamento diário, de cerca de 30 minutos. Durante essas sessões, revisamos as tarefas atribuídas a eles, analisamos suas abordagens de solução e oferecemos feedback construtivo. Esse acompanhamento visa proporcionar aos desenvolvedores juniores um ambiente mais seguro e propício ao seu desenvolvimento, aumentando sua confiança e habilidades.

RadarFit - Teste técnico

III. Conclusão

Neste documento, procurei descrever um pouco da rotina durante os ciclos de desenvolvimento, incluindo a quebra de tarefas, discussões e o processo de desenvolvimento dos membros da equipe ao longo da sprint. É importante destacar que outros processos paralelos devem ser implementados, especialmente o desenvolvimento de Planos de Desenvolvimento Individual (PDIs) com programas de capacitação alinhados a eles. Além disso, gostaria de ressaltar que não me limitei apenas à implementação da aplicação, que constitui a parte prática deste teste pois, devido ao tamanho reduzido da aplicação, nem todos os pontos que destaquei poderiam ser abordados em profundidade.