# Como engenheiros de software podem trabalhar melhor com times de produto

Por Escola Dnc

# Introdução

Times de produto e times de engenharia frequentemente têm dificuldades para trabalhar juntos de forma alinhada e eficiente. No entanto, uma boa colaboração entre essas equipes é essencial para o sucesso de qualquer empresa de tecnologia.

Neste ebook, exploraremos como engenheiros de software podem melhorar sua relação com times de produto, garantindo maior produtividade, qualidade e velocidade no desenvolvimento de produtos digitais.

Abordaremos os seguintes tópicos:

- O papel da engenharia na inovação de produtos
- Como garantir eficiência e escalabilidade
- Mantendo a tecnologia sempre atualizada
- Boas práticas para colaboração com times de produto
- Comunicação transparente e feedbacks construtivos

Ao final deste material, engenheiros de software terão uma visão muito mais clara de como podem apoiar e habilitar seus colegas de produto, levando a melhores resultados para os usuários e para o negócio.

Vamos começar!

# O papel da engenharia na inovação de produtos

Embora a maioria das ideias e requisitos venham dos times de produto, engenheiros têm um papel fundamental a desempenhar na inovação e evolução de produtos digitais.

Há duas grandes formas pelas quais engenheiros podem apoiar a área de produto e tecnologia como um todo:

### 1. Garantindo eficiência, organização e escalabilidade

Engenheiros precisam manter códigos e arquiteturas que possam ser facilmente entendidos e modificados por diferentes squads de desenvolvimento. Isso inclui ter uma boa estrutura, padrões de código claros e canais eficientes de comunicação.

À medida que a empresa cresce e mais desenvolvedores são contratados, um software bem organizado e escalável se torna fundamental. Engenheiros devem sempre pensar à frente em termos de adaptabilidade.

### 2. Impulsionando inovação tecnológica contínua

Além de manter sistemas eficientes, engenheiros também podem trazer inovações tecnológicas frequentes, como:

- Atualizações em bibliotecas e frameworks
- Melhorias e refatorações em códigos legados
- Novas técnicas de IA, machine learning, etc.

Essa mentalidade de sempre buscar melhorias técnicas permite antecipar demandas futuras e trazer mais recursos ao produto e ao usuário final.

Engenheiros não devem ter medo de apresentar suas ideias para produto e participar ativamente das discussões de roadmap. Quanto mais informações técnicas forem incorporadas desde o início, mais chance de construir produtos realmente revolucionários.

## Garantindo eficiência e escalabilidade

Conforme mencionamos, um dos principais papéis da engenharia é garantir que o software sendo desenvolvido tenha uma boa estrutura, seja fácil de manter e possa ser escalado conforme a empresa cresce.

Vejamos algumas práticas que ajudam a alcançar esse objetivo:

### Padrões de código e arquitetura

Ter padrões bem definidos de como o código deve ser escrito, formatado e organizado (arquitetura) é essencial. Isso facilita muito a vida de novos desenvolvedores e permite que squads diferentes trabalhem de forma alinhada.

Algumas coisas importantes neste contexto:

- Estilo e formatação de código
- Nomenclatura de variáveis, funções e classes
- Separação de responsabilidades (SRP)
- Modularização e baixo acoplamento

### **Testes automatizados**

Ter uma boa cobertura de testes (unitários e de integração) é fundamental para permitir refatorações e evolução contínua do código com segurança.

Testes também facilitam a colaboração entre squads diferentes, já que servem como "documentação executável" sobre como cada parte do sistema funciona.

### Integração e entrega contínua

Automatizar o máximo possível do processo de build, teste e deploy ajuda muito na eficiência das equipes.

Práticas como CI/CD (integração e entrega contínua) são essenciais para viabilizar entregas frequentes e diminuir riscos durante atualizações.

### Monitoramento e observabilidade

Ter métricas e logs bem estruturados facilita o trabalho de times de suporte e operações. Permite também tomar decisões de arquitetura e performance com mais dados e confiança.

Logs e métricas ajudam times de produto a entender como os usuários interagem com os produtos e onde melhorar.

# Mantendo a tecnologia sempre atualizada

Outro papel fundamental da engenharia é garantir que as tecnologias utilizadas estejam sempre modernas e atualizadas. Isso traz diversos benefícios:

### Melhoria contínua

Bibliotecas e frameworks são constantemente atualizados pela comunidade open source. Mantê-los updated permite acesso contínuo a melhorias de performance, segurança e novas funcionalidades.

### Aumento de produtividade

Novas versões de linguagens de programação e ferramentas trazem recursos que aumentam a produtividade dos engenheiros, permitindo entregar mais valor em menos tempo.

### Redução de bugs e incidents

Issues, vulnerabilidades e bugs são corrigidos em novas versões de bibliotecas e ferramentas. Estar sempre atualizado reduz chances de incidents em produção.

### Acompanhar tendências da indústria

Estar por dentro das últimas tecnologias permite avaliar e incorporar inovações como Inteligência Artificial, Cloud e outras que podem trazer vantagens competitivas ao produto.

### Como manter a tech atualizada?

- Criar um roadmap de atualização de tecnologias
- Separar um tempo exclusivo para atualizações e refatorações

- Automatizar checks de versões desatualizadas
- Fazer spike de tecnologias promissoras

# Boas práticas para colaboração

Além dos aspectos técnicos, alguns princípios e atitudes ajudam times de engenharia a terem uma relação mais próxima e produtiva com produto:

### Colaboração multidisciplinar

Engenheiros devem sempre buscar proximidade com designers e PMs para entender as necessidades reais dos usuários e como sua solução se encaixa no produto como um todo.

### **Entregas incrementais**

Apresentar versões parciais e incrementais do seu trabalho para os PMs com frequência traz diversos benefícios:

- Permite feedbacks mais rápidos
- Diminui riscos de idas em direções erradas
- Dá visibilidade sobre o progresso para os não-técnicos

### Testes completos antes de reviews

Time de produto não deve ser usado para encontrar bugs básicos. Faça testes completos (unidade, integração, manual) antes de submeter para revisão.

### Gestão de dependências

Mantenha todas as bibliotecas e dependências do projeto sempre atualizadas para evitar bugs e problemas de segurança.

### Métricas de engenharia

Acompanhar métricas como performance, estabilidade e segurança traz insights para melhoria contínua e facilita tomadas de decisões de produto e negócio.

### Documentação e versionamento

Ter uma documentação técnica consistente e um processo estruturado de controle de versão (git) é fundamental para organização e colaboração.

# Comunicação transparente e feedbacks construtivos

Por fim, mais do que processos ou ferramentas, o que realmente une times de produto e engenharia é uma comunicação transparente, honesta e empática.

Ambos os times devem sempre partir do princípio que todos querem entregar o melhor produto possível para os usuários e para empresa.

### Comunicar claramente problemas

Engenheiros não devem ter receio de apontar bugs, riscos técnicos, débitos de arquitetura ou outros problemas para produto. O importante é transmitir de forma profissional e construtiva.

### Receber feedbacks de forma positiva

Da mesma forma, críticas e sugestões de melhoria vindas de PMs não devem ser vistas como ataques pessoais, e sim como oportunidades de evoluir e crescer profissionalmente.

### Foco no usuário e no negócio

Em caso de divergências ou desentendimentos, o time deve sempre voltar suas atenções ao impacto que determinada decisão trará para o usuário e sucesso do negócio. O resto é secundário.

Com empatia, paciência e foco no cliente, times de produto e engenharia podem superar desafios e criar produtos realmente revolucionários e de sucesso.

Não se esqueça dos princípios apresentados neste ebook e boa sorte!