

eBook

05 DICAS

para contratar software



Você acabou de ter uma ideia genial para um aplicativo, ou precisa resolver algum problema importante no seu processo de negócio? Percebe que um pouco de tecnologia iria trazer grandes benefícios pra você, mas não sabe por onde começar?

comece por aqui!

Trouxemos aqui **5 dicas** para contratar desenvolvimento de soluções em software. Nesse e-book procuramos responder algumas das questões mais comuns que rodeiam a mente das pessoas que nos procuram para ajudá-las no desenvolvimento de produtos.

Trazendo as mais modernas técnicas, métodos e conceitos dessa indústria para ajudar que o seu projeto seja um **sucesso!**

sumário

01 como selecionar fornecedores?	4
Indicação	4
Reputação	4
Rotatividade	5
02 como definir escopo de projeto?	6
Entendendo o problema	6
Desenvolvendo a solução	8
Mas o que devo pedir em contrato?	10
03 quanto vais custar?	11
Já tenho um orçamento previsto	12
Não tenho um orçamento previsto	13
Muito importante!	14
Mais considerações sobre contratos	15

04 como saber se há qualidade?	16
O que são testes automatizados?	17
Outras questões a serem observadas	18
05 quanto tempo vai levar?	20
Iterações	21
06 bônus! E se eu for do governo?	22
conclusão	23
agradecimentos	24

01 como selecionar fornecedores?

Indicação

Você se lembra da última vez que precisou de um mecânico, médico ou qualquer outro profissional? Você achou seu mecânico na base da intuição? Ou encontrou seu médico sorteando na lista telefônica ou no Google? Ou pediu **indicação** a alguém de confiança? Com **software** não é diferente!

Se você não é da área de TI, provavelmente há muitas questões técnicas e metodológicas que fugirão do seu domínio. Você precisará de uma equipe em que possa confiar. Se alguém te indicar uma empresa, provavelmente será porque ela já usou seus serviços e gostou da forma com que o projeto foi conduzido. Ou então porque já ouviu falar bem da empresa através de outro conhecido. **Mas, não pare por aí.**

Reputação

Verifique também a reputação dessa empresa. Verifique se seus casos são relacionados ao projeto que você quer desenvolver. Procure os clientes, fale com eles, pergunte sobre a satisfação com o serviço prestado.

OK, até aqui não trouxe nada de novo sobre seleção de fornecedores, certo? Mas a próxima dica é, no mínimo, curiosa.

Rotatividade

Procure saber como é a **rotatividade** dos funcionários dessa empresa! Pode parecer contraintuitivo observar essa questão, mas se a rotatividade da equipe é alta demais, os serviços já prestados não serão indicação alguma para próximos serviços. Você poderá lidar com uma equipe totalmente nova.

Outro fator importante é que soluções de software ainda são muito complexas. Sua qualidade vai depender muito das pessoas envolvidas no seu desenvolvimento. A equipe precisa ter qualidade e conhecimento técnico, mas fatores como motivação, confiança, engajamento e comprometimento terão impactos diretos no resultado do projeto.

Você vai querer uma equipe que tenha **paixão** pelo que faz, assim como você, para que os resultados do trabalho sejam excepcionais. Se a relação entre a equipe e a empresa for apenas “funcional”, como se fosse “apenas mais um emprego”, é provável que os resultados do projeto sejam medíocres.

Nem toda empresa permite acesso do contratante à equipe, mas se possível, procure conhecer os empregados e peça pra falar dos projetos, dos colegas e da empresa. Tente observar se eles têm “brilho nos olhos” ao falar desses assuntos. Essa percepção é valiosa para tomar sua decisão.



02 como definir escopo do projeto?

Entendendo o problema

Imagine a seguinte situação: você sente uma dor imensa, pesquisa os sintomas no Google e descobre que precisa tomar uma injeção. Compra a injeção e vai ao médico, só pra ele aplicá-la em você, porque você mesmo não sabe aplicar.

Muita gente tem essa atitude hoje em dia. Com médicos e com relação a software também! Já chegam nas empresas com as soluções pensadas, e esperam que elas sejam apenas construídas.

Tudo bem se você acredita muito na sua ideia e nas suas soluções. Muitas soluções de sucesso foram desenvolvidas dessa forma. Mas se você soubesse o número delas que foram desenvolvidas assim e não deram certo, provavelmente procuraria uma abordagem mais eficaz.

Por exemplo, um **relatório de 2013** cita que 67% das aplicações para dispositivos móveis disponibilizados estão “abaixo da linha de pobreza para apps”, ou seja, faturam menos de U\$ 500 por mês. Um fracasso desses pode sair bem caro.

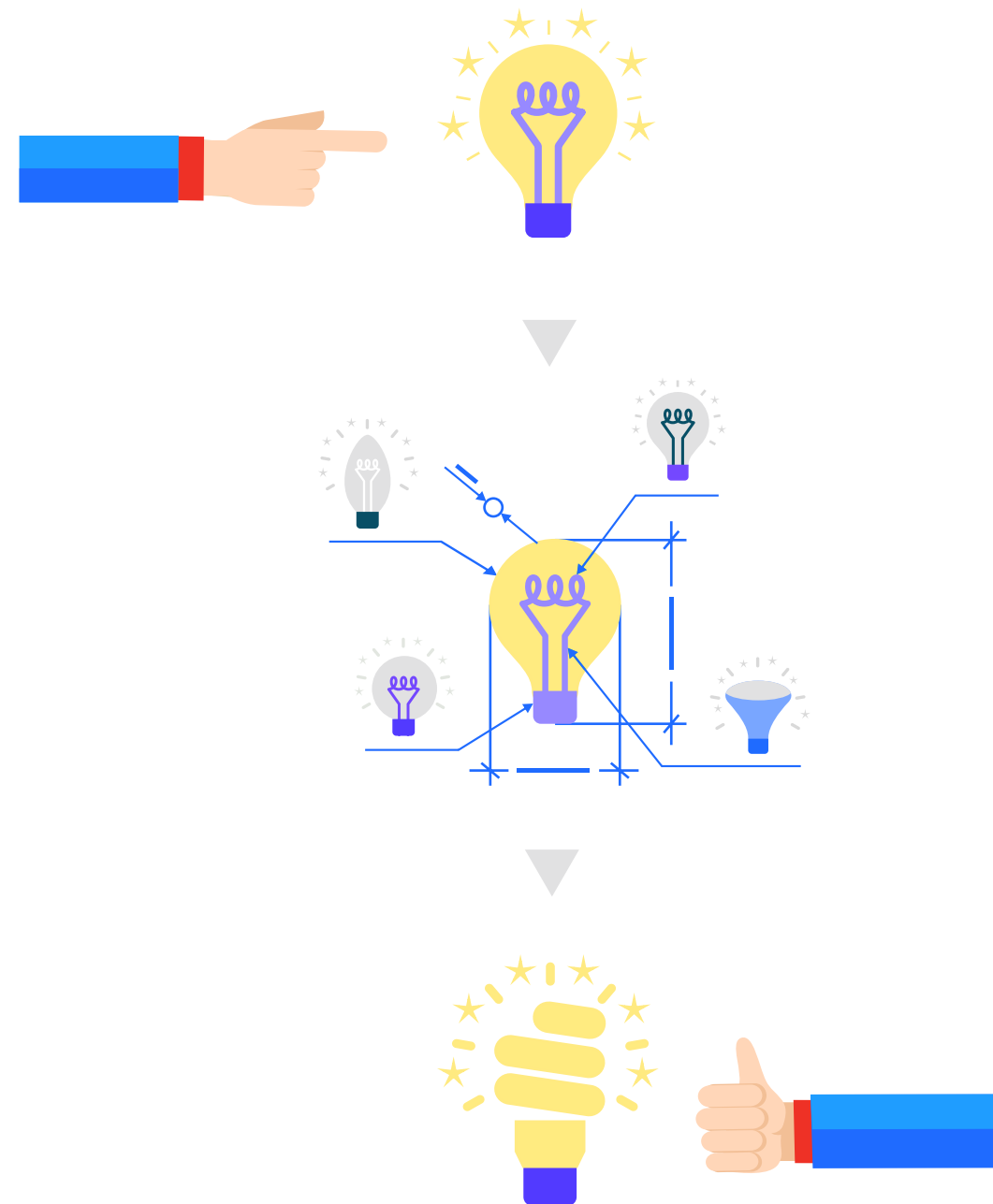
Que tal permitir a criação de um laço de confiança com o médico, que tem mais experiência no assunto e tem trabalhado com isso há anos? E se você tivesse um corpo de médicos capacitados, cada um na sua especialidade, com experiências bem diversas, pra discutirem seus sintomas e prescreverem um remédio? Melhor ainda, não?

02

Aí está como acreditamos que você poderia encarar o time de desenvolvimento de software: você é o paciente, que entende muito bem das dores ou do **problema** a ser resolvido — da **oportunidade** a ser aproveitada. E o time é uma equipe de médicos, que tem diversos conhecimentos, especialidades, métodos e técnicas para lidar com sua dor — ou aproveitar a oportunidade.

Na verdade, bem mais que um time de desenvolvimento de software, seria muito bom dispor de um time de desenvolvimento de **produto**, que envolve conhecimentos além daqueles técnicos e perfis mais peculiares.

Uma equipe assim trará toda a sua experiência, vivência, ferramentas e métodos para cocriar a solução. Por isso uma equipe **multidisciplinar** — com uma diversidade de perfis — é importante para enriquecer as propostas. Ela não se preocupa apenas em fazer o que for pedido, mas tenta realmente se colocar na pele do idealizador e dos usuários para ter as melhores ideias possíveis para resolver o **problema**.



Desenvolvendo a solução

Você entendendo bem de uma parte, e o restante da equipe entendendo bem de diversas outras partes, trabalharão em conjunto para conceberem as melhores soluções.

Só que tem um detalhe: estamos num mundo que muda muito rápido e reage a cada mudança no seu ambiente. Seria como se cada doença encontrada fosse nova, única, e que cada remédio aplicado mudaria essa doença de novo.

Isso quer dizer que dificilmente haverá solução correta proposta “de cara” pra resolver um problema. Por isso, pensaremos em cada proposta pra resolver o problema como uma hipótese, que será testada pra ver se estamos no caminho certo. Mediremos a efetividade de nossa solução, aprenderemos com esses experimentos e evoluiremos rapidamente a partir daí. Ou seja, usaremos uma abordagem **empírica**.

Mais que isso, como a solução é descoberta aos poucos, há reorientamento constante do projeto. Como se partíssemos do princípio que o problema era no fígado, mas vamos descobrindo que

era no estômago, e a partir daí, vamos entender melhor em que parte do estômago. Por isso, também dizemos que a solução é **evolutiva ou incremental**.

Observe que não estamos num mundo em que o importante é fazer “mais coisas” em um projeto, e sim que é importante **“descobrir as coisas certas a serem feitas”**, e então fazê-las rapidamente. Uma boa empresa fará isso iterativamente, ou seja, em vários ciclos curtos, pra aprender rapidamente; e incrementalmente - acrescentando funcionalidade aos poucos. Assim, o contratante não gasta dinheiro nem tempo demais com o que não deve.

Se você fizer uma pesquisa, notará que os aplicativos de sucesso hoje resolvem muito bem um problema, por exemplo: resolvem bem seu problema de transporte (Uber/99Taxi), de trânsito (Waze), comunicação (WhatsApp). É isso que você deve ter em mente ao contratar seu projeto: **resolver muito bem o problema**. A evolução só virá depois de um primeiro sucesso.

02

Some a tudo isso mais essa informação: há pesquisas que indicam que a maioria de todas as funcionalidades produzidas nos projetos são **nunca ou raramente utilizadas** (veja o quanto usa do seu Word ou Excel na média). Ou seja, são desperdício. **Dinheiro jogado fora**. E ninguém tem dinheiro sobrando, né?

Por isso as metodologias de maior sucesso hoje recebem o nome de **Lean** ou enxuto. Visam **evitar** ao máximo o **desperdício**. Fazer mais do que se deve, tomar decisões antes de ter boas informações/aprendizados ou produzir defeitos são formas de desperdício.

Por isso que sua preocupação maior não deve ser com o **escopo** ou a solução que será desenvolvida, mas com o **problema** a ser resolvido. Se há um negócio por trás dela, se há alternativas melhores, se há mais pessoas com esse problema, se alguém pagaria pela sua solução, se o nicho é suficientemente grande, se seu desenvolvimento é sustentável, se é fácil copiar sua solução e assim por diante.

LIVROS

"The Lean Startup" (Eric Ries),

"Business Model Generation" (Osterwalder, Pigneur),

"Lean UX" (Jeff Gothelf),

"Running Lean" (Ash Maurya).



THE LEAN STARTUP

02

Mas o que devo pedir em contrato?

Você deve se preocupar em ter o **código** em primeiro lugar. Ter um repositório, mesmo que na nuvem, e ter acesso a ele, e exigir que as últimas versões do código estejam atualizadas nesse repositório.

É importante que o código seja bem feito e inteligível por outros desenvolvedores. Isso será feito por uma boa equipe, com essa preocupação em mente. Para selecionar uma boa equipe, voltamos aos critérios da primeira dica, sobre seleção de fornecedores.

O código também deverá ter “**qualidade embutida**”. Complementarei essa questão da qualidade em outra dica posterior, onde falarei de testes automatizados. Bem como scripts e outras automações e tudo mais o que for produzido para sua solução. No mais, a resposta será vaga: documente o **necessário**.

Como é difícil saber qual documentação será necessária de antemão, procure por contratos que sejam flexíveis quanto a deman-

da por eventuais documentos ainda que no meio da execução do mesmo. Por exemplo, se no meio do projeto você identificar que precisará de “modelos arquiteturais”, você poderá incluir isso como um critério para que a demanda esteja pronta e o pagamento do serviço possa ser feito.

Os riscos que devem fazer você se atentar mais à documentação são:

01

**crescimento
da solução**

02

**crescimento
da equipe**

03

**alta rotatividade
da equipe**

04

**interrupção
do projeto**

Portanto, preocupe-se mais com o problema, o negócio (e seu valor), qualidade, prazo e orçamento. Trataremos justamente desses pontos nas próximas dicas.

03

quanto vai custar?

Pra saber quanto vai custar o projeto, temos duas abordagens: ou você já tem um orçamento previsto, ou quer uma estimativa para desenvolvimento de uma ideia que já tem em mente.



03

Já tenho um orçamento previsto

Se você já tem um orçamento previsto, a empresa deve te responder o que acha¹ que consegue fazer dentro desse orçamento, respeitando suas prioridades, para que tenha o maior **retorno** possível sobre o seu **investimento**.

A empresa deve levantar os fatores que impactam os custos de um projeto, como integração com legados, automatização de rotinas, uso de perfis de profissionais mais específicos para fazer sua estimativa. Assim, você terá uma ideia de prazo e trabalho que será entregue pela equipe.

Pesquise outras empresas! Veja o que estimam entregar pelo valor investido. Nesse momento, observe como eles se planejam para entregar as funcionalidades. Se usam alguma métrica padrão para dimensionar a solução (será que essa métrica realmente está levando todos os fatores em consideração?) ou se estão realmente pensando no trabalho envolvido para entregar as funcionalidades o quanto antes para gerar valor e aprendizado pra você.

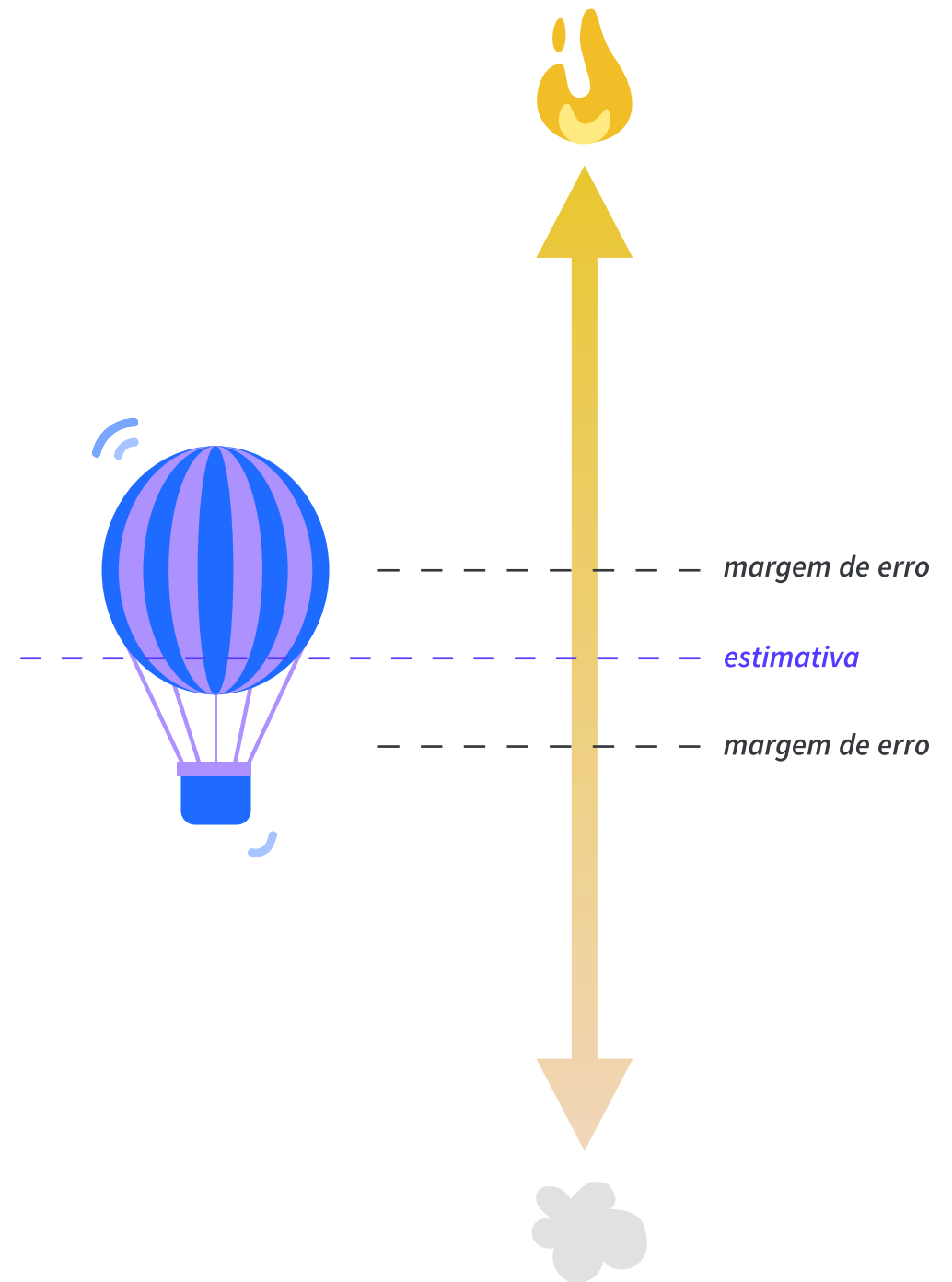
¹ Acha porque haverá mudanças de caminho ao longo do percurso. Haverá aprendizados e mudanças de ideia, mas o princípio de devolver funcionalidades de maior valor possível o quanto antes deverá continuar sendo respeitado.

03

Não tenho um orçamento previsto

Nesse caso, através de suas informações e do entendimento da equipe no momento de um levantamento, é possível fazer uma **estimativa grosseira** do custo do projeto. Expliquemos melhor o que essas duas palavras querem dizer nesse contexto:

- **Estimativa:** não é previsão do futuro! Não quer dizer que aquele valor tem precisão. Aliás, num primeiro momento, ele pode ter variações enormes do valor inicialmente pensado. Ela trará uma margem de erro e uma noção de ordem de grandeza do projeto. Por exemplo, estamos falando de algo que possa custar por volta de R\$ 50.000, R\$ 500.000 ou R\$ 1.000.000?
- **Grosseira:** porque é uma etapa muito inicial do projeto. Não se sabe como será a dinâmica com a empresa ou sobre a produtividade da equipe. Muita coisa ainda vai mudar quando o projeto começar. Quanto maior o projeto nesse momento, maior a margem de erro. Estimam-se melhor coisas menores do que maiores.



03

Lembre-se também de que o que será feito deve mudar ao longo do projeto. Novas ideias e aprendizados podem surgir a partir de cada entrega, então o que estamos fazendo aqui é refletir sobre um valor que seja suficiente para realizar experimentos e começar a fazer o projeto funcionar, pra então, com mais informações e conhecendo melhor o projeto, seus usuários e outros fatores, refazer estimativas e uma melhor ideia de custos ser projetada. A estimativa vai sendo refinada à medida em que há mais aprendizado e ela é refeita algumas vezes até estabilizar em alguma valoração de maior confiança.

Muito importante!

Em qualquer ocasião, sempre procure por contratos que permitam reavaliação do andamento pelas partes envolvidas em momentos determinados, e possa ser cancelado sem ônus. Você deve poder parar o projeto, caso não se sinta satisfeito com os serviços prestados, por qualquer razão que achar pertinente.

Por exemplo, a cada três meses, tanto fornecedor quanto cliente poderão solicitar o cancelamento do contrato, sem multas ou quaisquer ônus. Em caso de não manifestação, o contrato é renovado por período igual.

[Veja um exemplo desse contrato e mais alguns textos ricos sobre este assunto aqui:](#)



CONTRATO DE ESCOPO NEGOCIÁVEL



CONTRATO ABERTO OU MENOS DETALHADO

03

Mais considerações sobre contratos

No mais, procure um formato de contrato que seja claro e simples quanto à forma que irá cobrar pelos serviços. Quantas pessoas envolvidas, quantas horas previstas, valor de hora, perfis, etc.

Existem contratos em que o prazo é muito importante, então o valor do serviço varia de acordo com isso, por exemplo, se a entrega for feita no prazo, a hora sairá por R\$ 100,00. Se atrasar, R\$ 90,00. Se for entregue antes, R\$ 120,00.

Você pode ter um contrato grande e genérico, mas acertar maiores detalhes em entregas menores como, por exemplo, o uso de ordens de serviço, que permitirão ajustes quanto a valores, prazos, qualidade e entregas que devem ser feitas. Essas ordens de serviço também podem ter escopo melhor definido, já que o contrato terá um escopo mais genérico.

De qualquer forma, entenda que **você** como demandante **faz parte desse processo** de criação, e **deve estar presente** no projeto o máximo possível para que ele seja bem sucedido.

Contratar software não é como comprar um carro, em que você escolhe marca, cor, modelo, potência, acessórios e espera ele chegar pronto conforme sua encomenda. Está mais para fazer um carro sob medida pra você. **O seu nível de participação é fator crítico para o sucesso de um projeto de software!**

Assista no YouTube qualquer episódio do programa "Overhaulin" e observe o nível de interação entre a equipe e o dono do carro, e a importância do seu papel no projeto.



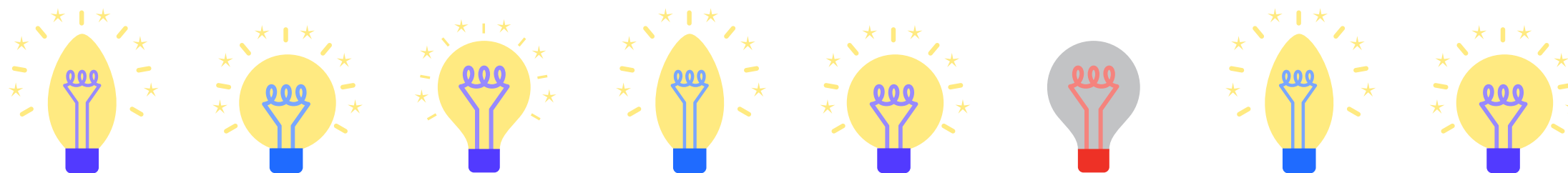
OVERHAULIN: OS SUPER CARROS

04

como saber se há qualidade?

Essa pergunta é de extrema importância, mas o que tem de importância, tem de dificuldade em ser respondida. A própria Academia tem muitas pesquisas pra tentar encontrar respostas nesse sentido, mas não há nada definitivo.

O que podemos afirmar com muita segurança é que sua solução, dentre outras coisas, deve ter uma "boa" cobertura de testes automatizados.



O que são testes automatizados?

Explicando de forma simples, são códigos desenvolvidos pelos desenvolvedores que verificam o comportamento do próprio código e sinalizam se o comportamento está de acordo com o que deveria ou não.

Por exemplo: um usuário quer comprar um livro numa e-livraria, e durante o processo de compra, o campo CPF precisa ser validado conforme um algoritmo para seguir o fluxo. Um teste automático simula a entrada de alguns CPF's válidos e verifica se o fluxo segue, e inclui alguns não válidos e verifica se o sistema retorna algum tipo de erro. Simples assim.

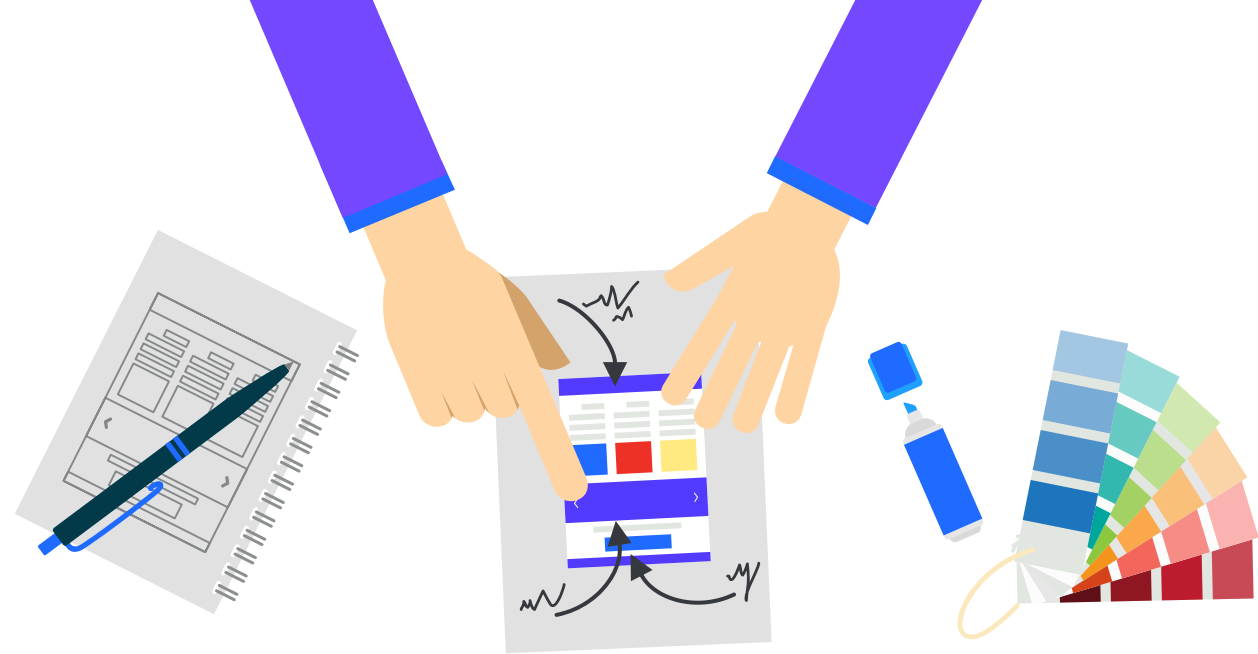
Apesar da simplicidade do exemplo, temos que considerar a complexidade do trabalho com testes automáticos. Saber o que

testar ainda depende muito de experiência e domínio das pessoas desenvolvendo o projeto. Testar absolutamente tudo não implica **necessariamente** numa boa qualidade de software. Agora, não testar deixa o código muito vulnerável.

São precisos bons testes - testes inteligentes, que verificam as funcionalidades e mantêm o sistema em evolução sustentável.

Você deve cobrar esse tipo de testes nos seus contratos.

Eles não implicam que testes manuais não devam ser feitos. Principalmente porque há questões visuais que não são captadas por testes automatizados. **Um olho humano testando o sistema tem valor!** Se for o seu ou de alguém de sua confiança, melhor!



Outras questões a serem observadas

Ultimamente tem se levado muito em consideração questões relativas à experiência do usuário com o produto. Nos referimos a essa disciplina como UX, de User Experience. São aspectos relacionados a todos os sentimentos do usuário no contato com a empresa e com o produto.

As interações devem ser projetadas para serem agradáveis, fáceis e úteis. Um exemplo de empresa referência nesse assunto é a Apple: todo contato com seus produtos ou com a empresa é pensado pra trazer sentimentos agradáveis. Você pode observar pelo projeto de suas lojas, que são espaçadas, a caixa e a experiência de abrir a caixa de um de seus produtos, quando liga um dos seus produtos pela primeira vez, a facilidade de seu uso, etc. Por isso, o emprego de um **profissional de UX** pode ser um fator de grande

valor para seu projeto (lembra que foi mencionado o valor de um time multidisciplinar?).

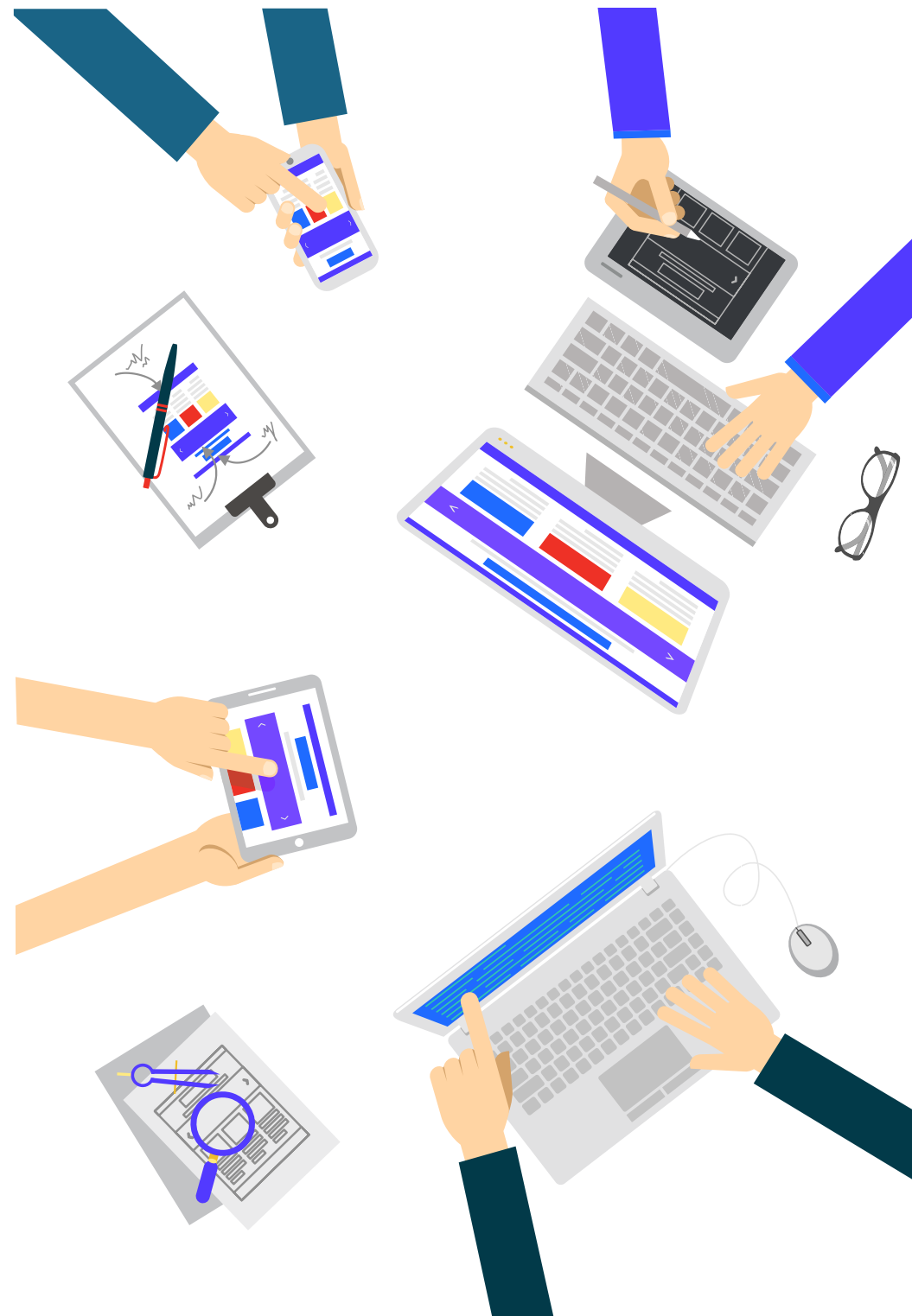
Há padrões de codificação, de nomenclatura, bons padrões arquiteturais, padrões de performance, tecnologias recomendadas que também podem ser aferidos. Inclusive essa aferição pode ser automatizada por ferramentas, como o "Sonar".

Isso também influenciará para uma boa qualidade técnica do código. Se você tiver algum desenvolvedor de confiança com domínio desse assunto para estabelecer esses critérios técnicos para a entrega, isso seria muito bom para seu contrato. Melhor ainda se ele puder colaborar com a equipe, contribuindo com seu conhecimento e experiência.

04

Outra prática comum de mercado é contratar **empresas especializadas em testes**. O cuidado é que essas empresas podem demandar alguns artefatos para conseguirem realizar seus testes e acabar encarecendo o projeto. Para evitar isso, seria bom que profissionais da empresa de testes trabalhassem em equipe com a equipe de desenvolvimento, visando **qualidade do projeto**.

Lembre-se de que a **qualidade do projeto** está relacionada a resolver bem os problemas dos usuários do sistema. Verificar o uso e abrir canais de avaliação por parte deles é uma boa prática para melhoria e avaliação da qualidade do projeto. Podem ser feitas pesquisas com usuários, avaliações da experiência, tempo de uso e operações podem ser mensurados pra inferir se há dificuldades ou onde o projeto pode ser melhorado.



05

quanto tempo vai levar?

As estimativas de tempo vão funcionar como as estimativas de custo, como explicamos acima: ou você já tem um prazo e a equipe avalia quanto trabalho consegue fazer pra entregar naquele tempo, ou são feitas estimativas de prazo e estas são refinadas com o desenvolvimento do projeto. **Em software, os projetos acabam conforme acordos e prioridades**, pois podem evoluir indefinidamente.

Imagine a construção de uma residência: mesmo depois de "pronta", sempre haverá uma nova ideia a ser construída, um reparo, uma reforma, algo a ser mudado... O que dá o limite de um projeto são prioridades feitas de acordo com o momento de vida. Pode não haver dinheiro num momento, algo pode ser adiado em prol de uma troca de carro ou viagem etc.

Observe as soluções que você usa atualmente e você verá que, ou novas versões são lançadas com alguma frequência, ou elas estão sendo descontinuadas para dar lugar a alguma solução diferente. Existem diversas técnicas para visualizar e priorizar a solução, de forma que, qualquer que seja o tempo do projeto, o que tiver maior valor será entregue.

Pergunte para a empresa ou equipe quais técnicas são usadas para priorizar funcionalidades. Tente entender o domínio que tem em desenvolvimento de projetos. Qual entendimento que tem sobre prazo, escopo, qualidade etc.

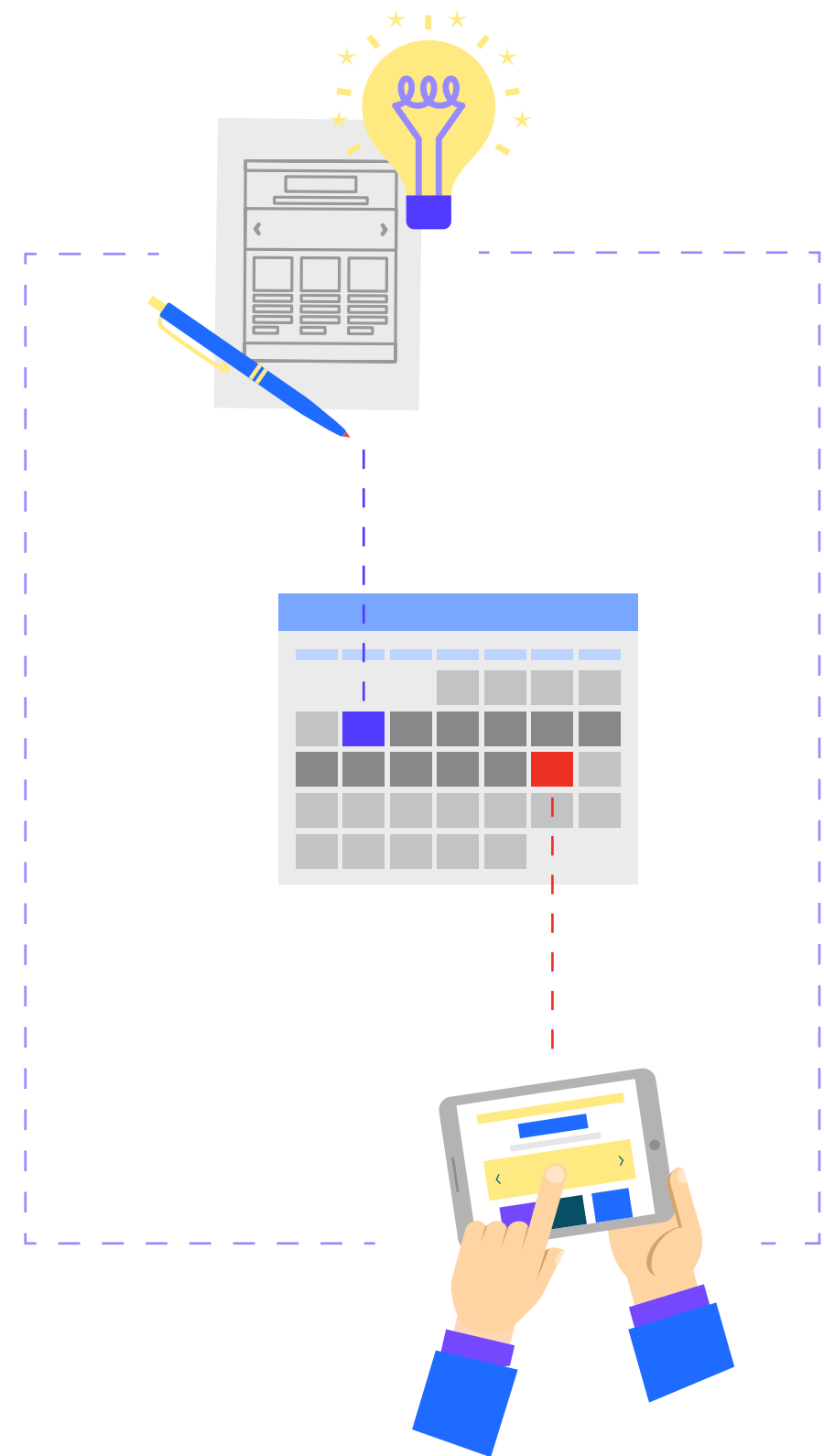
Peça para contar experiências nesse assunto. Assim, poderá avaliar a maturidade da empresa e o entendimento sobre projetos dessa natureza.

05

Iterações

O mais importante sobre prazo é que entregas sejam feitas iterativamente! Ou seja, de tempos em tempos, conforme combinado, uma funcionalidade ou um pacote de funcionalidades deve ser entregue e de preferência colocado no ar. Isso em intervalos **curtos** de tempo! Se uma entrega for levar mais de um mês, você já está assumindo um alto risco no projeto.

Lembre-se de que estamos falando de um projeto de aprendizado. Para que todos possam aprender, é preciso que funcionalidades fiquem prontas, sejam colocadas no ar, sejam verificadas em ambientes seguros e recebam feedbacks de usuários. Quanto mais próximas da realidade forem as experiências, melhor.



06 bônus!

E se eu for do governo?

Se você for um servidor público procurando por dicas de contratação de software, saiba que há um grupo de pessoas realizando algumas experiências de sucesso e as relatando no endereço digital do **Governo Ágil**. Pode procurar qualquer autor do site para discutir ideias que colaborarão com prazer.

Há também uma lista de discussão no Google:

✉ CONTRATACAO-PUBLICA-TI@GOOGLEGROUPS.COM.BR



conclusão

No fundo, o que você quer é previsibilidade, certo? É saber quanto vai custar, quanto tempo vai durar e o que terá no final. E não é o que todos queremos?

O que tentamos apresentar aqui é que a natureza de um projeto de software é diferente. É um processo de aprendizado e descoberta orientado a valor e a solução de problemas. Um processo que trará ganhos inerentes e intangíveis ao negócio e a você.

Encarando-o de forma mais adequada, você saberá onde está pisando, terá sua expectativa ajustada e suas chances de sucesso se multiplicarão.

Nós da **Novatics** desejamos sucesso nas suas empreitadas! Somos uma equipe de produtos capaz de compreender a fundo o que você realmente precisa. Somos orientados a valor e estamos dispostos a ajudá-lo a desenvolver o produto mais adequado para resolver seu problema.

Conheça alguns clientes que confiam no nosso trabalho:



Veja todos os clientes que já trabalhamos e nossos projetos no nosso site!



Agradecimentos

Agradecemos a todos que contribuíram para o desenvolvimento deste material:

Renato Willi
Flávio Alves
Eraldo dos Santos
Diogo Riker
Jana Simões
Luiz Eduardo de Araujo
Gustavo H. Maultasch de Oliveira
Nanci Hokino
Raianne Rodrigues
Beatriz Bukvic