



Talita Pagani

[Follow](#)

Mestre em Ciência da Computação | Consultora/Tester de Acessibilidade na Utilizza |  
Designer na Nexaas

Jan 5 · 9 min read

## Como estimar o esforço de desenvolvimento de software e por que isso é tão difícil? — Parte 1

Chega uma nova demanda para desenvolvimento de um projeto ou funcionalidade de um produto e a pessoa responsável pela gestão do projeto/produto chega pra você e pergunta na reunião: *“e aí, quanto tempo você leva para fazer isso?”*.

Por mais que tenhamos experiência em desenvolvimento, estimar tempo e prazo é uma tarefa desafiadora. O que mais ouço de colegas com relação ao tema é *“nossa, é muito difícil falar quanto tempo vou levar pra desenvolver algo, como você consegue fazer isso?”*.



Este tweet do Marcelo me representa. Descrição: contém um GIF de uma menina pequena com as mãos levantadas e balançando a cabeça com a boca aberta de forma confusa.

É difícil, mas não impossível. Em quase 13 anos atuando em TI, percebi vários fatores que impedem desenvolvedores de conseguir estimar software de forma adequada:

- Requisito incompleto e muito subjetivo;
- Falta de conhecimento de desenvolvedores sobre métricas de software para aplicar às suas próprias tarefas. Nesse ponto, recomendo os livros “Indicadores De Gerenciamento De Projetos” do Armando Terribili Filho e “Métricas Ágeis” do Raphael Albino;
- Gerentes de projeto querendo que usemos uma bola de cristal para tirar uma estimativa do nada sem dar tempo para fazermos uma análise dos requisitos;
- Funcionalidade, projeto ou produto muito distinto do que já foi desenvolvido anteriormente, o que impede uma estimativa por analogia.

Estimativas incorretas podem comprometer as entregas de diversas formas: acabam gerando um cronograma impreciso (muito apertado ou

com muita folga que deixa a equipe ociosa), pode gerar mais custos para o projeto, horas extras para entregar no prazo acordado com o cliente, entre outros.

## Tempo x Prazo

Podem parecer sinônimos, mas tempo e prazo de realização para uma atividade são conceitos distintos, embora complementares. Às vezes, até quem é responsável pela gestão de um projeto se confunde nesses conceitos no momento de solicitar uma estimativa.

Na perspectiva de desenvolvimento de software, o tempo é o esforço total que você levaria para desenvolver um determinado requisito ou funcionalidade. Geralmente esse tempo é medido em horas, mas também pode ser medido em dias e, em casos raros, em semanas.

**Exemplo:** *para fazer o front-end de um formulário de cadastro de clientes, você levaria 16 horas.*

Já o prazo é um intervalo ou período de tempo dentro do qual alguma atividade deve ser realizada. Se você tem uma atividade que leva 8h para ser feita, não significa que ela tem o prazo de um dia ou que será feita em um dia. Mesmo porque, ninguém trabalha certinho as 8 horas por dia, né (alô gerentes de projetos que fazem cronograma considerando que todo mundo trabalha 8h, tá errado isso aí).

**Exemplo:** *(1) esse formulário de cadastro de clientes, que leva 16 horas, tem um prazo de cinco dias para ser desenvolvido e entregue; (2) você pode estimar que o formulário de cadastro, de 16 horas, será entregue em um prazo de três dias.*

## Como desenvolvedores fazem estimativas?

Eu tenho um grande viés acadêmico-científico, então eu não poderia ficar somente na minha perspectiva para escrever este post, porque só minha vivência profissional não é um argumento forte para generalizar uma realidade do mercado.

Precisava de fatos e evidências sobre como pessoas da área de desenvolvimento de software fazem estimativas e quais as principais dificuldades que elas enfrentam. Assim, eu resolvi fazer um questionário online no SurveyMonkey:

**Talita Pagani**

@talitapagani

Pessoal, estou fazendo uma pesquisa sobre estimativas de software para um post que estou escrevendo, o questionário leva 2 minutinhos pra responder: [pt.surveymonkey.com/r/3PK55C5](https://pt.surveymonkey.com/r/3PK55C5)

RT dos parças

9:43 AM - Jan 2, 2018

27

49 people are talking about this

Vamos a uma breve análise dos resultados.

## Estrutura do questionário

Primeiramente, obrigada a todas as pessoas que contribuíram com a pesquisa! Obtive 90 respostas em 3 dias em que deixei o questionário aberto. Não estava tão rigorosa quanto à relevância estatística, então não tinha a pretensão de obter um volume maior de respondentes, até mesmo porque o plano gratuito do SurveyMonkey só permite 100 respostas, rs.

A pesquisa era composta por três questões técnicas e três questões demográficas, um formulário bem sucinto mesmo, somente com perguntas fechadas de múltipla escolha (*radio button*) ou seleção múltipla (*checkbox*). As questões técnicas foram:

1. Quais técnicas ou estratégias você utiliza quando te pedem para estimar o desenvolvimento de um projeto ou uma funcionalidade de um projeto?
2. Como você realiza as estimativas de desenvolvimento?

### 3. Quais as maiores dificuldades que você tem com relação a estimativas de desenvolvimento?

## Perfil dos respondentes

Dos 90 respondentes, 34% são desenvolvedores de software (back-end, web, mobile), 23% são desenvolvedores full-stack, 16% são de front-end, 10% são engenheiros(as) de software, 8% são designers e somente 2% são analistas de teste. Na opção “outros”, há respondentes que são: líder técnico, analista de suporte, gerente de projetos/analista de sistemas, Product Owner (PO)/analista de requisitos, dev full-stack / *team leader*, e arquiteto de software, sendo um respondente de cada uma dessas funções.

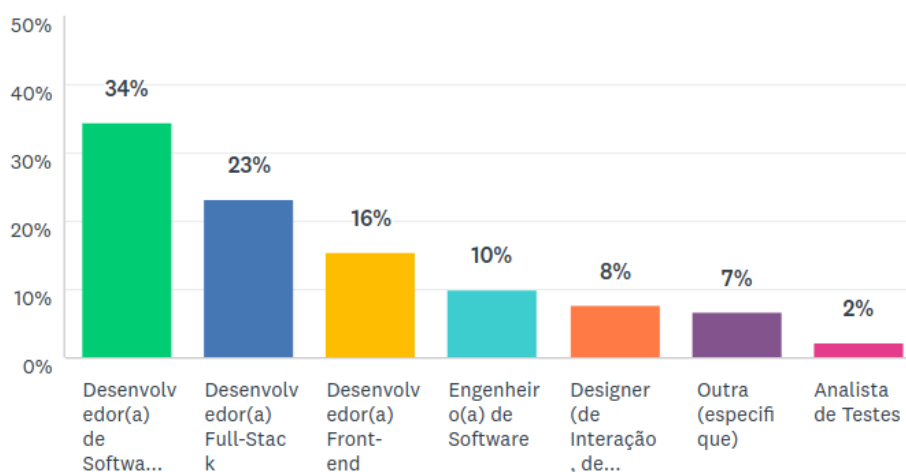


Gráfico de barras com a distribuição das funções ordenadas do maior percentual para o menor, sendo: 34% desenvolvedor(a) de software — back-end, web ou mobile, 23% desenvolvedor(a) full-stack, 16% desenvolvedor(a) front-end, 10% engenheiro(a) de software, 8% designer — de interação de interface ou UX, 7% outros e 2% analista de testes.

A maioria dos respondentes são de nível pleno, atuando entre 4 e 6 anos na área (36%). Houve uma distribuição idêntica de profissionais com mais de 10 anos de experiência em TI (19%) e profissionais juniores, com 1 a 3 anos de experiência (19%). Outros 18% estão no nível sênior, entre 7 e 10 anos de atuação e somente 9% dos respondentes possuíam de menos de um ano de atuação (7% = de 6 meses a 1 ano, 2% = menos de 6 meses).

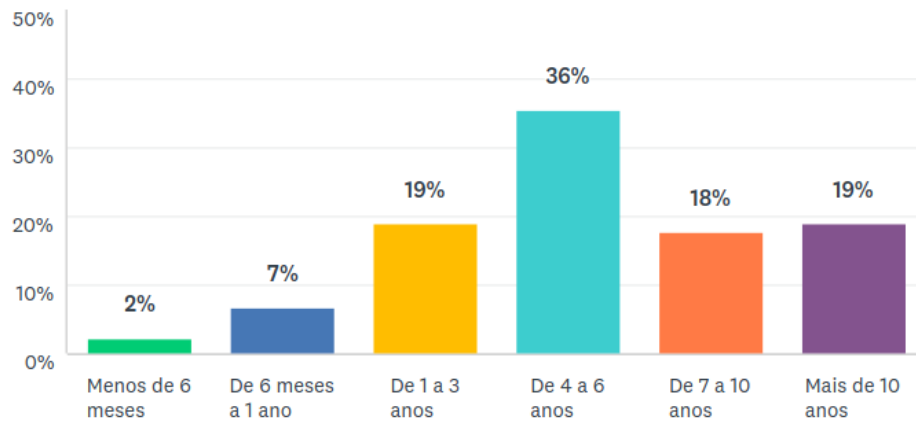


Gráfico de barras com a distribuição do tempo de atuação no mercado, sendo: 2% menos de 6 meses, 7% de 6 meses a 1 ano, 19% de 1 a 3 anos, 36% de 4 a 6 anos, 18% de 7 a 10 anos e 19% mais de 10 anos.

## As principais técnicas de estimativas utilizadas

Existem diversas técnicas que ajudam desenvolvedores na estimativa de software: de Planning Poker, vinda dos métodos ágeis, até a Estimativa por Analogia. Mas a mais conhecida ainda é o famigerado ***Cálculo Hipotético Universal Técnico Estimativo***, o **CHUTE**.

GIF do Carlos Alberto de Nóbrega rindo. Prassômetro disparou aqui.

Brincadeiras à parte, foi surpreendente ver que a técnica de estimativa mais utilizada pelo pessoal, independente do tempo de atuação no

mercado, é justamente o chute (41%), que deu até empate técnico com o segundo lugar, que foi o *Planning Poker* (39%)! Em terceiro lugar, vem a Estimativa por Analogia (30%) e, com uma boa diferença, os Pontos por Caso de Uso (PCU), com 14%, que confesso nunca ter utilizado.

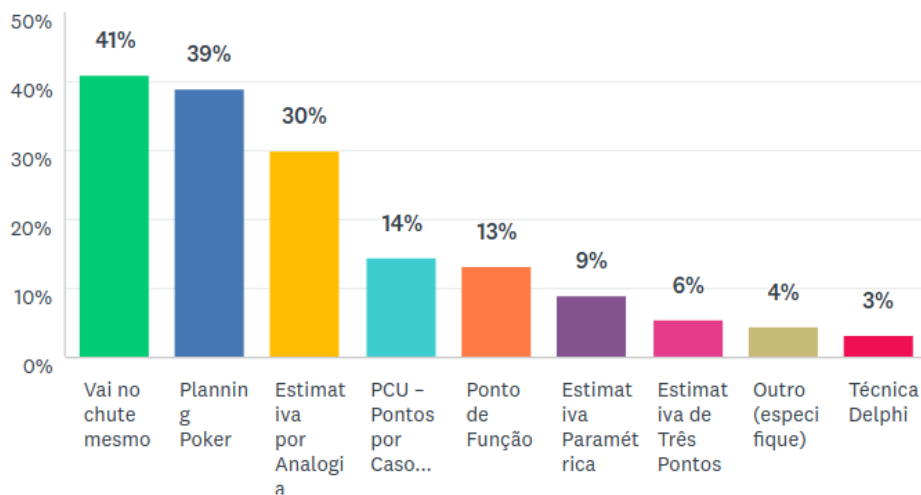


Gráfico de barras com a distribuição das técnicas de estimativas ordenadas do maior percentual para o menor, sendo: 41% vai no chute mesmo, 39% planning poker, 30% estimativa por analogia, 14% pontos por caso de uso, 13% pontos de função, 9% estimativa paramétrica, 6% estimativa de três pontos, 4% outros e 3% técnica Delphi.

Como a pergunta era de seleção múltipla, pode ser que em muitos casos as pessoas usem outras estratégias mas deixam de lado técnicas estabelecidas e optem por uma estimativa super hipotética porque foram pegadas de surpresa, houve pressão para fornecer um prazo de entrega ou não têm base para poder estimar o que foi solicitado.

Fazendo um comparativo das técnicas utilizadas por tempo de atuação no mercado, observei uma tendência curiosa. Profissionais juniores, com 1 a 3 anos de atuação, são muito propensos a “chutar” as estimativas (59%), mas este percentual cai abruptamente em profissionais de nível pleno (38%) e sênior (19%), que têm o *Planning Poker* como a técnica mais utilizada, sendo 41% e 56%, respectivamente. Porém, o número volta a crescer vertiginosamente nos profissionais com mais de 10 anos de atuação. Cerca de 47% responderam que usam o chute como técnica de estimativa.

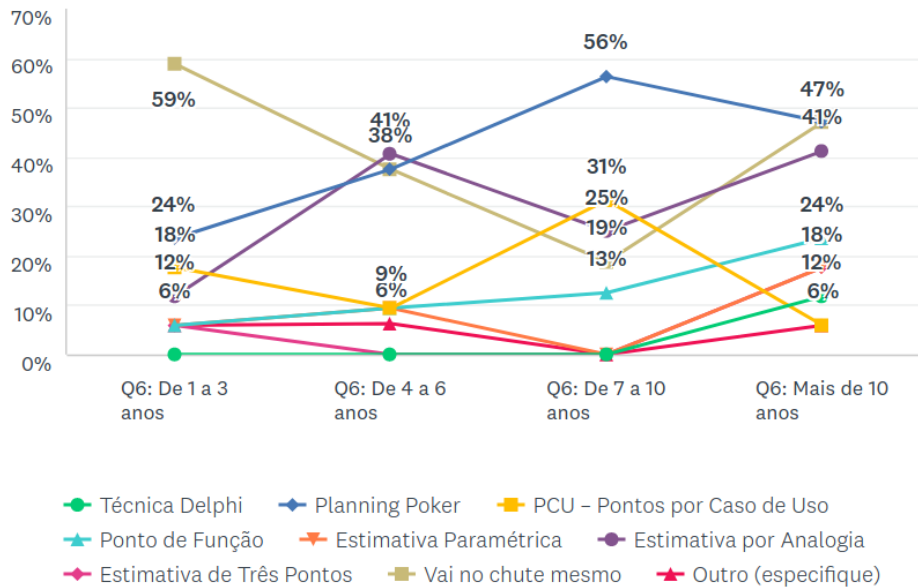


Gráfico de linhas mostrando a mudança no uso das técnicas de estimativas de acordo com diferentes níveis de senioridade dos profissionais respondentes.



Meme do Nick Young confuso onde ele olha com uma expressão franzida de perplexidade e na outra parte ele está com um sorriso confuso com sinais de interrogação ao redor dele.

O que será que acontece para eles voltarem a “chutar”? Maior conhecimento? Mais confiança nas suas habilidades? Maior pressão? Já estão no nível do dane-se e vai assim mesmo? Ainda não descobri, mas sintam-se à vontade para debater isso nos comentários.

## Como são realizadas as estimativas

Muitas empresas adotam estratégias colaborativas para estimar os projetos, onde os todo o time se reúne para discutir o escopo, realizar o planejamento e entrar em consenso sobre as estimativas.



Em times que usam métodos ágeis, como Scrum, esse processo é realizado no Planning Poker. Mas há casos em que os gestores perguntam individualmente para cada membro do time as estimativas para suas respectivas partes. E há momentos em que não nos sentimos seguros sobre como estimar e consultamos colegas mais experientes ou a pessoa que coordena o time. Eu mesma já passei por todas essas situações, mas queria entender o que é mais comum no mercado.

A forma mais utilizada (58%) é a realização das estimativas em conjunto com os colegas de trabalho, chegando em um acordo sobre as estimativas. Já a segunda é a realização de estimativas sozinho, com 33%. A terceira, com 31%, é a realização das estimativas com os colegas somente consultando as pessoas quando há dúvidas.

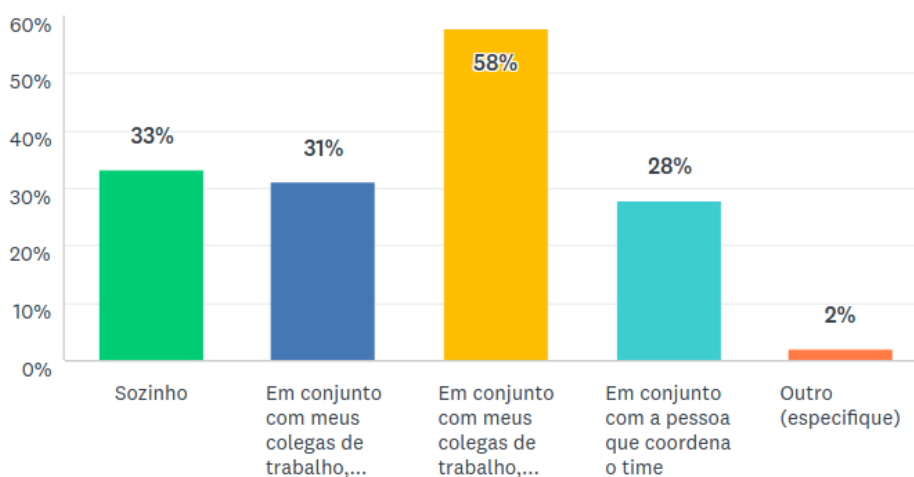


Gráfico de barras com a forma mais utilizada para fazer as estimativas, sendo: 33% sozinho, 31% consultando os colegas de trabalho, 58% em consenso com os colegas de trabalho, 28% em conjunto com a pessoa que coordena o time e 2% outros.

Comparando novamente por tempo de atuação, 53% dos profissionais juniores realizam as estimativas em consenso com os colegas, mas também fazem estimativas sozinhos (29%) ou somente consultando os colegas (29%). Já entre os profissionais de nível pleno e especialista (mais de 10 anos de mercado), 65% trabalham com este método de estimativas em consenso. E foi interessante observar como os profissionais especialistas valorizam a opinião dos colegas de trabalho: 59% indicaram que consultam seus pares no momento de realizar as estimativas.

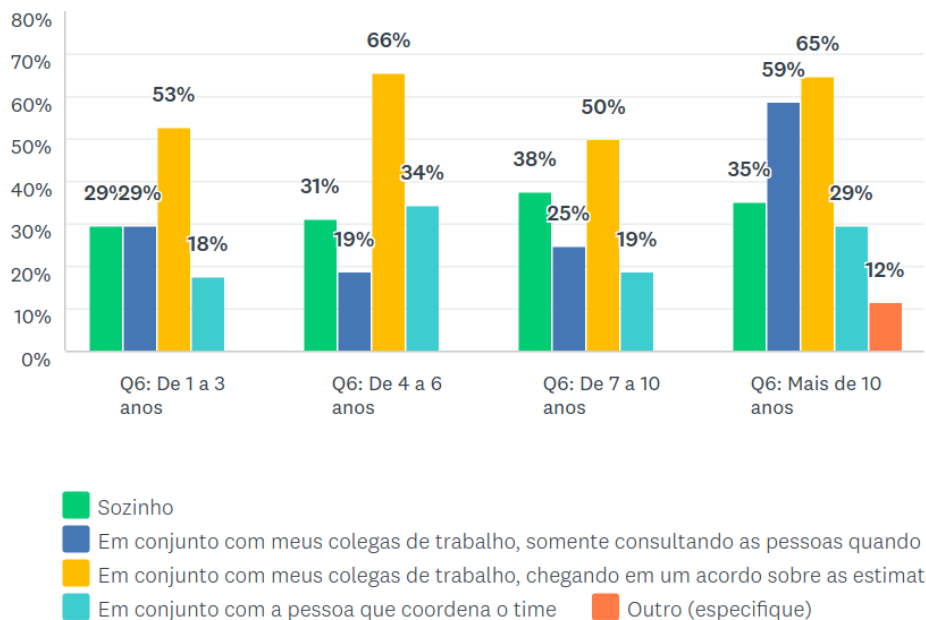


Gráfico de barras com a forma de realizar estimativas por tempo de atuação, sendo: De 1 a 3 anos — 29% sozinho, 29% consultando colegas de trabalho, 53% em consenso com colegas de trabalho, 18% em conjunto com a pessoa que coordena o time; De 4 a 6 anos — 31% sozinho, 19% consultando colegas de trabalho, 66% em consenso com colegas de trabalho, 34% em conjunto com a pessoa que coordena o time; De 7 a 10 anos — 38% sozinho, 25% consultando colegas de trabalho, 50% em consenso com colegas de trabalho, 19% em conjunto com a pessoa que coordena o time; Mais de 10 anos — 35% sozinho, 59% consultando colegas de trabalho, 65% em consenso com colegas de trabalho, 29% em conjunto com a pessoa que coordena o time e 12% outros.

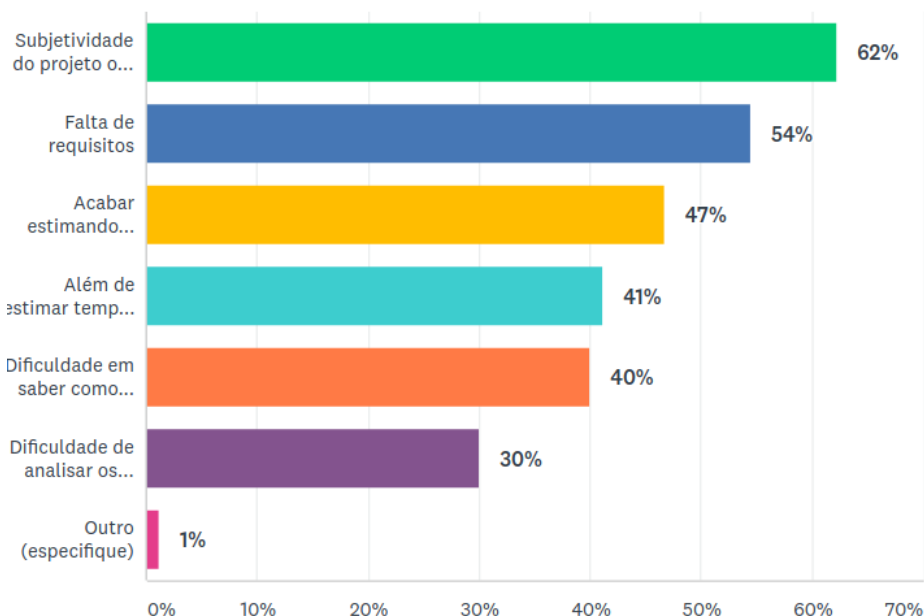
Mesmo que sua empresa não adote métodos ágeis ou técnica de *Planning Poker*, realizar estimativas de forma colaborativa entre todos os membros do time é muito válido para aumentar o comprometimento na entrega, tirar dúvidas, trocar ideias, esclarecer requisitos e perceber riscos que você não perceberia sozinho(a).

## As maiores dificuldades

Por fim, será que minhas hipóteses apresentadas no início do texto sobre as dificuldades em estimar projetos estariam corretas? Além da subjetividade de requisitos, que é algo que foge do controle de quem é dev e vem de cima, pensei que um dos maiores problemas seria a própria habilidade de desenvolvedores em estimar.

O que percebi, foi que os maiores problemas vêm da definição de escopo e análise de requisitos dos projetos. Para 62% dos respondentes, a subjetividade do projeto ou da funcionalidade a ser desenvolvida é uma das maiores dificuldades no momento de estimar os esforços, seguido da falta de requisitos (54%).

Em seguida, vêm as dificuldades que são mais relacionadas à habilidade de cada dev: acabar estimando tempo/esforço demais ou pouco tempo/esforço (47%); além de estimar tempo, conseguir estimar o prazo de entrega (41%); dificuldade em saber como calcular quanto tempo levaria para desenvolver o que foi solicitado (40%).



Comparando por tempo de atuação, a preocupação com a subjetividade dos requisitos tende a ser uma preocupação maior com o passar do tempo. Enquanto este fator é preocupante para 41% dos profissionais juniores, ele acaba sendo uma dificuldade para 76% dos especialistas. Já a dificuldade em saber como calcular quanto tempo levaria para desenvolver o que foi solicitado cai de 53% nos profissionais juniores para 19% nos profissionais seniores, porém, aumenta um pouco para 35% nos profissionais especialistas.

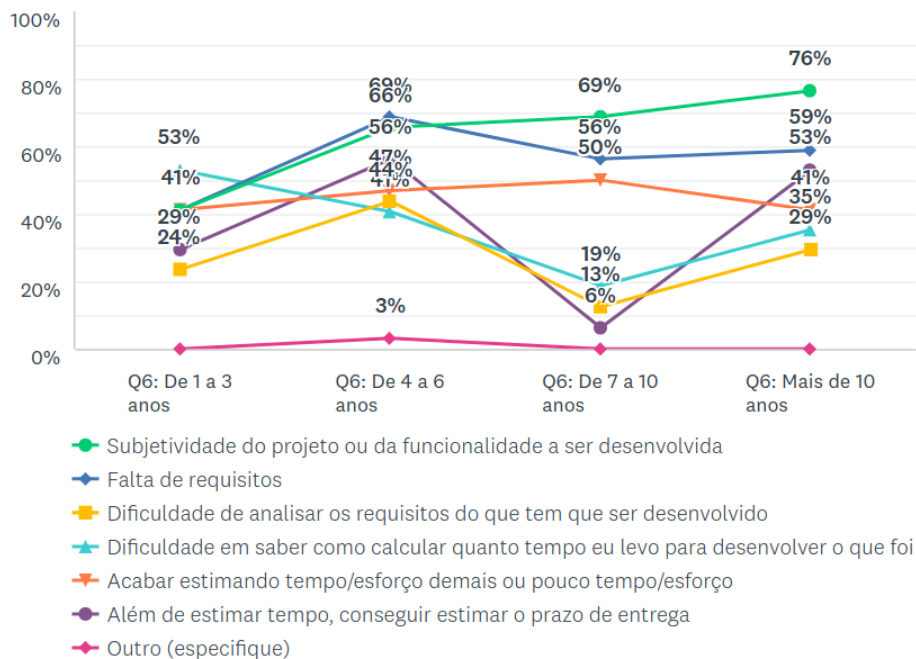


Gráfico de linhas mostrando a mudança nas dificuldades para estimar projetos de acordo com diferentes níveis de senioridade dos profissionais respondentes.

Ou seja, a tendência é irmos refinando nossa habilidade em realizar estimativas com o avanço na carreira, porém, passamos a ter uma preocupação maior com as dificuldades que fogem da nossa alçada e são de responsabilidade da gestão do projeto/produto. Mas mesmo nesses casos, há formas de contornar os empecilhos.

. . .

## Considerações Finais

Estimativas de software são uma tarefa desafiadora para profissionais de TI em qualquer momento da carreira. Há muita pressão nessa atividade e também ocorre excesso de confiança para confiar em uma percepção subjetiva pessoal (o chute).

Porém, os maiores problemas para estimar software partem da forma como os requisitos chegam até à equipe: incompletos, subjetivos ou inexistentes. E aí entra um processo em que precisamos ter um trabalho de tentar extrair mais informações e decompor o que já sabemos para transformar um requisito abstrato em um conjunto de partes concretas.