

Escola Regional de Engenharia de Software 2020

Melhores Práticas para Desenvolvimento Remoto de Software

Marco Tulio Valente

ASERG, DCC, UFMG

12/11/2020

Hipótese 1: Pandemia vai acabar!



Hipótese 2:

Legado da pandemia será um aumento expressivo de **trabalho remoto**

Algumas evidências



Pinned Tweet



Dropbox  @Dropbox · Oct 13



It's clear that distributed work is here to stay, and we believe that brings an opportunity to redesign the way we work for the better. Starting today, Dropbox is becoming a **Virtual First company**. bit.ly/33Ti9I1

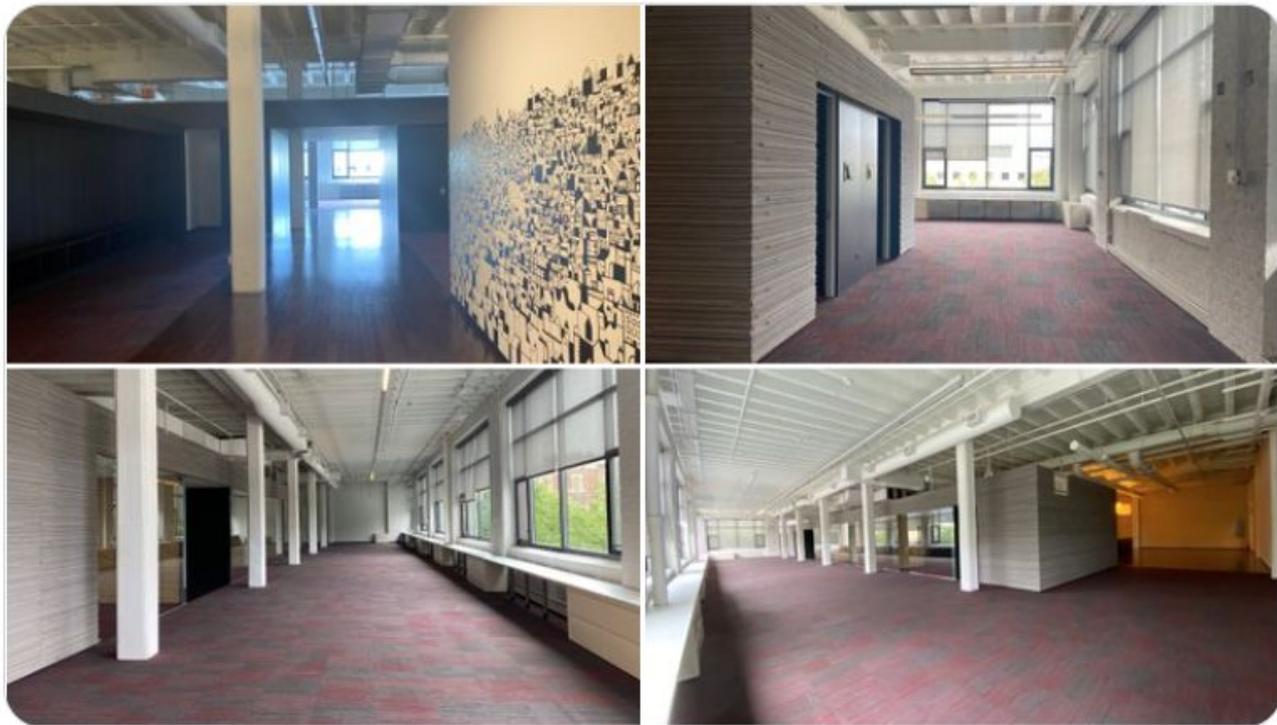


Jason Fried  @jasonfried · Jul 30



Replying to @jasonfried

Some last day empty space pics.



 18

 4

 258





Tobi Lutke

@tobi



As of today, Shopify is a digital by default company. We will keep our offices closed until 2021 so that we can rework them for this new reality. And after that, most will permanently work remotely. Office centrality is over.

11:55 AM · May 21, 2020



40.3K



9.4K people are Tweeting about this

Permanently remote workers seen doubling in 2021 due to pandemic productivity: survey

By Gertrude Chavez-Dreyfuss

2 MIN READ



NEW YORK (Reuters) - The percentage of workers around the world that is permanently working from home is expected to double in 2021 as productivity has increased during the coronavirus pandemic, according to a survey from U.S.-based Enterprise Technology Research (ETR).

O que **não** está no contexto da
palestra

Não vamos falar de trabalho remoto na pandemia

- Ansiedade
- Incerteza
- Medo
- Crianças em casa
- etc

Não vamos tratar das vantagens de TR

- Mais tempo com família
- Menos tempo no trânsito
- Menos interrupções
- etc

Não vamos tratar das desvantagens de TR

- Falta de contato social
- Falta de motivação
- Fadiga de reuniões virtuais
- etc

Trabalho remoto não é uma bala de prata

Já que trabalho remoto será mais comum, quais as **melhores práticas** que devemos seguir?

Trabalho remoto não é novidade!

Trabalho remoto - Pré-COVID

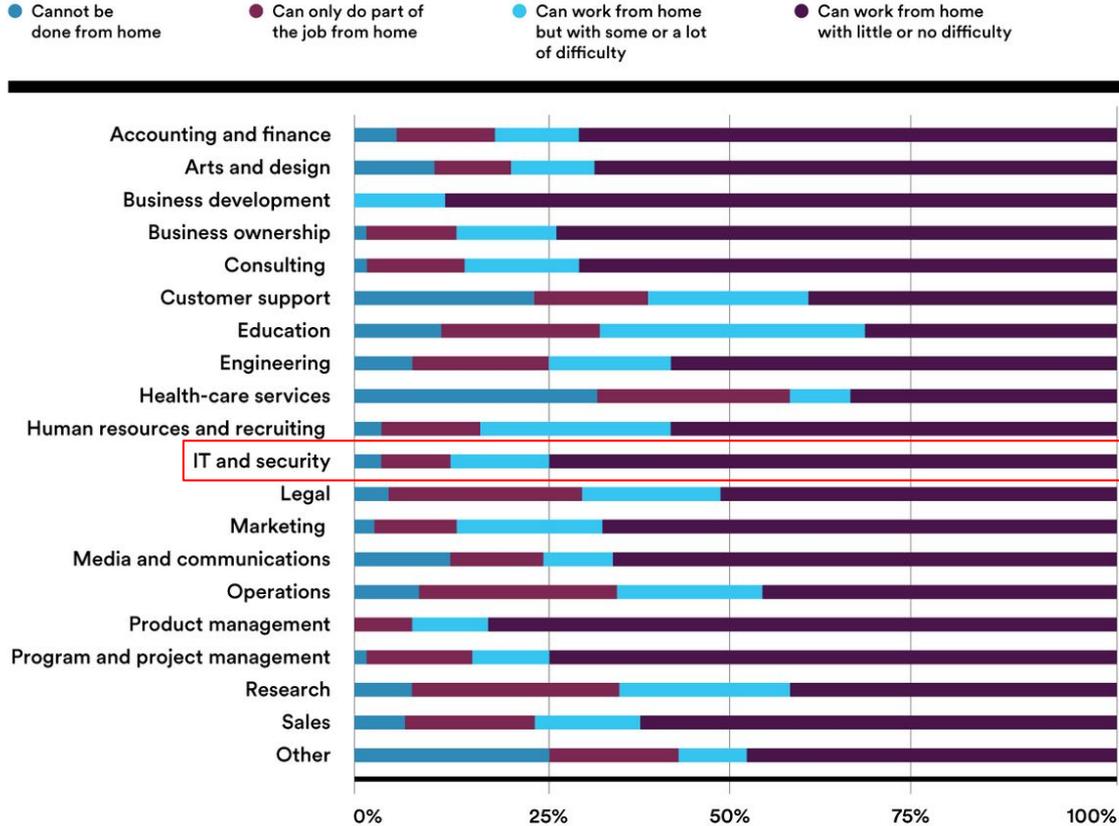
- GitLab (~1,300 colaboradores, 65 países)
- Automattic (~900 colaboradores, 68 países)
- Basecamp (~50 colaboradores, 32 cidades)
- Zapier (~250 colaboradores, 28 países)
- DuckDuckGo (100-150 colaboradores)
- Discourse (< 10 colaboradores)

Trabalho remoto - Pré-COVID

- GitLab (~1,300 colaboradores, 65 países)
- Automattic (~900 colaboradores, 68 países)
- Basecamp (~50 colaboradores, 32 cidades)
- Zapier (~250 colaboradores, 28 países)
- DuckDuckGo (100-150 colaboradores)
- Discourse (< 10 colaboradores)

**Excelente
oportunidade
para devs
brasileiros**

Suitability of job for remote work by industry



TI é uma das indústrias mais "adequadas" para trabalho remoto

Source: Slack survey of knowledge workers, conducted March 23-27 (n=2,877)

Como fazemos software hoje em dia?

Fizemos a seguinte pergunta para 415 devs brasileiros

Qual é o processo de desenvolvimento usado por sua empresa?

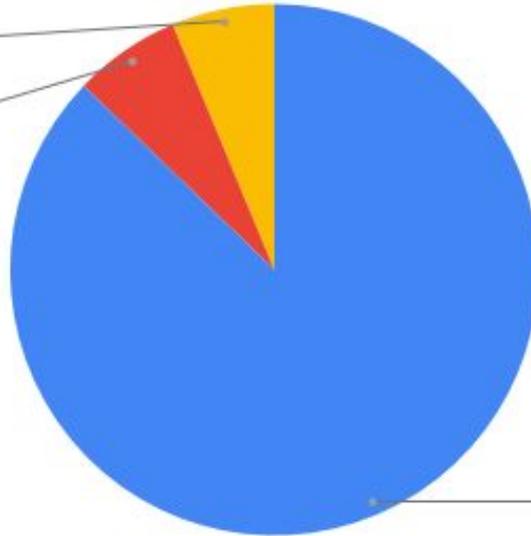
Resultados relativos a 415 respostas de desenvolvedores de software brasileiros

Outros

6.3%

Waterfall

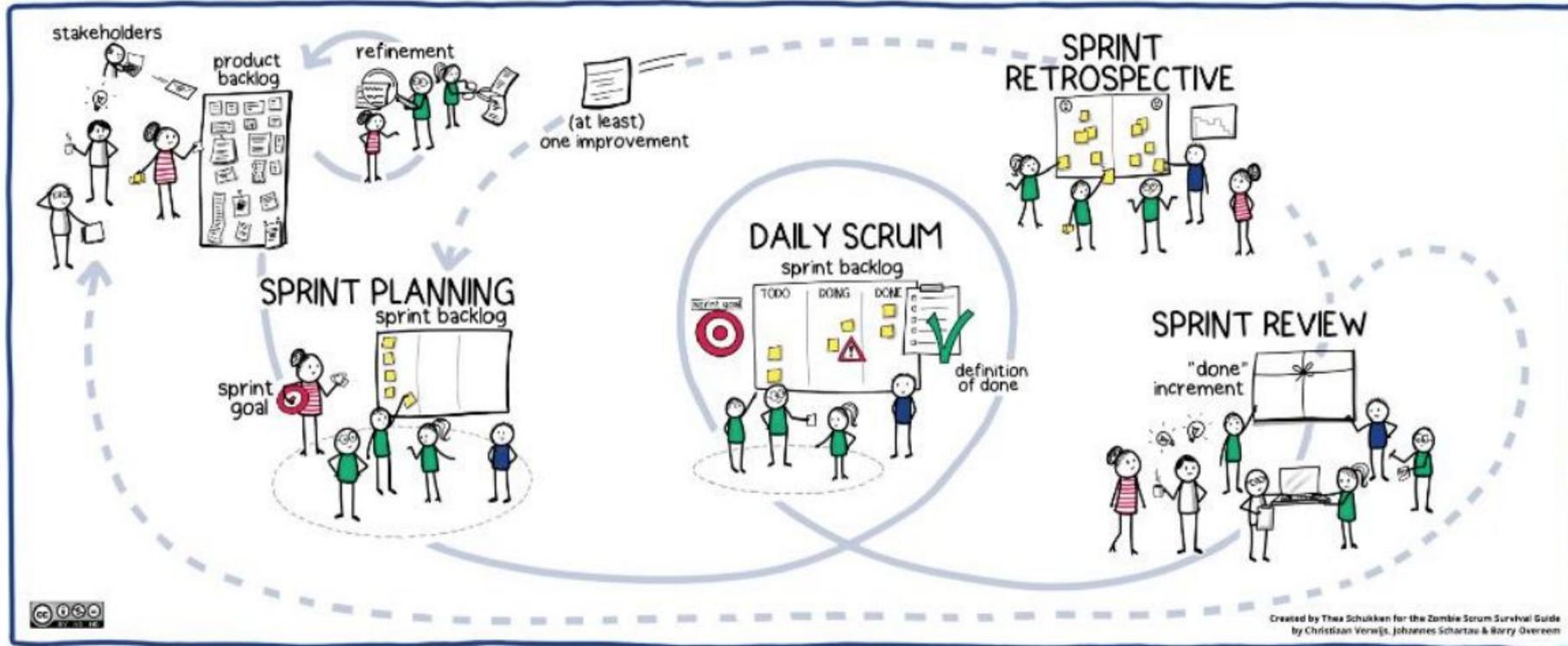
6.5%



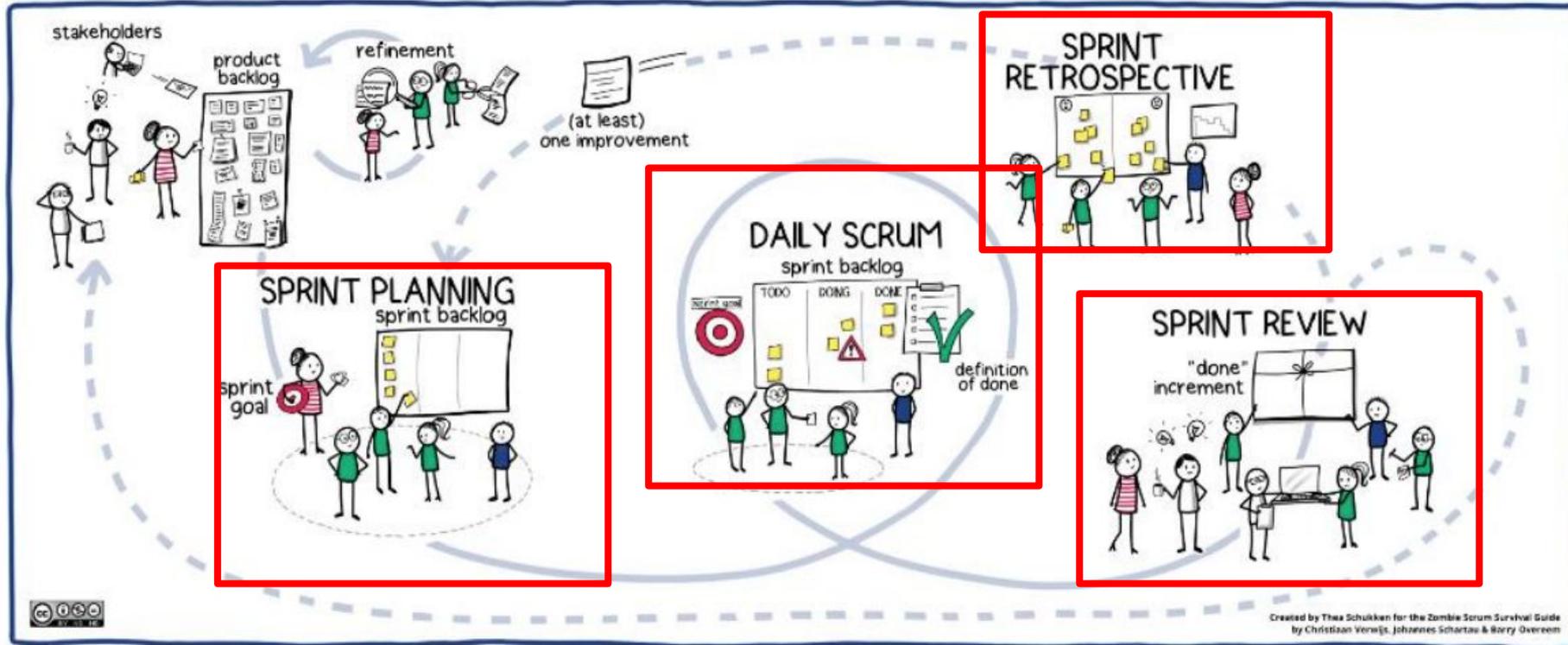
Ágil

87.2%

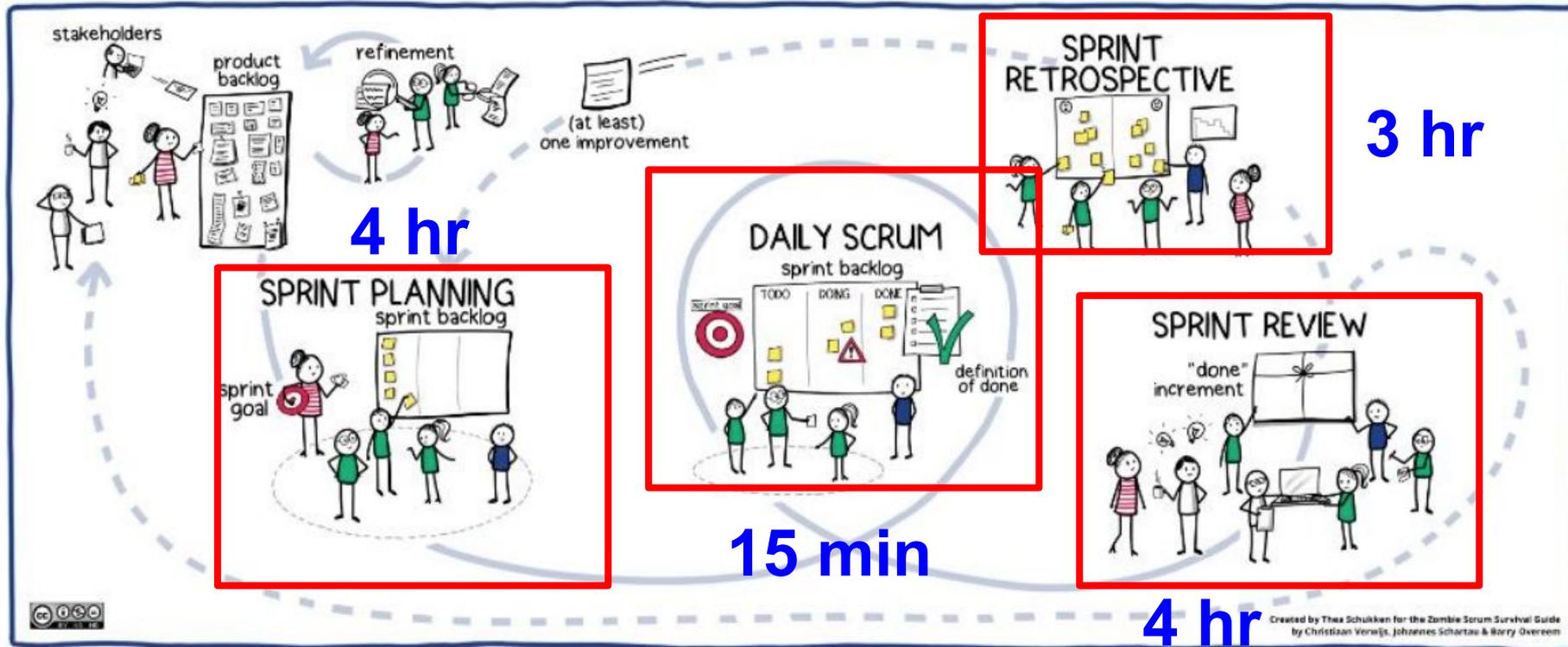
Scrum em 1 slide



Scrum em 1 slide



Scrum em 1 slide



**Não parece ser tão difícil, pois maior evento
dura 4 horas, para sprints de 15 dias**

Porém, não é tão simples assim....

Problema: natureza "síncrona" das comunicações durante um sprint

Hoje ...



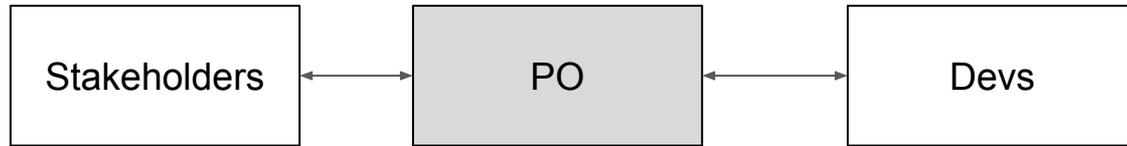
PO

Devs

Product Owner senta junto dos desenvolvedores e
"explica" requisitos para eles

Fonte: @engsoftmoderna

Hoje (Scrum)



- Durante sprint, PO explica histórias (requisitos) para devs
- Troca-se documentação formal e escrita por documentação verbal e informal
- Conversas entre PO e Devs

Mandamento #1 de Trabalho Remoto:
minimize comunicação síncrona

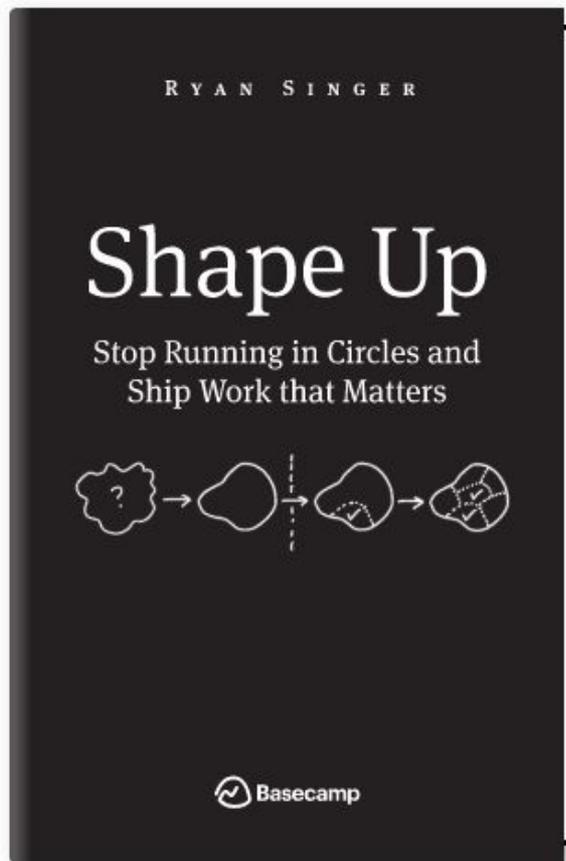
Não dá para ficar o dia inteiro no Zoom,
Slack, WhatsApp, mail, etc.

Será que temos que voltar para Waterfall?



Provavelmente não!

Basta dar um pequeno passo para trás



Processo de desenvolvimento da BaseCamp

<https://basecamp.com/shapeup>

Três Fases

- Shape Up
- Ciclos (~ sprints)
- Cool down

Shape Up

- Especificação simplificada de requisitos
- Resultado: **pitch** (documento simplificado de requisitos)
 - Problema
 - Esboço da solução (sketches)
 - Rabbit holes (soluções para possíveis impasses)
 - No-gos (limitações que serão aceitas)

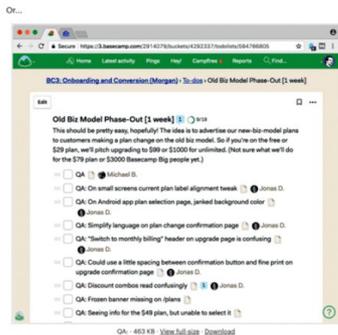
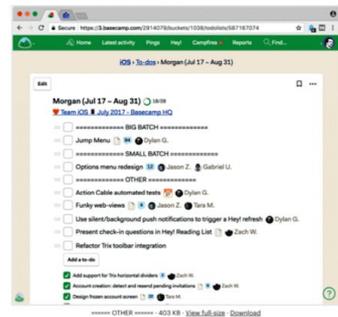
Exemplos de Pitches

Exemplo de Pitch

To-Do Groups

Jason Fried
Sep 12, 2017 · Notified 3 people

For 13 years we've done stuff like this...



What we're trying to do is create simple groups of to-dos within a single list. So we're hacking it. We're either creating "Artificial Dividers" or prepending a batch of to-dos with a label like "QA" or our old standby, the trusty - which is code for "maybe". Yeah it works, but it's messy, and it's not a refined pattern to share with customers. 13 years in, it's time to level up.

To-do groups

A couple weeks ago, Ryan and I spent an afternoon working out an idea out to make to-do groups first class citizens without adding noticeable complexity to the purity of today's to-do lists. We're pleased with where we landed, and we'd like to put it forward as a big batch project this cycle.

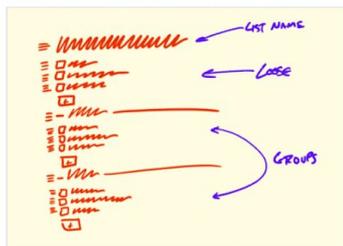
Here's how it works:

First, to-dos without groups will look identical to today's to-dos. We won't add any additional UI around them, no new buttons, etc...



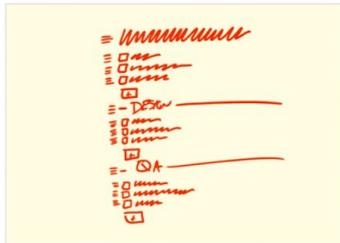
A basic list remains the same - 185 KB · [View Full Size](#) · [Download](#)

But, if you add a group (or groups), it'll look like this...



A list with some loose to-dos plus two groups of to-dos - 230 KB · [View Full Size](#) · [Download](#)

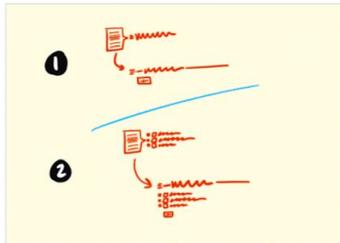
This means now you could make a list for a specific scope that had a section for design-specific to-dos, and QA-specific to-dos.



With labels - 189 KB · [View Full Size](#) · [Download](#)

A group is defined by a divider. You can name a divider anything you want. It can't be checked off, it's not a to-do itself. Any to-dos you put below the divider become part of that group. And critically, groups always go below any ungrouped (loose) to-dos. This simplifies a bunch of behaviors, and prevents loose to-dos from being lost between groups. If it's loose, it's at the top.

You can create a divider in two ways...



Two ways to make a divider - 152 KB · [View Full Size](#) · [Download](#)

1. Click the hamburger menu next to the list title. You'll get a menu. One of the items will be "Create a group" (or add a divider, exact language TBD). It'll then add a blank divider at the end of the loose to-dos, and scroll right to that point so you can name it.
2. You can multi-select (shift key) multiple loose to-dos, and instantly create a group from those to-dos. If you do this, an unnamed divider will be created, and the to-dos will be placed under that divider.

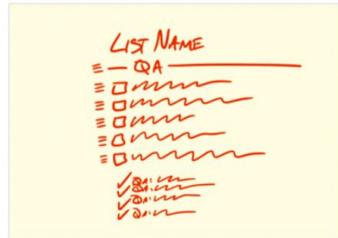
Key to this concept is that we'll add "Add to-do" buttons to the bottom of each group (and the loose to-dos at the top). So once a divider exists, you can add to-dos directly to that section. If we didn't do this, and we only had one add-to-do button at the top or bottom, you'd have to add and then move into place. That's a hassle. It's much better to add it place when we have places to add.

Completed to-dos will still be grouped into a single collection at the bottom of the entire list. If they were part of a group when you checked them off, we'll prepend the group name before the to-do like so...



If you uncheck a completed to-do from a group, it could go back up to the group (assuming the group divider hasn't been deleted) - 185 KB · [View Full Size](#) · [Download](#)

Groups will also have their own perms. So you can click the title of a group/divider, and you'll see a new page with just the to-dos in that group...



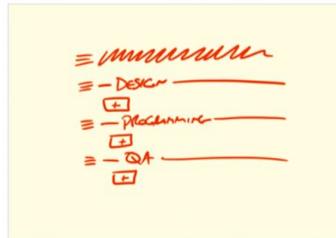
Group perms for QA - 195 KB · [View Full Size](#) · [Download](#)

What we like about this concept overall is that it's very straightforward. It doesn't change existing to-do lists at all. No new permanent UI in the way, no behavior changes if you don't use groups.

But if you want to level up your organization - and we'd use the shit out of this - then you can add groups to a list. No groups within groups, no indenting - just as many one-level groups as you'd like.

There are open questions. Things like if you move the divider, do all the items move with it? I'd say yes since it's a group, but we have to work that out. Also, since you can't put groups above loose to-dos, we'd have to prevent dragging above a certain point, but that's doable as well. I'm sure there are a few other questions as well, but we can work those out as we go.

BONUS: Down the road we'd like to explore adding to-do templates to BC3. We have project templates, but now we're just talking to-do templates. You can imagine creating a template with no to-dos, but with groups in place. This is a process boost - people can organize projects in similar ways with just a little bit of structure like this...



Just used in place - 105 KB · [View Full Size](#) · [Download](#)

Group Notifications



Last November I wrote a pitch for adding Groups aka BCX to BC3: [Group - Product Strategy](#).

After looking at the pitch, Jason pointed out it wasn't entirely clear where to expect Groups to appear. Example: Suppose you could autocomplete a Group when you @mention somebody. Does that set an expectation you should see Groups in other places where you autocomplete people, like when assigning To-Do's?

We could answer "yes" and expose Groups everywhere we can think of. But that increases the scope. And it's a bad product design smell to add functionality for "completeness" sake. It's better to understand use cases and target them.

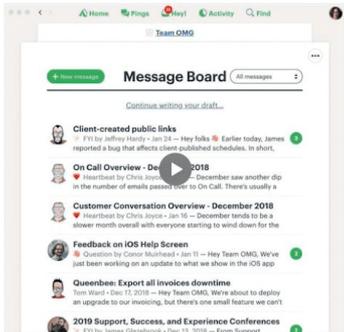
With our last BC3 cycle approaching, I wanted to take one last swing at this. Here's a narrower use case and a narrower solution that I think sets clear expectations about where and how Groups should appear in the app.

Narrowing down a problem case

To find a more specific use case to target, I turned to Kristin for perspective. She and others in support have wanted Groups for a long time.

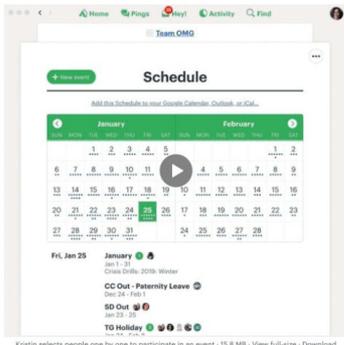
I asked Kristin to demonstrate workflows she's already doing today that are painful without Groups. The aim was to see some concrete "before" cases to design against.

She had two really good ones. Both of them involve **narrowing down** ~50 people to 15. Here's a video of Kristin hunting around to check off support people when posting a Message:



Narrowing recipients down when posting a message - 23.9 MB - [View full size](#) - [Download](#)

The same problem comes up when choosing who an event is "with." Here she has to pick 15 people one by one using the autocomplete UI:



Kristin selects people one by one to participate in an event - 15.8 MB - [View full size](#) - [Download](#)

Watching these videos, you see a palpable pain. With this as the "before," it's clear how Groups could offer a better "after" in this use case.

@Mentions: Cool, but maybe not calm enough

After this, I asked: can we find a similar pain point involving @mentions? We couldn't come up with one as compelling.

It's easy to think of cases where mentioning everybody via a Group could be handy. But that's good for our Hey menus or how we communicate with each other? With 15 people behind a single @mention, it would be easy to spam each other without thinking too hard about it.

With that in mind, I looked for broad design solutions for those two use cases only.

Solution for choosing Message subscribers

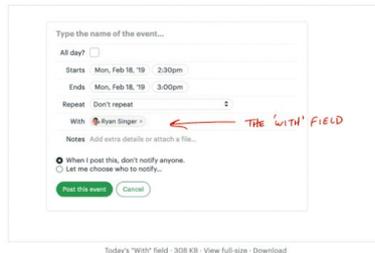
We could use a BCX-style approach to solve the first case. We have a single modal today for setting Subscribers app-wide, whether you are posting a new Message or changing who gets notified about some commentable. That "Who should be notified?" modal could offer Groups at the top. Clicking them checks members of the Groups below.



Groups on the Who Should Be Notified? modal - 90.7 KB - [View full size](#) - [Download](#)

Do Events get a different solution?

Solving the Events "with" field case is not as straightforward. The current UI is a single autocomplete field.



We could offer Groups inside that autocomplete, but that's a whole bunch of problems follow. We don't want to set the precedent that a Group might appear inside any autocomplete, per JF's concerns in the intro above. And if we did... there are still UX problems. Suppose you chose a Group in the autocomplete. Would the Group name appear then in the With field? That'd be misleading because maybe only 2 people from the Group are on the project. We could insert the people from the Group into the autocomplete field instead of the Group... but then it's hard to see which Groups you already added. So autocomplete Groups are out.

Next idea: What if we replaced the autocomplete with a modal picker ala "Who should be notified?" Then we could re-use the same solution as messages.

Technically that could work. Some "Choose participants..." button would launch the modal and then populate the field.

However, this could be a UX step backwards from today. Suppose the vast majority of people are autocompleting one or two names. That's a very different behavior from choosing Message subscribers, with everyone checked by default. Depending on the number of people on the project, you might have to scroll through a few screens in the modal to check the one other name you want. It's easier to autocomplete.

I took a look at the data for perspective. Indeed, the majority of Events have one or two participants. And the pattern is the opposite for Message subscribers. Here's a peek (explaining the analysis would take another write-up).



The data also showed that our use case — choosing 14 people in the With field — is extremely rare. So we don't want to rethink the UI just for us on this one.

Workaround: This is only about Subscribers

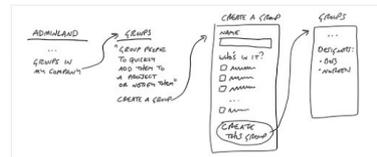
I came back to Kristin with a proposal: if she had to stop using the "With" field and instead relied on the "who should be notified" featured, would that still work for her when she posts events?

She said it would be fine for us. For others, it could conceivably be a problem because setting Subscribers vs. Participants isn't the same — Participants get a calendar integration that Subscribers don't. But in our case — and we're the outliers here — it's not a problem.

This suggests we can confine the Groups feature entirely to "Who should be notified?" — the Subscribers modal.

Proposed solution

We could define these "Subscriber groups" in Adminland, as pitched previously:



Adding Groups on Adminland - 435 KB - [View full size](#) - [Download](#)

At a minimum, we could confine the new functionality to that Subscribers modal.

If we want to, we could expand to also invite people to projects with the same approach. The modals are nearly identical. JZ's sketch for that is on the [prior Diigo](#).

What we're not doing

This goes a step further narrow than the last pitch. No autocompleting Groups anywhere. Still no connection with Teams. Just a better way to narrow down Subscribers any place in the app that we let you choose Subscribers via the "Who should be notified?" modal.

Last chance for this one in the next cycle! Hope this sharpened it down enough.

Shape Up

- Quem propõe e escreve o pitch?
 - Gerentes sêniores (produto)
 - No caso da Basecamp: 4 pessoas
- Processo de escrita:
 - Assíncrono
 - Reunião final, síncrona, para decidir os pitches do próximo ciclo

Ciclos

- Duração: 6 semanas
 - Mais longos que sprints de Scrum
 - Embora possam existir ciclos de duas semanas
- Não existe prorrogação na duração de um ciclo

Times

- Tamanho: 2 devs + 1 designer
 - Ainda menores que os times Scrum
- Autonomia:
 - Implementar pitches
 - Definir horários de trabalho, reuniões, daily, etc

Como os devs recebem o pitch e como os times são menores, necessidade de coordenação e comunicação síncrona é também menor

Cool Down

- Duração: 2 semanas
- Tempo para os devs "respirarem"
 - Corrigir bugs
 - Refactorings
 - Estudar um nova tecnologia
 - etc

Concluindo

Boas práticas para trabalho remoto [dentre outras]

- **Trabalho remoto ~ Trabalho assíncrono**
- Algumas práticas:
 - Documento simplificado de requisitos
 - Micro-times
 - Período de cool down, entre sprints

Obrigado!

Marco Tulio Valente

ASERG, DCC, UFMG

Foco: Práticas de desenvolvimento de software

- **Não** vamos tratar de recomendações genéricas:
 - Ter um bom espaço de trabalho
 - Ter boa conexão com a Internet
 - Ter bons equipamentos
 - etc