



Como otimizamos a entrega de código

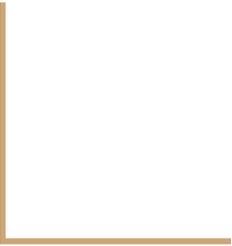
*...e isso melhorou a nossa
agilidade*

Luís Fernando Murara



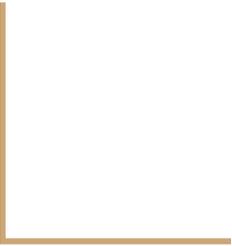


Por que estamos levando tanto tempo
para entregar nossos cards (no Kanban)?





“Só mais um detalhe e tá pronto!”





Kanban

Quebra de tarefas



WIP



Desenho de Solução

ToDo

Doing

Review/QA

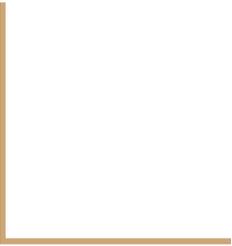
Shipping

Throughput: 2 a 7 cards por semana
Cycle Time: > 14 dias em alguns casos





O escopo dos cards está muito grande



**PRs Grandes
>800 linhas**

Os Pull Requests estão muito grandes

+1,410 -314 

Verificamos que...



Pull Requests
envelheciam



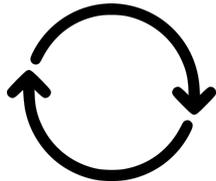
Complexidade
cognitiva aumentava



A demora na entrega
desmotiva os devs



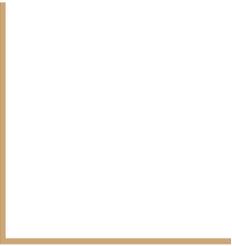
Reviews lentos



Muitos ciclos de
review

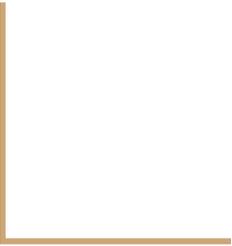


Qual o escopo ideal para um card?





O que é um PR pequeno?



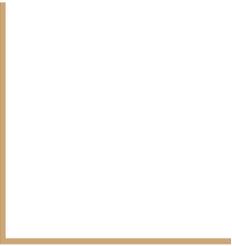


Quantas linhas ele deve ter?



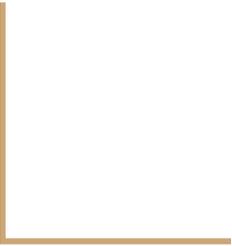


Como lidar com dependências entre cards?





*Após fazer debates internos e
benchmark com outros times...*





Um card deve fazer
apenas uma coisa!



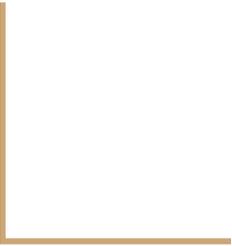


O PR pode ter apenas
UMA linha de código.





A entrega de valor pode estar em
apenas UMA ÚNICA linha de código!





A dependência entre os
cards é algo aceitável.



Quebrando melhor as tarefas



PRs menores que 800 linhas
Cards que fazem apenas uma "coisa"

Desenho de Solução

ToDo

Doing

Review/QA

Shipping

Throughput: 4 a 7 cards por semana
Cycle Time: > 12 dias em alguns casos





Notamos que...



Refactor além do escopo



Falta de detalhamento



Descobertas (isso é OK)



Determinamos que...



Nunca deve
usar "E"



Refactor em cards e Pull
Requests separados

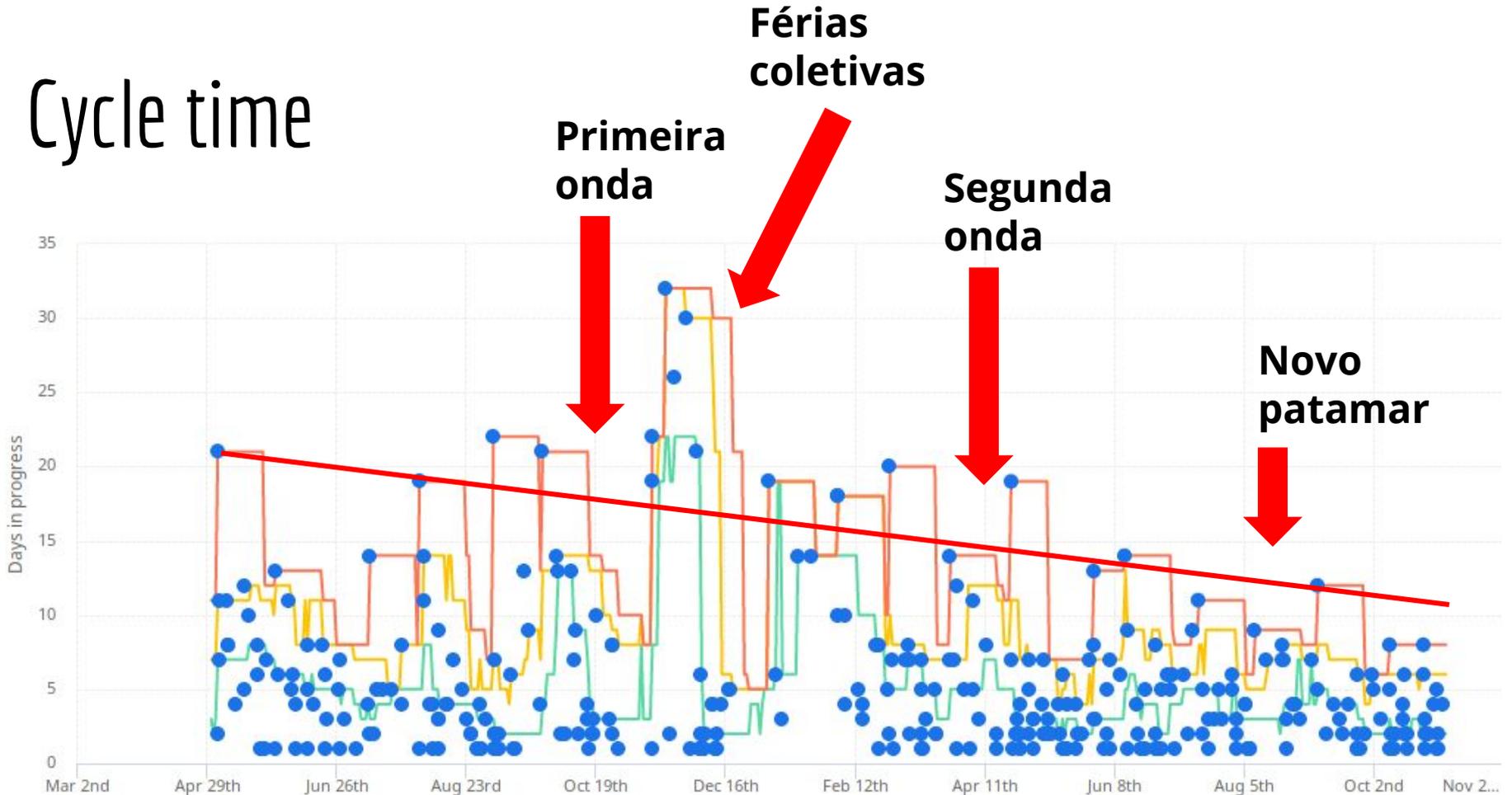


Mais críticos na
quebra de tarefas



Descobertas cards em
Pull Requests separados

Cycle time



Tempo até o merge

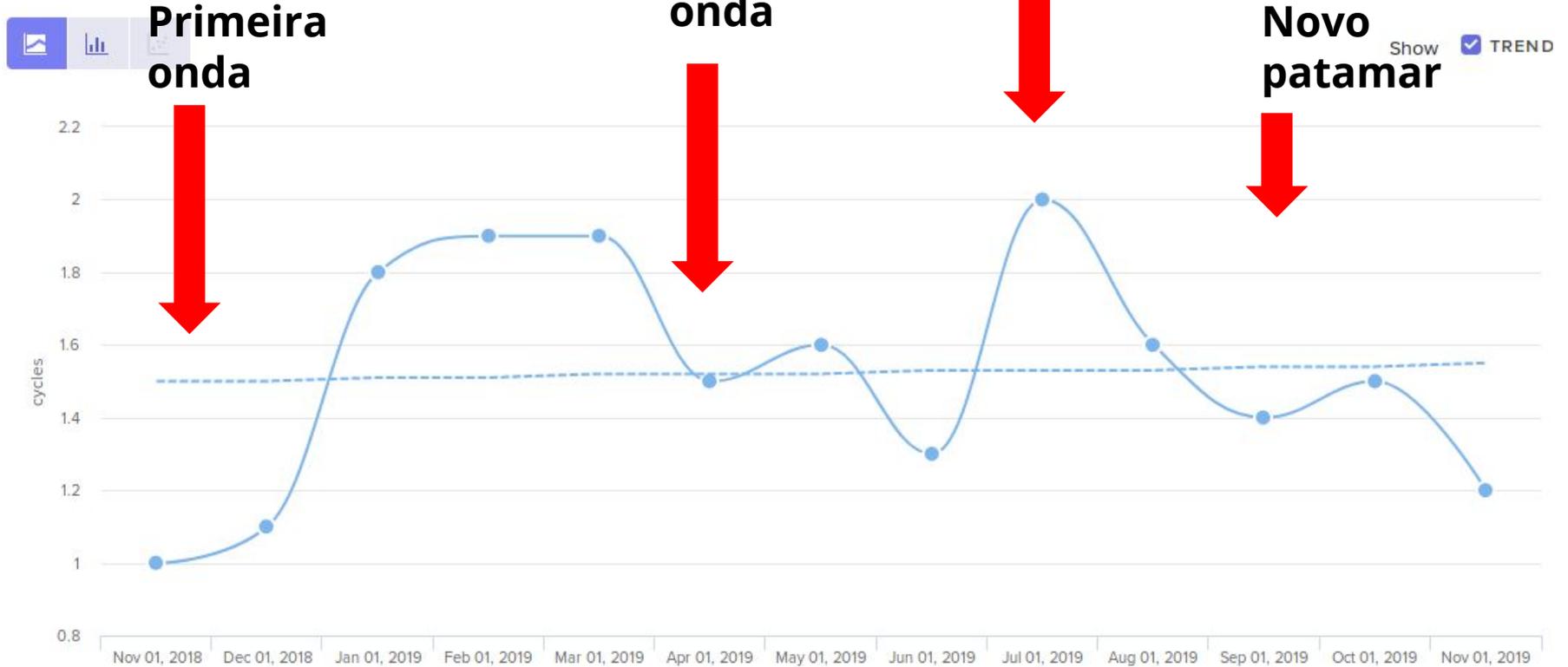


Show TREND



Fonte: Velocity

Ciclo de review

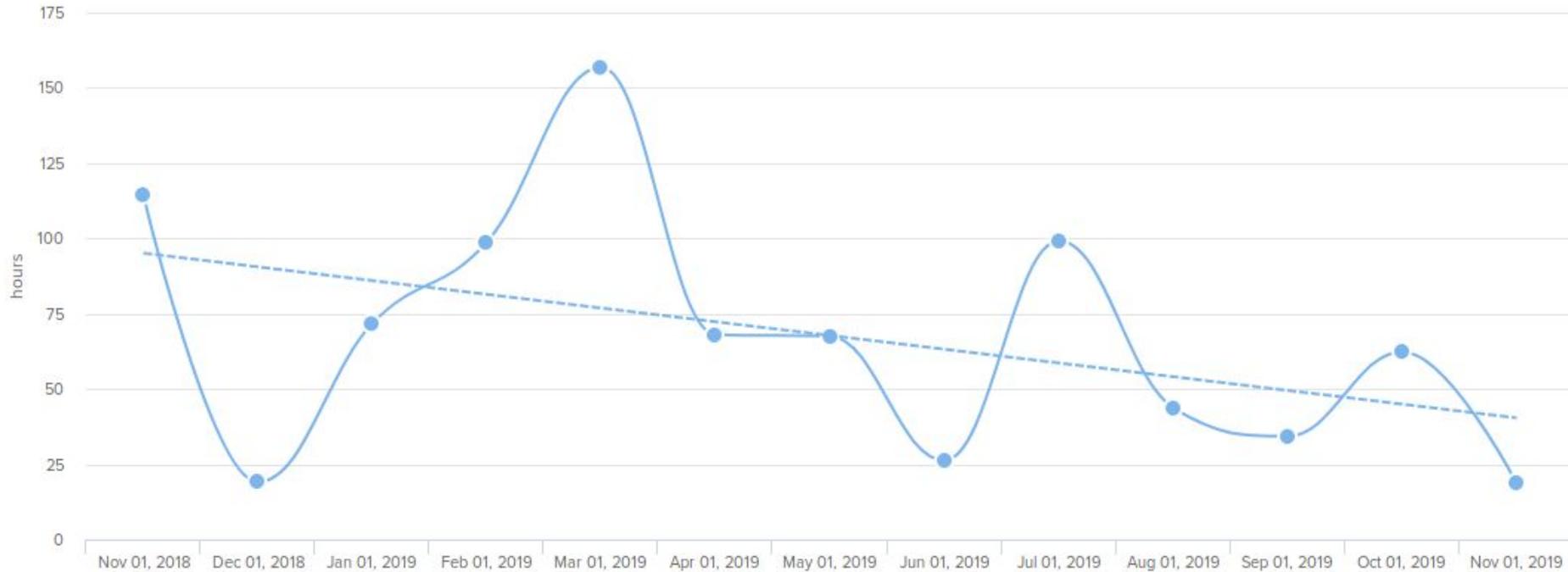


Fonte: Velocity

Velocidade do review



Show TREN



Fonte: Velocity

Analizando o todo



**Primeira
onda**

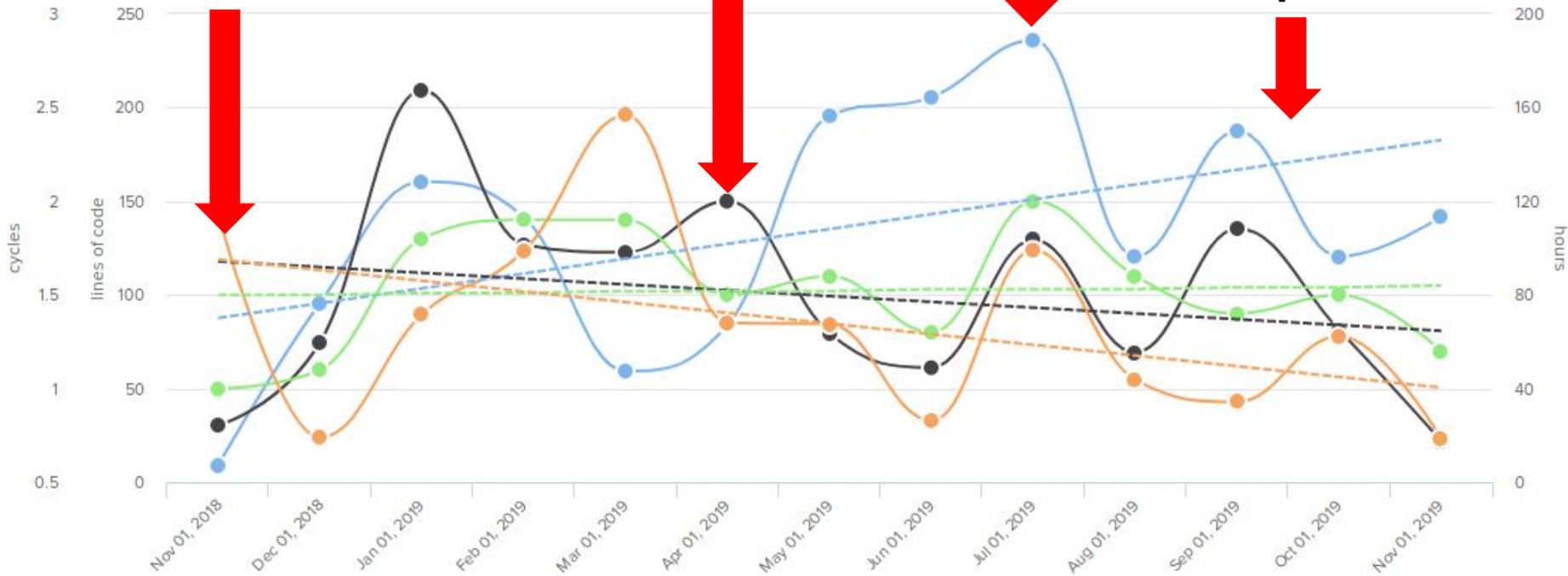
**Segunda
onda**

**Migrações
para Redux**

**Novo
patamar**

- Review Cycles
- Review Speed
- Time to Merge
- PR Sizes

Show TREND



Fonte: Velocity

Conclusão

- Menos código significa mais tempo para escrever com qualidade
- Board Kanban mais fluído
- Reviews mais rápidos e simples
- Entregas mais ágeis
- Revert menor em caso de incidente
- Devs mais motivados

Throughput: estável em 6 cards
Cycle Time: < 8 dias

Obrigado

LUIS FERNANDO MURARA

Software Engineer at Resultados Digitais



murara@gmail.com

Linkedin: <https://linkedin.com/in/murara>

Github: [@murara](https://github.com/murara)

Facebook: [@SirMurara](https://facebook.com/SirMurara)



We're hiring!