

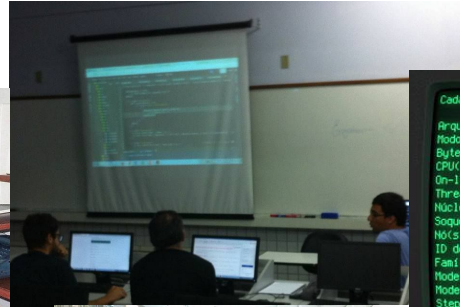
Transição de On-premises para Cloud na AWS e Desafios Culturais Corporativos

Cristian M. & Anna K.

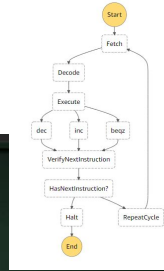




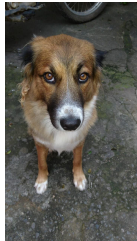
Sobre o Cristian....o que gosta, pensa?



```
Cada 0,1s: Iscpu
Arquitetura:           x86_64
Modo(s) operacional da CPU: 32-bit, 64-bit
Byte Order:           Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Thread(s) per núcleo: 1
Núcleo(s) por soquete: 4
Soquete(s):            1
Número de NUMA:        1
ID de fornecedor:      GenuineIntel
Família da CPU:         68
Modelo:                Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz
Model name:             3
Step:                   3599.008
CPU MHz:                3700.000
CPU max MHz:            800.000
CPU min MHz:            6585.10
BogoMIPS:               6585.10
```



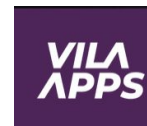
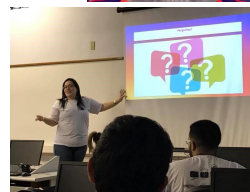
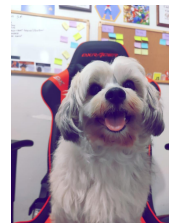
UNIVERSIDADE
VILA VELHA
FACULDADE VILAVELHA



Laradock



Sobre a Anna... o que gosta, pensa...



A cultura corporativa traz um grande vilão: as práticas de waterfall



Por que falar de waterfall ainda hoje? Ainda levamos os resquícios dos antigos modelos produtivos da 2ª Revolução Industrial



Divisão de tarefas:
cada um faz função
específica



X



Modelo Fordista

Repetições exaustivas.
Resultados vistos “só
olhando o que estou
fazendo”



Paralelo com a TI: muito trabalho manual repetitivo. Sem ideia de um todo. Não sobra muito tempo para qualificação e diversificação profissional.



O Fracasso do Fordismo



General Motors



A Ford era rígida. Pouca customização. Somente “carros pretos”. Pouca inovação. Havia esteiras....mas modelo produtivo engessado. Todos os carros eram iguais.



Surgimento da 3ª Rev.
Industrial que é o

Modelo Toyotista

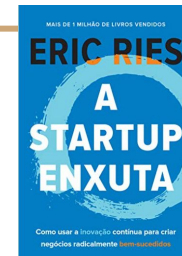
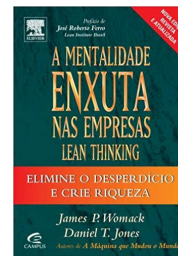
Modelo de Produção por demanda, baseado em Small Batches ou fluxo de uma peça só. Lembrando que a Toyota superou a Ford na década de 1960 com essa abordagem



Nos livros indicados tem um relato de um experimento de um pai com suas duas filhas tratando a forma de fechar envelopes.



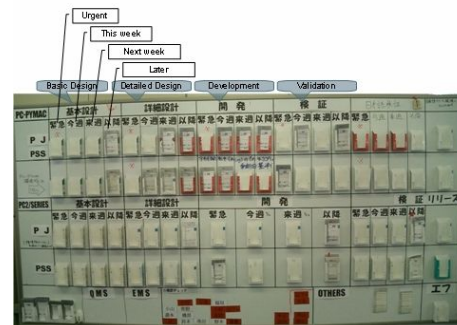
A cultura corporativa traz um grande vilão: as práticas de waterfall. Dizemos que são práticas “ineficientes” do experimento. Resquícios de Ford....



Small Batch

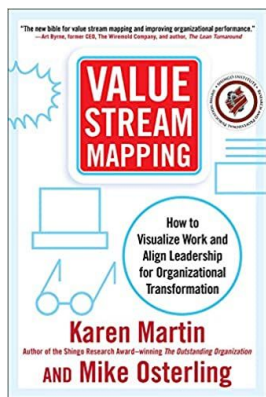


Large Batch



Teoria básica da Manufatura LEAN

DevOps se fundamenta no Lean. O Lean teve origem no Japão por Taichii Ohno (Toyota)



Temos que trazer a ideia do Fluxo de Valor. O que é? Martin & Osterling diz....

Sequência de atividades para entregar um pedido do cliente

Buscamos o enfoque em pequenas entregas

Envolve evitar Retrabalho. É a maior fonte de custos de tempo e pessoal.



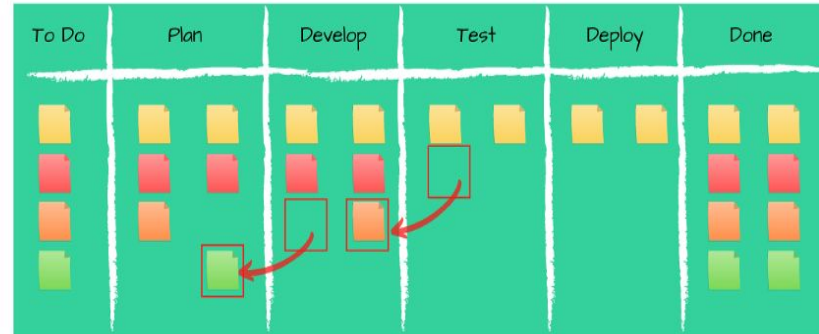
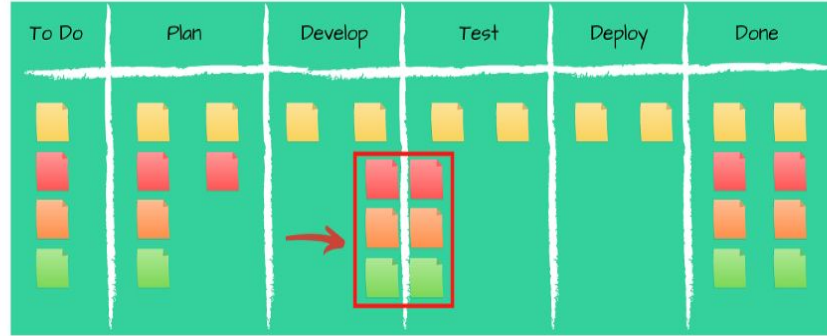
Busca-se a Redução do WIP (Trabalho em Progresso)

Por que você deve limitar o trabalho em progresso? - WIP

- Sistema Empurrado (Produzir máximo possível e entregar em grandes lotes)
- Sistema Puxado (Puxam trabalho quando é capaz de realizar. Ritmo sustentável)

Limitar WIP → Ajuda a encontrar gargalos antes que se tornem impedimentos

Limitar WIP → Traz valor ao Kanban → Time num ritmo de entrega ótimo, balanceado



Entrada: **Hipótese comercial** que entrega valor ao cliente



Início do Fluxo: quando o **desenvolvimento** é iniciado.

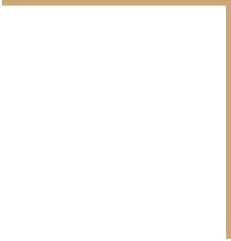


Seguimos com **processo ágil** ou iterativo



Transformação de ideias em User Stories → concretizados pela implementação em **código/aplicação**

Como funciona o chamado “Fluxo de valor em TI”? Trazemos o fluxo de maneira mais clara aqui.



Como o valor é gerado?



Quando os serviços estão funcionando em produção

Lembrando: Hotfix x
(Patch, Coldfix) →
correções em produção



Lembrando: Bugfix →
correções em desenvolv.

Não apenas rápido, mas entregando sem causar interrupções e caos



Paradas de
Serviço



Falhas de
Segurança

GESTÃO DE
MUDANÇAS/RISC
O/CORREÇÕES

A relação LEAN com Dev + Ops

Pode ser explicada da seguinte forma

1ª fase: Design &
Desenvolvimento (Dev)

↓
Semelhante ao
Desenvolvimento LEAN de
Produtos

↓
Etapa criativa, variável, incerta. Tempo de
processo muito variável.

2ª fase: Testes e Operações
(Ops)

↓
Semelhante a Manufatura
LEAN

↓
Previsível, mecanicista. Tempos de
execução curtos.

Cenário Ruim de Implementação da 1ª e 2ª fase

Excesso de centralização e cadeia de comandos para aprovar mudanças

1ª fase: Design & Desenvolvimento (Dev)



Waterfall de lote grande

Estarmos sujeitos a fluxo lento e muitas correções de alto impacto

Ferramentas de Infra/Dev pouco auto-servidas. Pouca autonomia

Excesso de terceirização. “Tarefas adiadas para outros setores”. Geração de dívidas técnicas.

Abertura de chamados para muitas atividades que poderiam ser auto-servidas

2ª fase: Testes e Operações (Ops)

Executadas de forma sequencial após entrega do lote grande



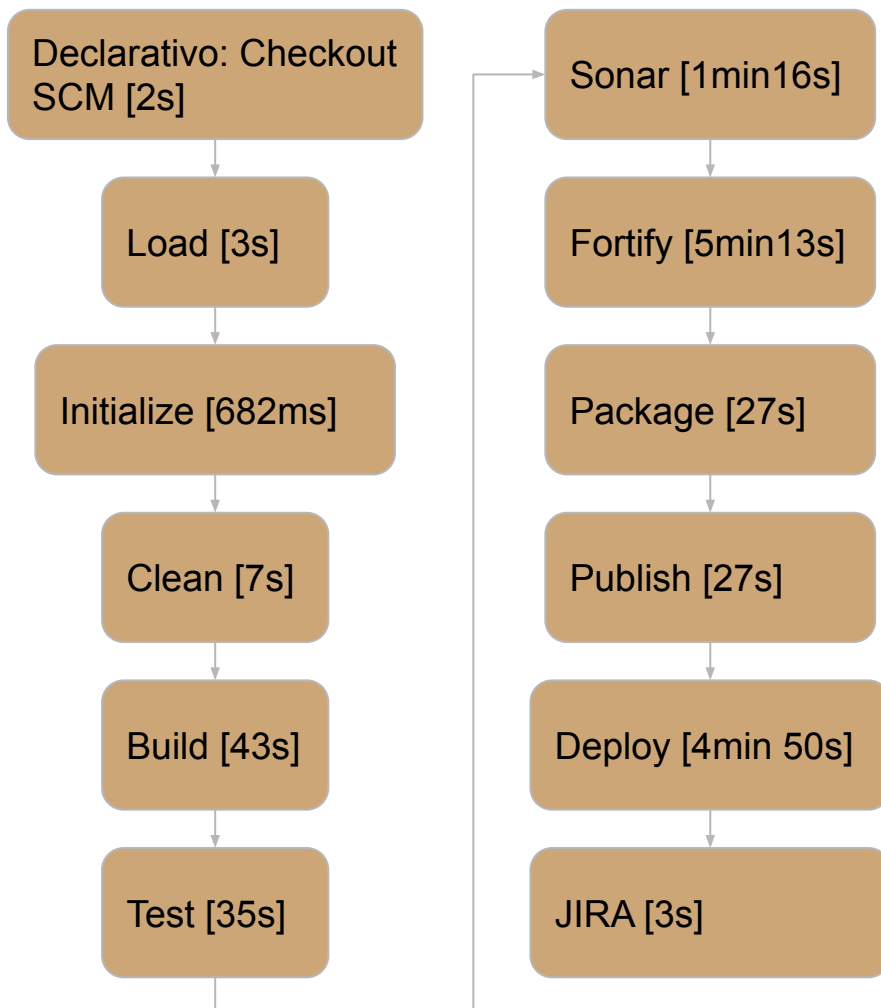
Atrasos nos deploys e acúmulo de tarefas a fazer ou a consertar muito tardiamente

Pipeline de CI/CD típico de cenários ruins (On-premises)



Jenkins

TEMPO TOTAL DA
RELEASE = ~13 min



Pipeline Sequencial

Totalmente
waterfall

Modelo Waterfall é
PANDÊMICO em
grandes organizações



OPENSIFT

Descrição das Steps (On-premises)

Declarativo: Checkout SCM
Step que faz o checkout da esteira Jenkins do repo Git

Load
Carrega credenciais e parâmetros utilizados pela esteira Jenkins

Initialize
Seta valores de variáveis e cria pastas para o pacote da aplicação e o RPM

Clean
Limpa o cache do output do projeto para nova compilação

Build
Realiza o build e gera o pacote da aplicação

Test
Efetua os testes unitários e integração da aplicação descrito na classe de testes

Sonar
Passa pelo sonar para verificar as métrica de qualidade de código

Fortify
Passa pelo fortify para avaliação de vulnerabilidades de seguranças do código

Package
Empacotamento

Publish
Publicação do pacote no artifactory



OPENSIFT

Deploy
Realiza a publicação no ambiente de desenvolvimento

JIRA
Gera card no JIRA para evolução em outros ambientes via esteira de CD

Cenário Ideal de Implementação da 1ª e 2ª fase

1ª fase: Design & Desenvolvimento (Dev)



Ágil/iterativo com Small Batches



Fluxo rápido e alta qualidade

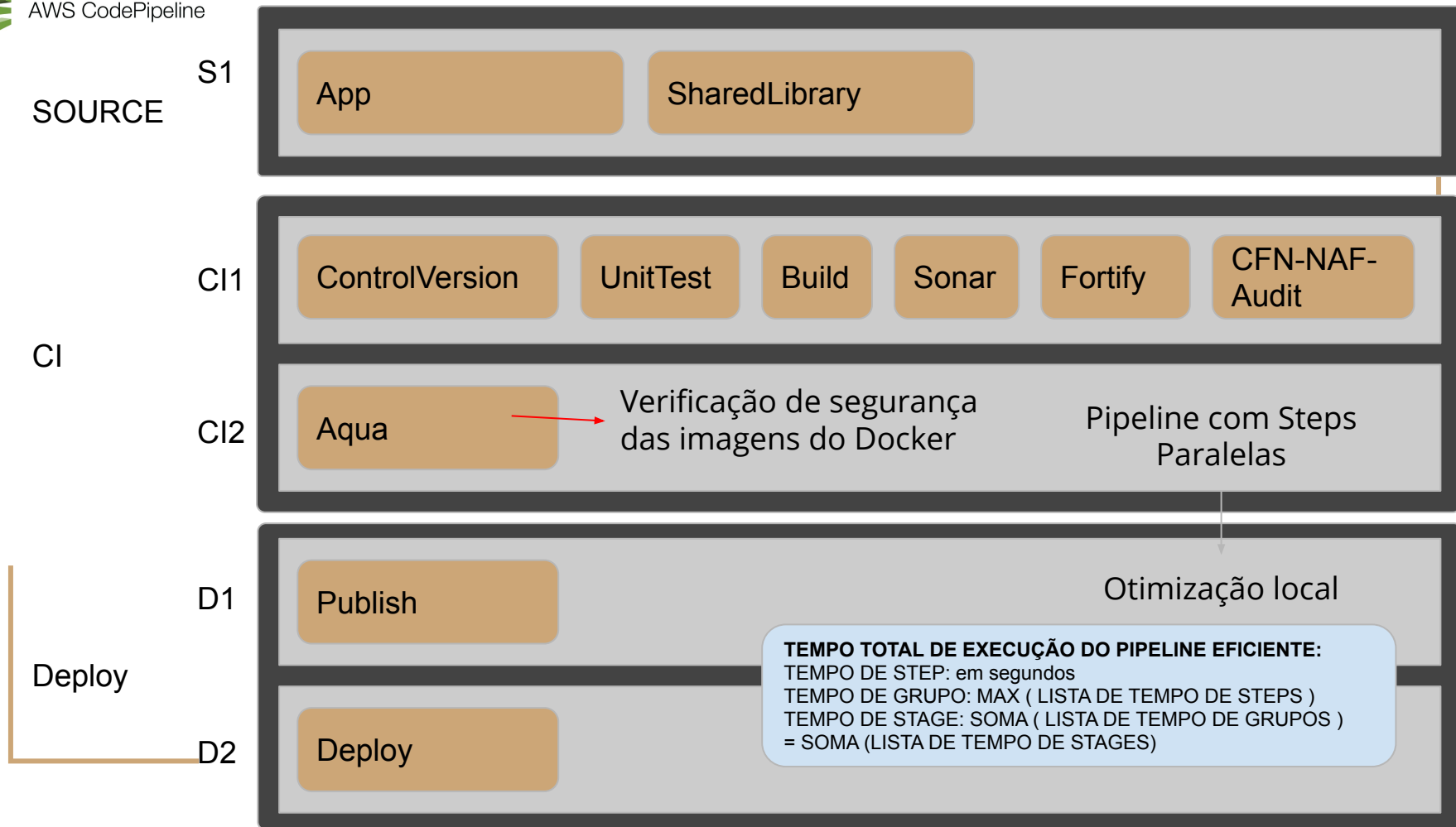
2ª fase: Testes e Operações (Ops)



Executadas simultaneamente com a 1ª fase



Deploys mais antecipados, melhor previsibilidade da entrega e atendimento do fluxo valor





Para DevOps funcionar, precisamos seguir o cenário ideal.
Apenas colocar “pipelines” não traz satisfação ao cliente.
Prova disso: Ford usava esteiras. Mesmo assim não trouxe
elementos competitivos.



Medida de Performance para traçar o futuro de uma empresa de tecnologia



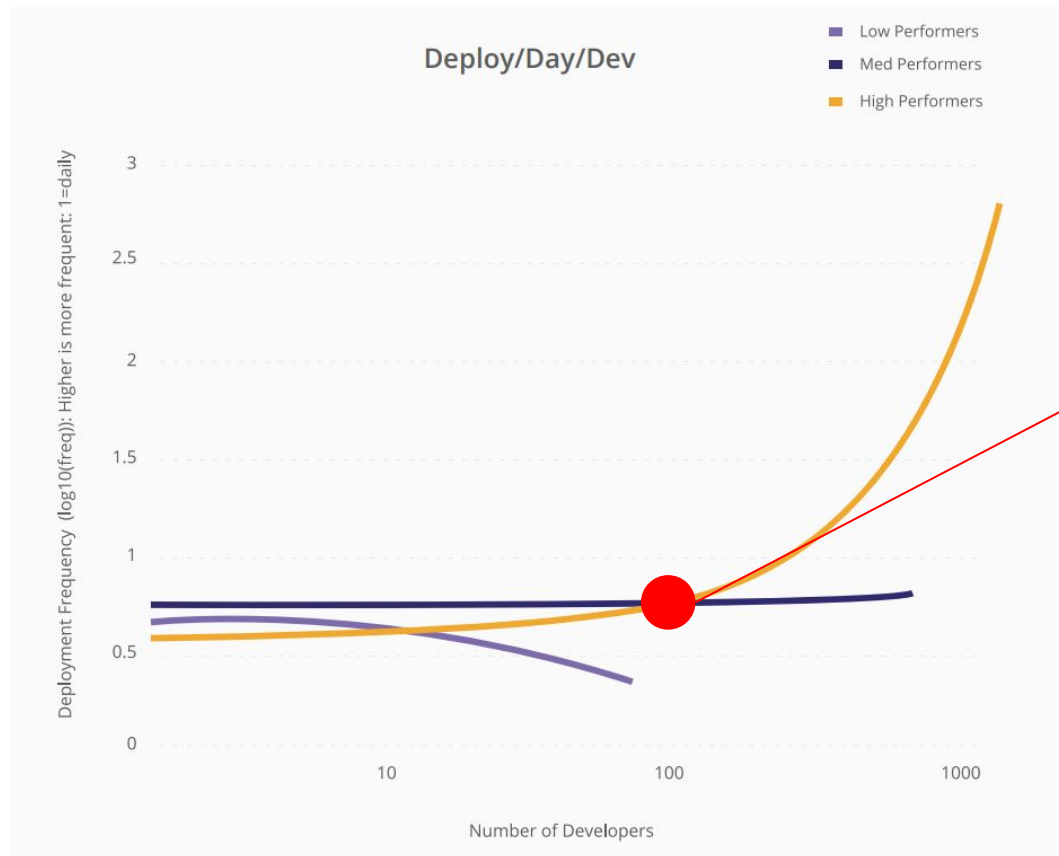
A seguir podemos encontrar uma pesquisa feita pela State of DevOps em 2015. Essa pesquisa traz uma estatística interessante...



Temos hoje um gráfico com comportamento matemático de empresas que vão “patinar”, outras que irão “inovar pouco ou nada” e outras que não irão inovar a médio e longo prazo. Isso é uma constatação impressionante.



Gráfico de Implementações por Dia vs. Número de desenvolvedores



Segundo o Livro “DevOps Manual” (KIM, Gene et al.)

- Empresas de alto desempenho
→ A partir de 100 devs → Já conseguem superar a frequência de entrega de empresas médias e de baixa performance
- Empresas de média performance
→ com mais pessoas, não aumentaram a capacidade de entregar mais. Comportamento linear.
- Empresas de baixa performance
→ com mais pessoas, é reduzida a capacidade de entregar.

As três métricas para medir desempenho do fluxo de valor

Tempo de Processamento
(TP) (tempo de tarefa)

Tempo de Execução (TE)

Métrica de Retrabalho

O Cliente só conhece o Tempo de Execução

TE = Tempo do card na fila + Tempo para
iniciar e terminar o card

FOCO NA RELAÇÃO ENTRE TEMPO DE
PROCESSAMENTO E EXECUÇÃO

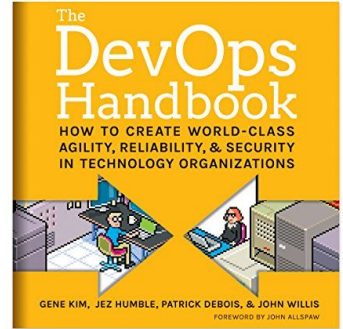
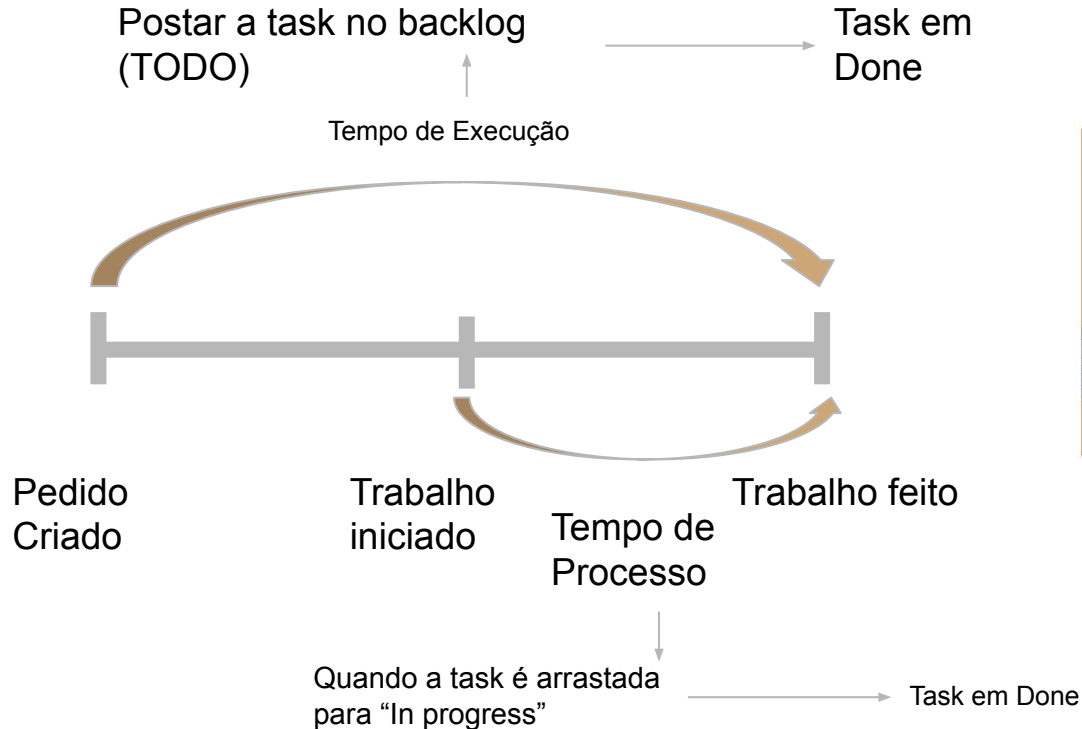
TE curto → exige redução
de tarefas nas filas de
Backlog

TP = tempo para
terminar um card que
está em “in progress”

Proporção entre TP / TE

TP curto → melhorias das
capacidades de execução →
Liberação mais rápida para pegar
novas tarefas e assim reduzir tarefas
das filas de Backlog

Gráfico de Tempo de Execução x Tempo de Processo



Impactos positivos da utilização de Small Batches

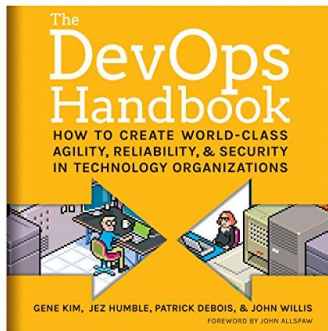
Aumento da interação do time, maior disseminação do conhecimento, rápida identificação e solução de problemas de qualidade, Redução de quantidade de defeitos encontrados,

Maior engajamento do dia nas cerimônias Scrum, aumento das entregas de sucesso ao cliente e empoderamento das pessoas com relação às suas responsabilidades no sucesso da Sprint.



Como sabemos que tipo de performance temos da nossa equipe/empresa? Se ela é baixa, média ou alta.

Empresas de média/baixa performance tem um sério problema com RETRABALHO (3ª MÉTRICA DO FLUXO DE VALOR)



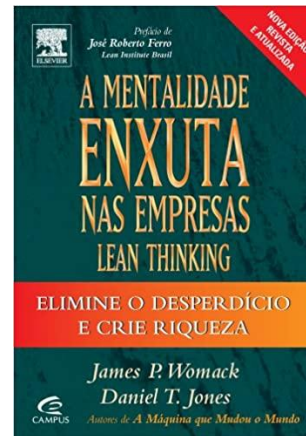
Então, como podemos medir?



$\text{RETRABALHO} = \% \text{ de Conclusão/Precisão}$



Métrica que reflete a qualidade de saída de cada etapa no fluxo de valor



Como começar HOJE o Small Batches?

Tarefa para um colaborador OU
Pair-programming no máximo



Podemos delegar na ferramenta
(JIRA/Trello) para 1 usuário OU no máximo
2

Sprint Backlog com mais atividades precisas
e bem estimadas

Tarefa que sai do Backlog para Doing precisa
ser concluída em 1 dia



Algo que estamos buscando implementar



Isso implica em dirimir melhor as tarefas....Não aceitamos
mais tarefas do tipo “Criar API”



Buscamos investir em refinamentos técnicos

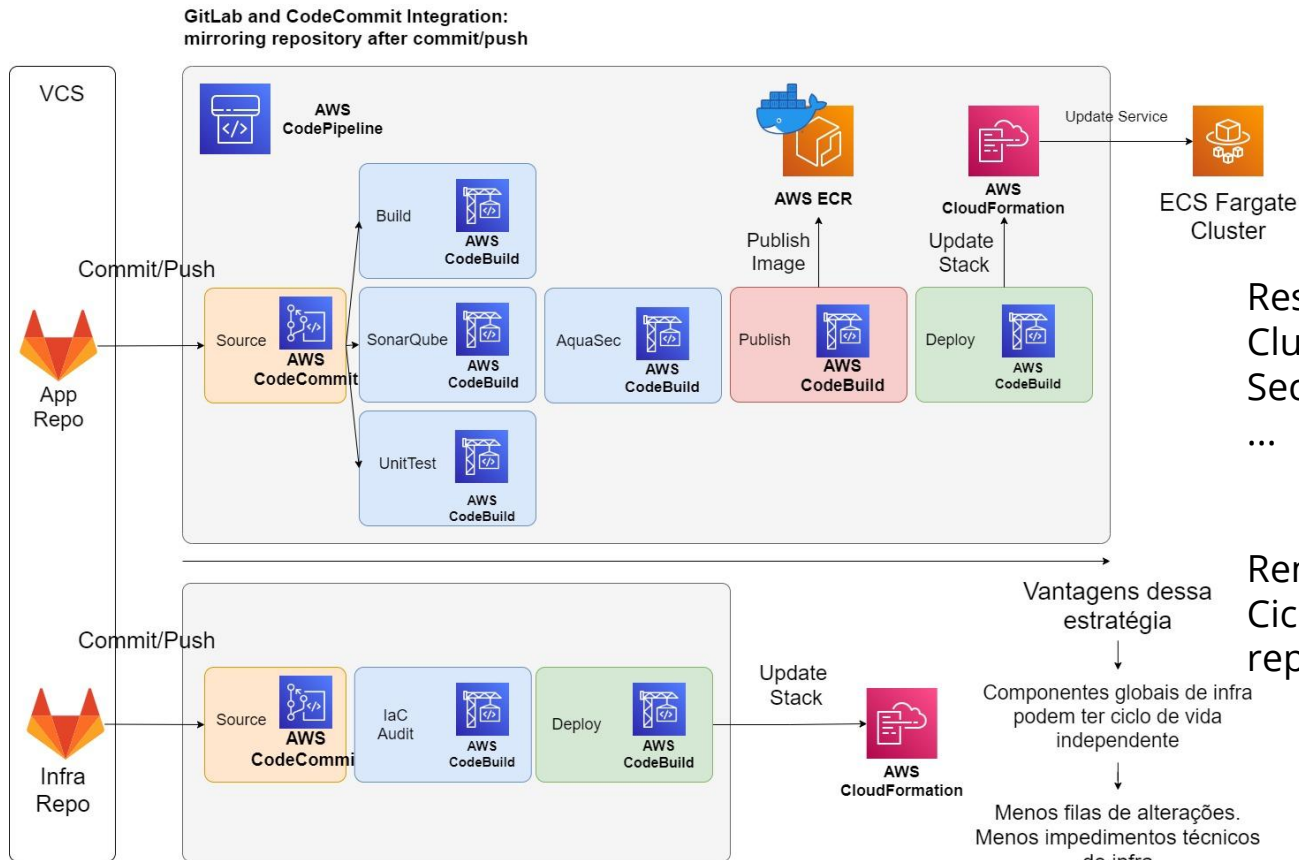


Separação do código de aplicação e infra;
Sugestão de Arquitetura de Pipelines de CI/CD na
AWS

Deploy de Infra/App Multicíclica

Ciclo Rápido

Ciclo Longo



Removemos para
Ciclo Longo (outro
repositório)

Vantagens dessa
estratégia

Componentes globais de infra
podem ter ciclo de vida
independente

Menos filas de alterações.
Menos impedimentos técnicos
de infra

Reuso de alguns recursos de
infra mais "globais" como
SecurityGroups entre outras
infras mais abrangentes e
infras de apps

Desvantagens

- Maisinfras de pipelines
para gerenciar
- Utilização de Parâmetros ou
Importação de Identificadores de
resources muitas vezes baseado em
nomes



Apêndice: Sistema de Melhoria Kata para promover rotinas de melhoria



Rotina de 4 etapas....

Identificar desafios e mover a equipe para atingi-los

Precisamos saber onde estamos

Criar uma condição-objetivo

Mover entre Condição-atual e
Condição-objetivo



Perguntas???



Nossos contatos

Cristian

✉ cristianc.mello@gmail.com

in <https://www.linkedin.com/in/cristian-mello/>

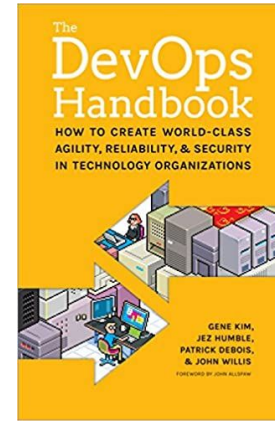
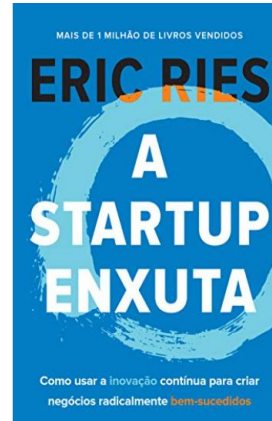
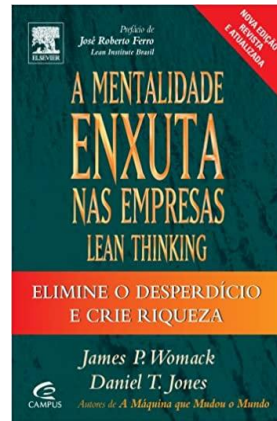
🐙 <https://github.com/cristiancmello>

Anna Karinane

✉ annakarine@gmail.com

in <https://www.linkedin.com/in/anna-karinane-silva-40101a174/>

Referências



Agradecimientos



THE
DEVELOPER'S
CONFERENCE