

The background of the slide is a teal color with a white network pattern. This pattern consists of numerous small dots connected by thin, white lines, creating a complex web-like structure that covers the entire background.

igti

—

Devops

Capítulo 2. Configurações

Prof. Thiago Chierici



Aula 2.1. As 5 diretivas de SCM mais importantes para ter sucesso com o Devops



- ❑ Definição de SCM.
- ❑ Principais diretivas de SCM para ter sucesso com Devops.



- “The most powerful tool we have as developers is automation.”

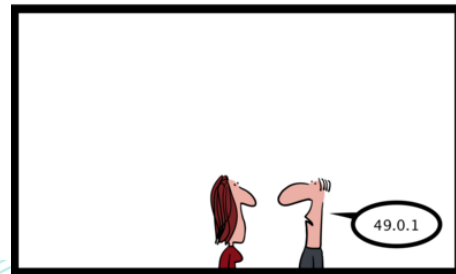
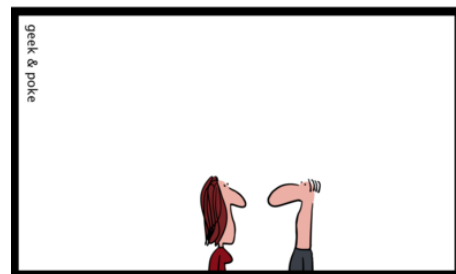
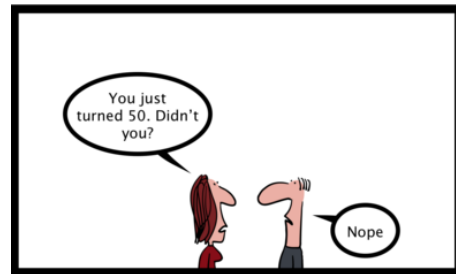
Scott Hanselma.



Source control management

- Processo ou sistema que permite a gestão das mudanças em itens de configuração.
 - Normalmente cada versão tem um identificador único.

igti
MINOR FIXES



SEMANTIC VERSIONING

- Mantenha tudo no controle de versão.



- Integre com frequência.

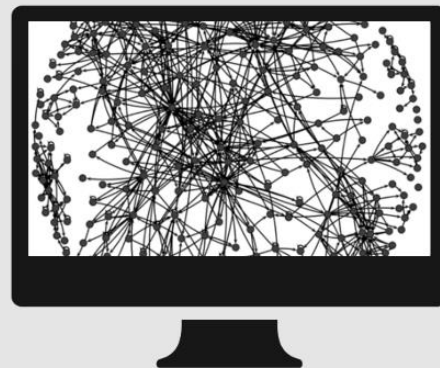


- Descreva bem as mensagens.

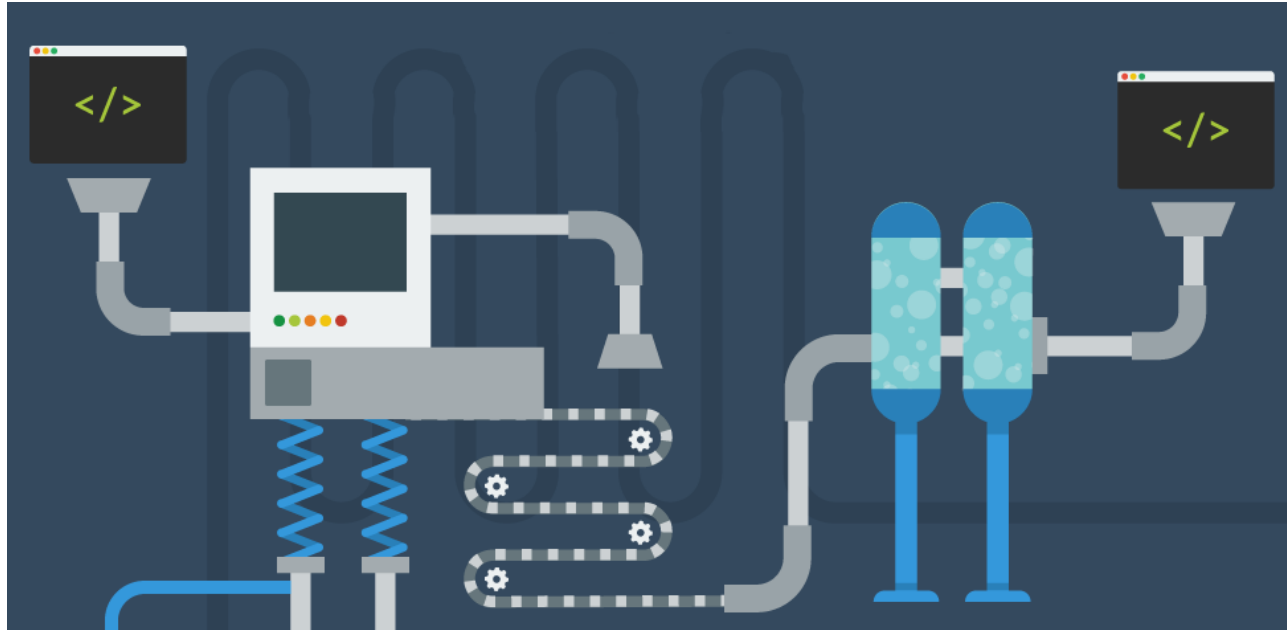


- Gerencie bem as dependências.

software
dependencies



- Gerencie configurações e ambientes.



- ☑ **Mais importante que** a escolha das **ferramentas** de SCM, é o **mindset** de automação.
- ☑ Diretivas baseadas em casos de sucesso, podem contribuir na busca pela melhoria dos seus processos.



- ❑ Principais modelos de versionamento de código.





Aula 2.2. Como versionamos códigos



- ❑ Técnicas mais comuns para versionamento de código.
 - Fácil para quem já desenvolve com estratégia de branches.
- ❑ O que mais precisamos?
- ❑ Se não sou técnico, precisarei saber algo sobre isso?



- “It is not the strongest of the species (processos, times, produtos) that survive, nor the most intelligent, but the one most responsive to change.”

Charles Darwin



O que são branches?



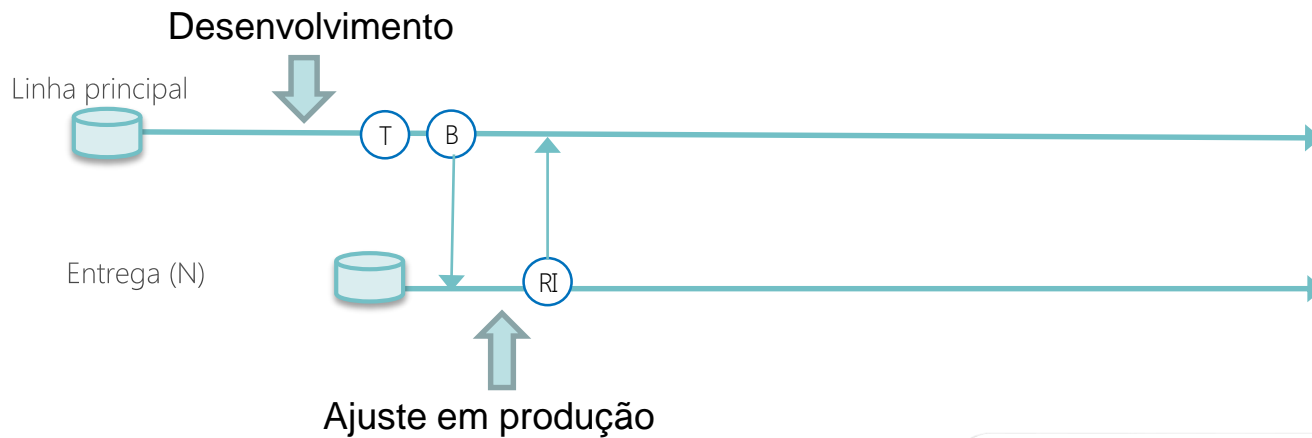
- “Ramificação, no controle de revisão e gerenciamento de configuração de software, é a duplicação de um objeto sob controle de revisão para que as modificações possam ocorrer em paralelo ao longo de ambas as ramificações.”

Wiki



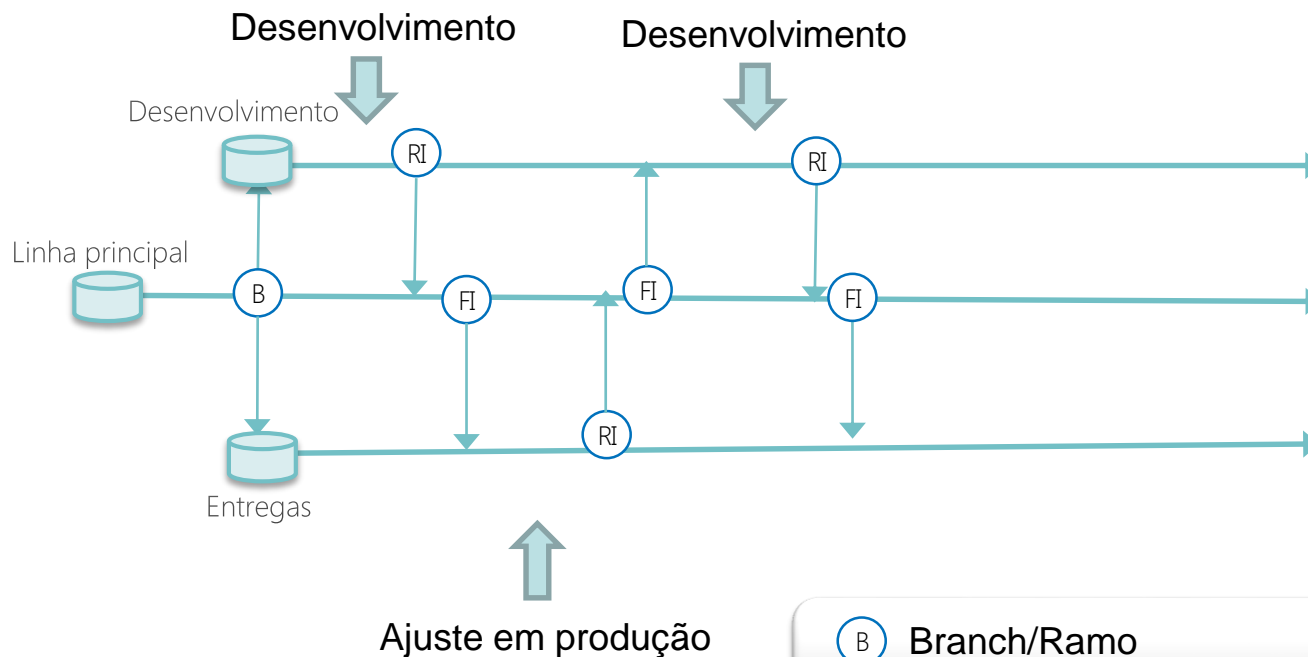
- Alguns exemplos de como trabalhar com branches:
 - Desenvolvimento clássico.
 - Entregas sequenciais.
 - Entregas paralelas.





- T Tag / Marcação
- B Branch / Ramo
- RI Reverse Integration

Entregas sequenciais



B

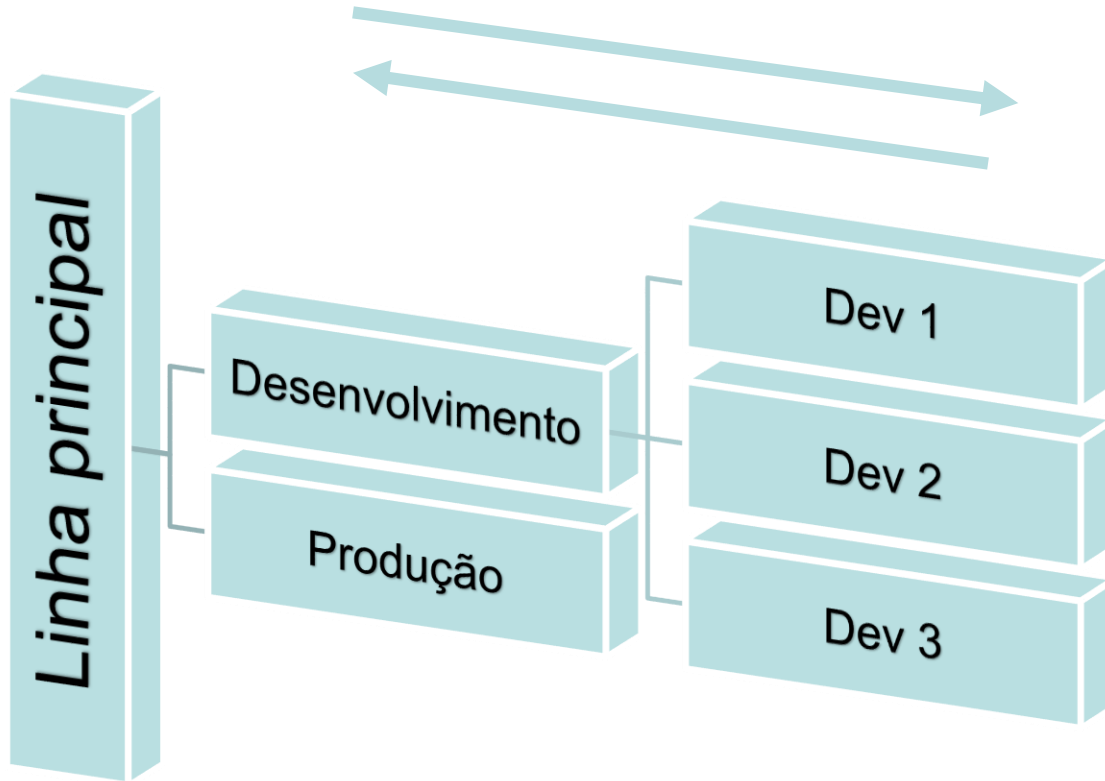
Branch/Ramo

RI

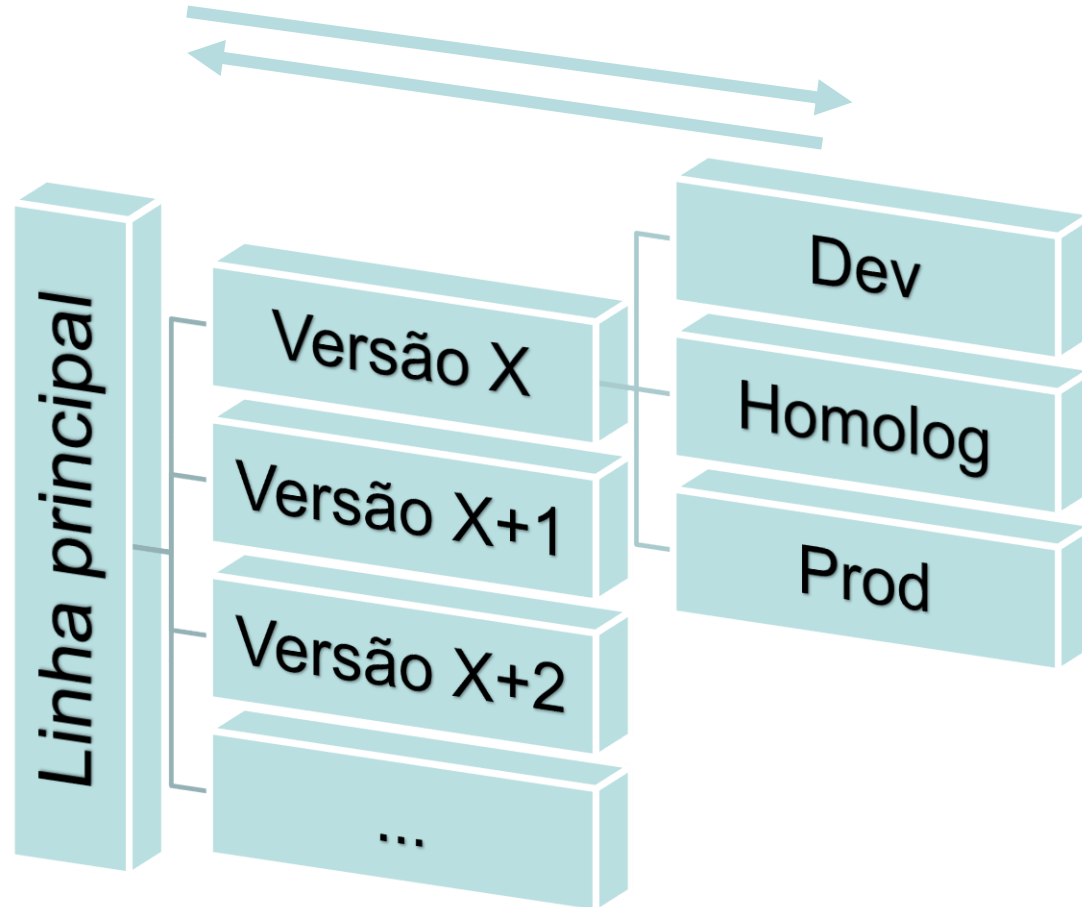
FI

Reverse/Forward Integration

Homologações paralelas



Produto com múltiplas versões



- ☑ Diferente projetos -> Diferentes SCMs.
- ☑ Escolha o SCM **mais simples** que seja capaz de atender suas **necessidades atuais**.
- ☑ Algumas convenções e ferramentas podem tornar a escolha de alguma dessas estratégias um pouco mais simples.



- ❑ Git flow: uma estratégia concreta para gerenciar suas branches.
- ❑ Na sequência, vamos entender o trunk based development e como escolhemos entre essas duas das principais estratégias atuais.





Aula 2.3. Devops com Gitflow



- ☐ O que é o Gitflow.
- ☐ Processo proposto.
- ☐ Caminho para Devops com Gitflow.



O que é o Git flow

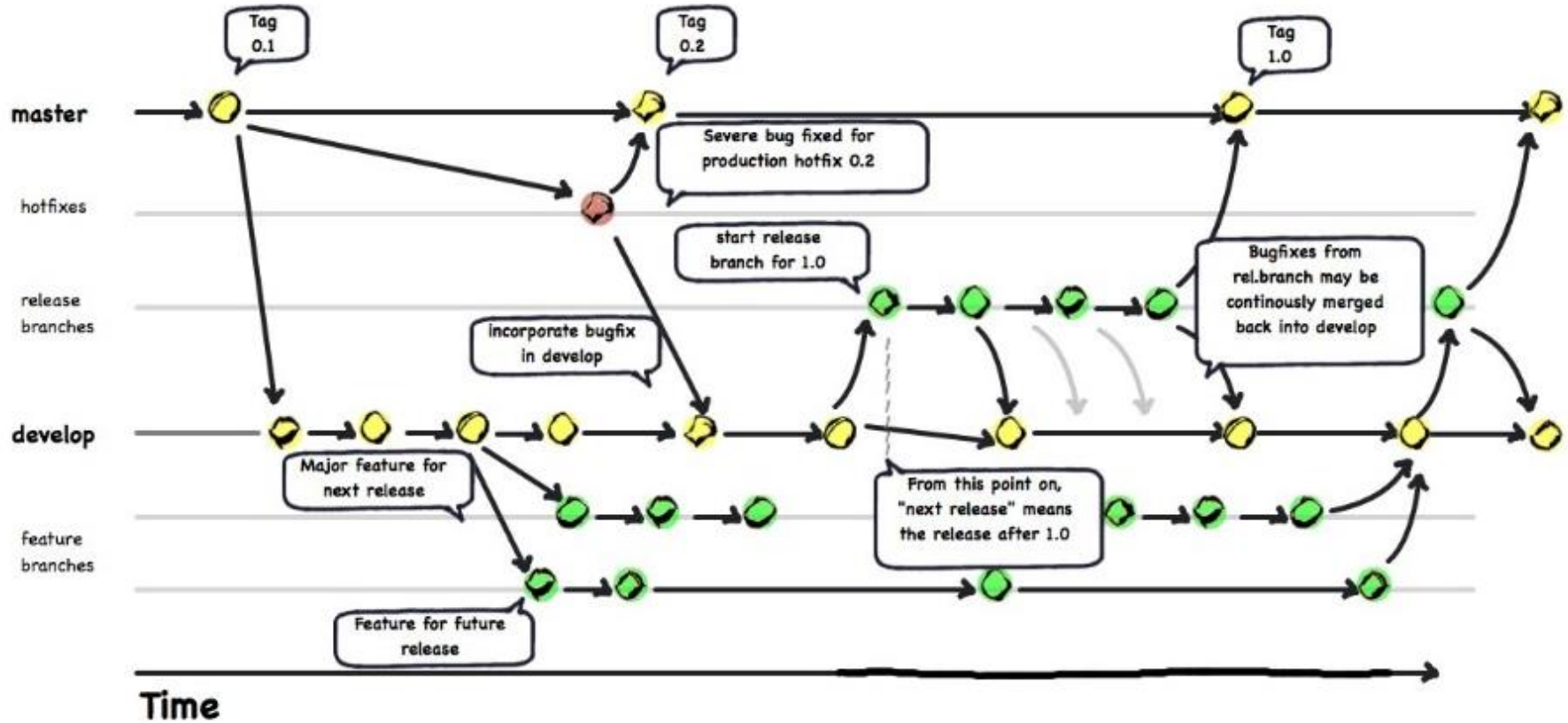


- “... Uma estratégia de branches e release”.
 - <https://jeffkreeftmeijer.com/git-flow/>
- + Lib para auxiliar no uso dos comandos git.



- Master:
 - Código equivalente à produção.
- Develop:
 - Código pronto pra ir pra produção.
- Feature:
 - Código de funcionalidade em desenvolvimento.
- Release:
 - Branch de integração para testar merge para Master.
- Hotfix:
 - Branch para correções emergenciais em produção.





Quando o Git flow ajuda



- Controle nas mãos:
 - Segurança.
 - Qualidade.
- Funciona bem para projetos Open-Source.
- Time com senioridade concentrada.
- Produto já bem estabelecido.



Quando o Git flow complica



- Projetos/produtos começando.
- Em iterações muito curtas/rápidas.
- Com time homogêneo e sênior.



- ☑ Git flow oferece um modelo reconhecido no mercado e que resolve os principais problemas de SCM.
- ☑ Alguns tipos de projetos podem ter dificuldades para se beneficiar desse processo.



- ❑ Uma alternativa ao Git flow e similares.
- ❑ Trunk based development.





Aula 2.4. Trunk based development



- ❑ Como simplificar seu SCM, reduzir retrabalho e acelerar entregas.



Problemas do SCM com branches



- Feature branches com vida longa.
- Branches por ambiente.
- Excesso de branches.
- Muito tempo gasto com merges, testes e atualizações.



- Não precisamos criar branches.
- Branch única, o trunk.
- O código sempre está em um estado que pode ir para produção.



O que conta a favor do trunk-based



- Bom para projetos que estão começando.
- Bom para cenários com iterações muito curtas/rápidas.
- Bom para times de maior senioridade.
- Sempre prontos para o deploy.
- Produtividade do time x produtividade de cada um.



O que vai contra o trunk-based

- Projetos open-source.
- Time de baixa senioridade.
- Produto já bem estabelecido.
- Grandes ou muitos times.



- ☑ Trunk based development pode reduzir retrabalho e acelerar o ritmo de entregas.



- ❑ Técnicas para acelerar drasticamente seu ritmo de entregas.





Aula 2.5. Como implantar código em produção diariamente



Nesta aula



- ❑ Frequência de entregas.
- ❑ Como entregar com alta frequência.



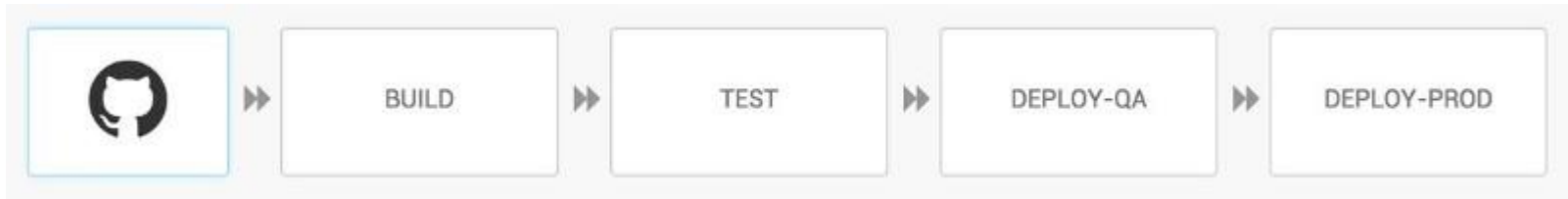
- “Once we accept our limits, we go beyond them.”

Albert Einstein

A cada deploy, temos novas oportunidades de aprendizado e melhoria



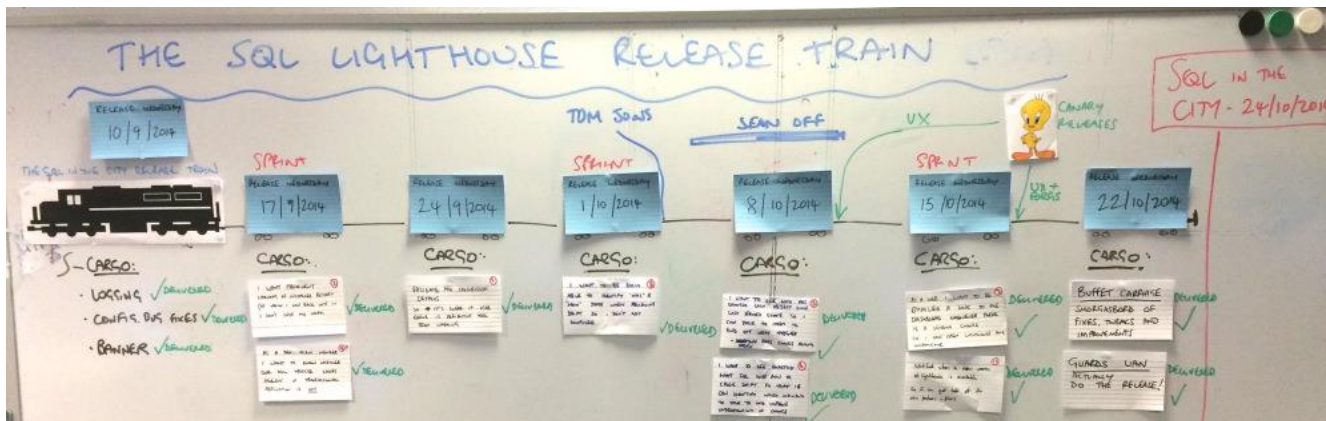
- Facilitado pelo uso de trunk-based development.
- Configurado como uma série de estágios, sempre em direção à produção.
- Jenkins é uma das ferramentas mais utilizadas pra esse tipo de automação.



Release train



- Definição de uma cadência de entrega fixa (alta frequência).
- Tasks são liberadas na medida em que estão prontas.
- Atividades pequenas e independentes, e trunk-based, também podem ajudar aqui.



- Solução complexa que envolve múltiplos release trains.



Ref: https://medium.com/@media_75624/solution

- Para dar certo:
 - Visão integrada do planejamento e priorização de cada frente.
 - Acompanhamento em tempo real do projetado x realizado.
 - Boa gestão de dependências.
 - Acompanhamento dos impedimentos.



- ☑ Aumentar o ritmo de entregas em produção pode trazer vários benefícios diretos e indiretos.
- ☑ Algumas técnicas sugeridas pelo Devops podem aumentar drasticamente a quantidade de entregas que seu time pode fazer.
- ☑ Trunk based + pipeline pode viabilizar um ritmo de entregas acelerado.
- ☑ Release train pode viabilizar um ritmo consistente.



- ❑ Mais um case inspirador com um grande influenciador do mercado.





Aula 2.6. Inspiração – Trunk based com a Red Gate



- ❑ Como a Red Gate vem utilizando trunk based development com sucesso?



- Empresa de desenvolvimento de software.
- Cambridge – UK.
- .NET, SQL e Azure.



Por que amamos o trunk based?



- Redução de algumas dores:
 - Conflitos de merge.
 - PRs abertos por muito tempo.
 - Longas revisões.
- Incentivo a boas práticas.
- Desencoraja maus hábitos.



- Trunk based:
 - Exige a quebra em pequenas unidades de entrega.
- Flags podem ajudar:
 - Para desabilitar uma funcionalidade, você precisará encapsula-la corretamente.
 - Mais um benefício indireto.



- Primeira ferramenta pra isso.
 - Testes/validações automatizados.
- É importante que o feedback seja rápido.
 - Busca constante.



Red x Green build

- Muito importante manter a trunk em perfeito estado.
- CI é crucial para fazer todas as validações necessárias.



- ☑ Apesar da desconfiança dos mais tradicionais, a Redgate mostra que o trunk based development vem se tornando uma importante técnica para dar ainda mais agilidade para fluxos de entrega contínua.



- ❑ Um novo capítulo em que discutiremos sobre a infraestrutura e como ela deve ser gerenciada em um ciclo Devops.

