

ENGENHARIA DE SOFTWARE

Integração contínua



INTEGRAÇÃO CONTÍNUA

- **Prática de fundir componentes com o incremento de software em evolução, uma ou mais vezes ao dia**
 - Comum em métodos ágeis (o termo nasceu com XP)
 - **Teste de integração** deve ser ágil

Difere da prática de construir novas funcionalidades de forma isolada e integrá-las no fim do ciclo de desenvolvimento.

INTEGRAÇÃO CONTÍNUA

- **Mesmo organizações que não usam métodos ágeis podem usar integração contínua**
 - **Estima-se que mais de 50% das desenvolvedoras de software usam integração contínua**

POSSÍVEL ABORDAGEM: TESTE FUMAÇA

1. Componentes de software recém-codificados são integrados em uma nova *build*.
 - A *build* deve incluir todos os arquivos de dados, bibliotecas, módulos reusáveis e componentes necessários para implementar uma ou mais funções do produto.
2. Uma série de testes é criada para expor erros que impedem a *build* de executar corretamente sua função.
 - Busca-se encontrar erros bloqueadores (show-stoppers) que apresentam a mais alta probabilidade de atrasar o cronograma do software.
3. A *build* é integrada a outras *builds*, e o produto inteiro passa diariamente pelo teste fumaça.

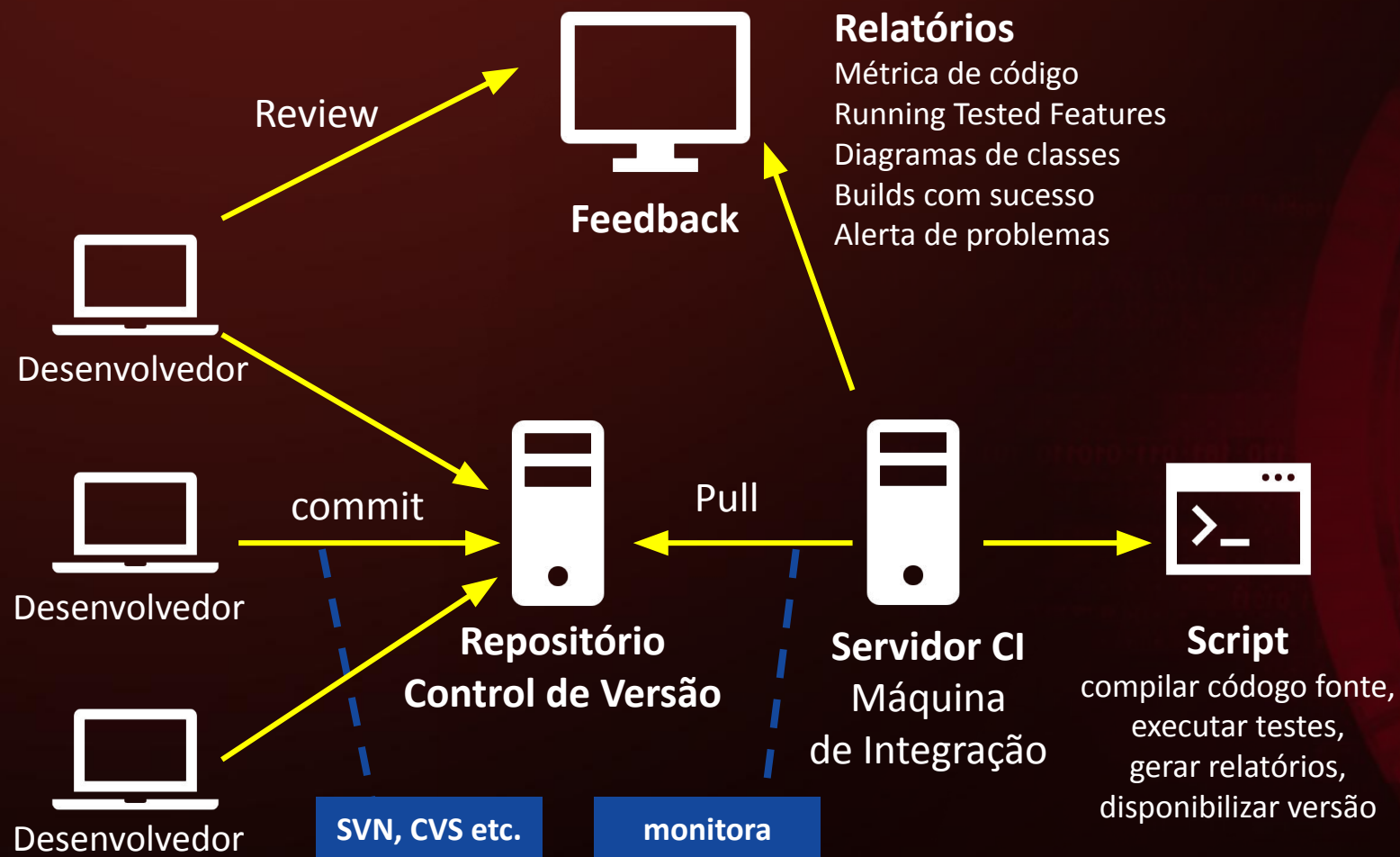
POSSÍVEL ABORDAGEM: TESTE FUMAÇA

- **O teste fumaça exercita o sistema fim-a-fim**
 - Não precisa ser exaustivo, mas ser capaz de expor os principais problemas
 - Deve ser bastante rigoroso, pois ao passar por ele, entende-se que os testes de aceitação/sistema podem ser iniciados

INTEGRAÇÃO CONTÍNUA NA PRÁTICA

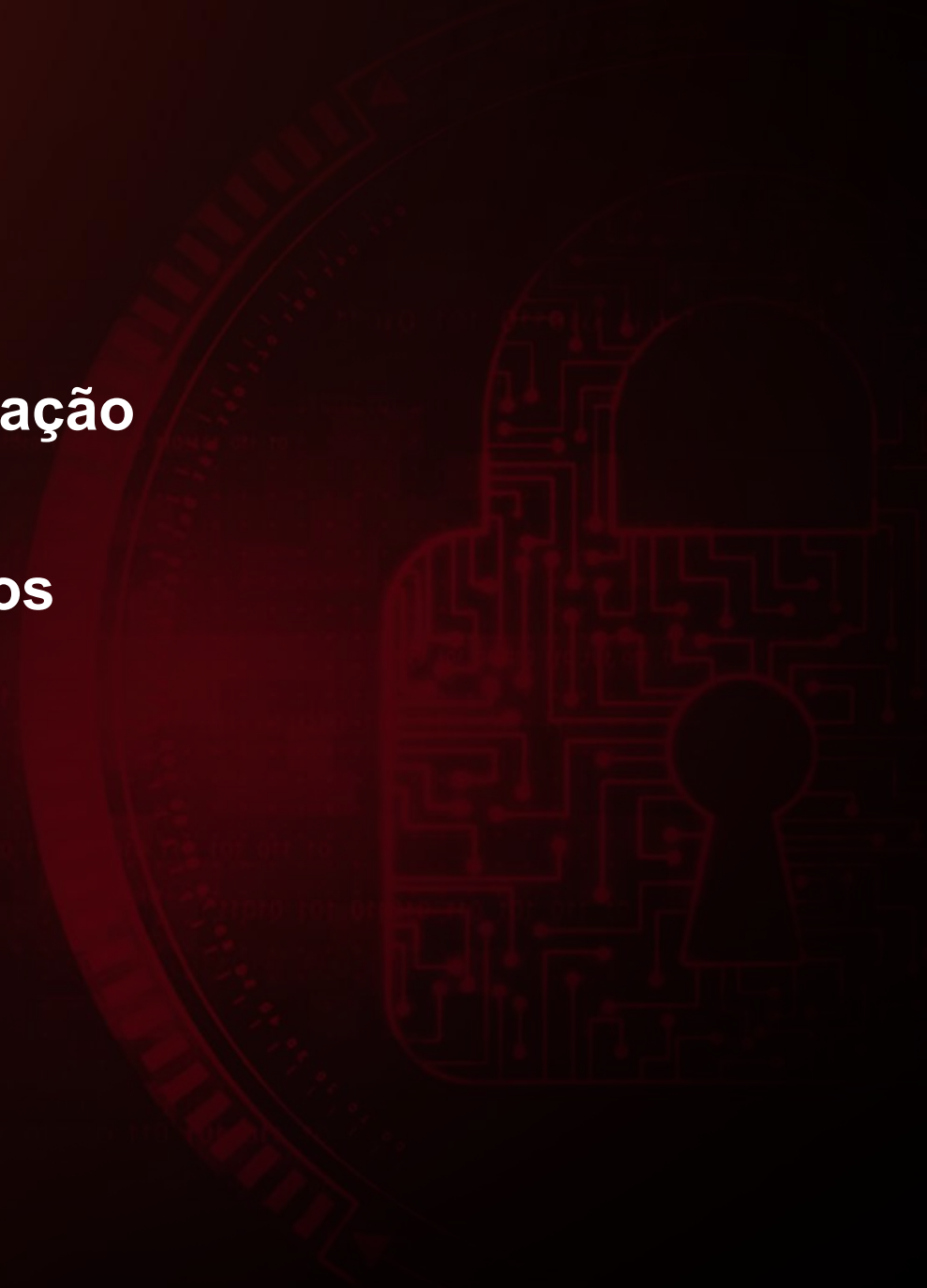
- **Os desenvolvedores realizam *commits* de pequenas atualizações regularmente...**
 - ...e são notificados rapidamente caso suas modificações provoquem falhas no sistema em desenvolvimento
- **A integração contínua deve ser um processo simples, de repetibilidade**
 - Parte do fluxo de trabalho diário, para reduzir custos e corrigir defeitos o mais rápido possível

INTEGRAÇÃO CONTÍNUA NA PRÁTICA



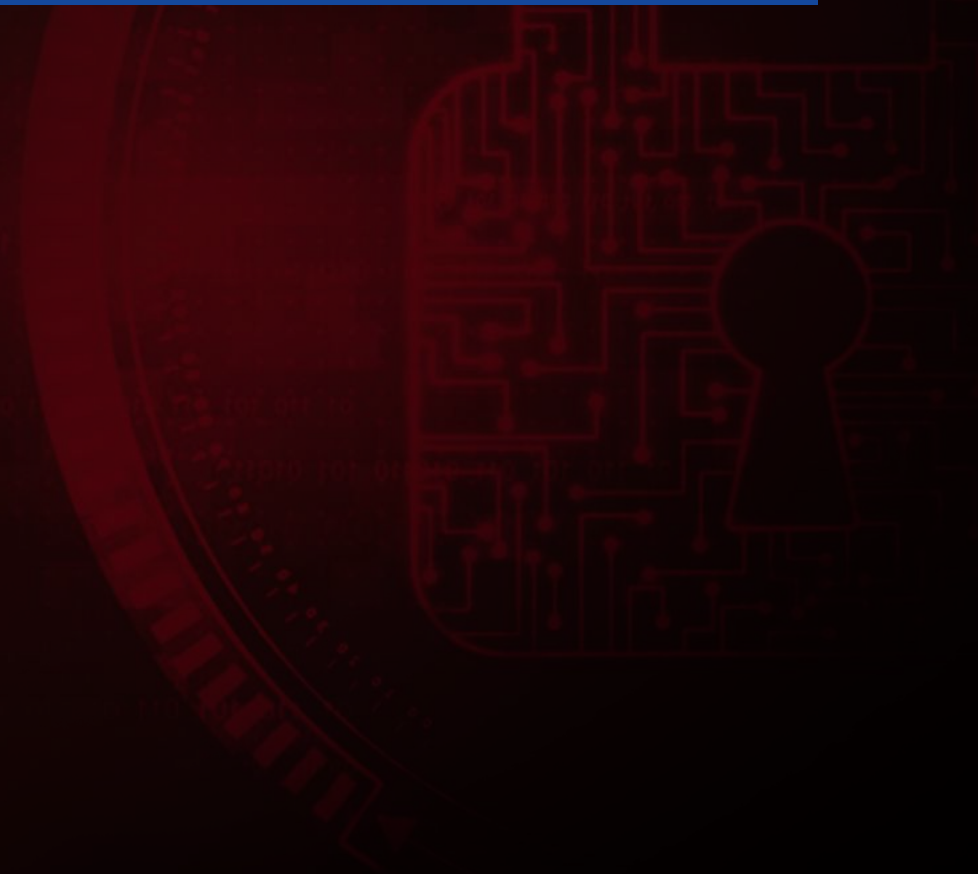
VANTAGENS/BENEFÍCIOS

- **Risco minimizado para o processo de integração**
- **Qualidade melhorada do produto final**
- **Diagnóstico e correção de erros simplificados**
- **Avaliação de progresso facilitada**



AUTOMAÇÃO DOS TESTES

Espera-se que os testes criados sejam automatizados para comporem um conjunto de testes de regressão para as próximas integrações.



FERRAMENTAS DE APOIO À INTEGRAÇÃO CONTÍNUA

- AWS CodePipeline
- Azure Pipelines
- Bitbucket Pipelines
- CircleCI
- CloudBees
- CodeShip
- GitLab CI/CD
- Jenkins/Hudson
- Sauce Labs
- Travis CI

REFERÊNCIAS

- **POLO, R. C. Validação e teste de software. Porto Alegre: SAGAH Educação S.A., 2019.**
- **PRESSMAN, R. S.; MAXIM, B. R. Engenharia de software: uma abordagem profissional. 10a. ed. Porto Alegre: AMGH Editora Ltda, 2021.**