

ARQUITETURA DE SW

EAD - DevOps



Análise e Desenvolvimento de Sistemas

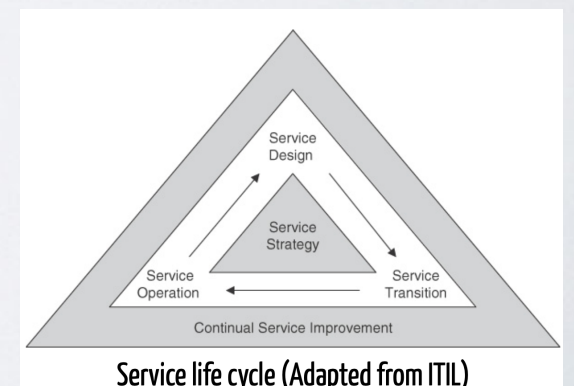
Prof. Ms. Renato Leite
renato@sectras.edu.br

OBJETIVOS

- Tradicional - ITIL
- Conhecer a cultura DevOPS;
- Conceito de DevOPS;
- Monitoramento em DevOPS

ITIL - INFORMATION TECHNOLOGY INFRASTRUCTURE LIBRARY

- Conjunto de melhores práticas que promove a padronização e melhoria contínua dos processos de gerenciamento de serviços de TI.
- Não é preditivo;
- Orientado ao negócio;
- Service Value System
 - Entregar TI é resultado de um sistema integrado composto por elementos estruturantes, como: governança, práticas, melhoria contínua, princípios orientadores.
- Service Value Chain
 - Cadeia de entrega de valor: Negócio, Tecnologia e Atendimento.



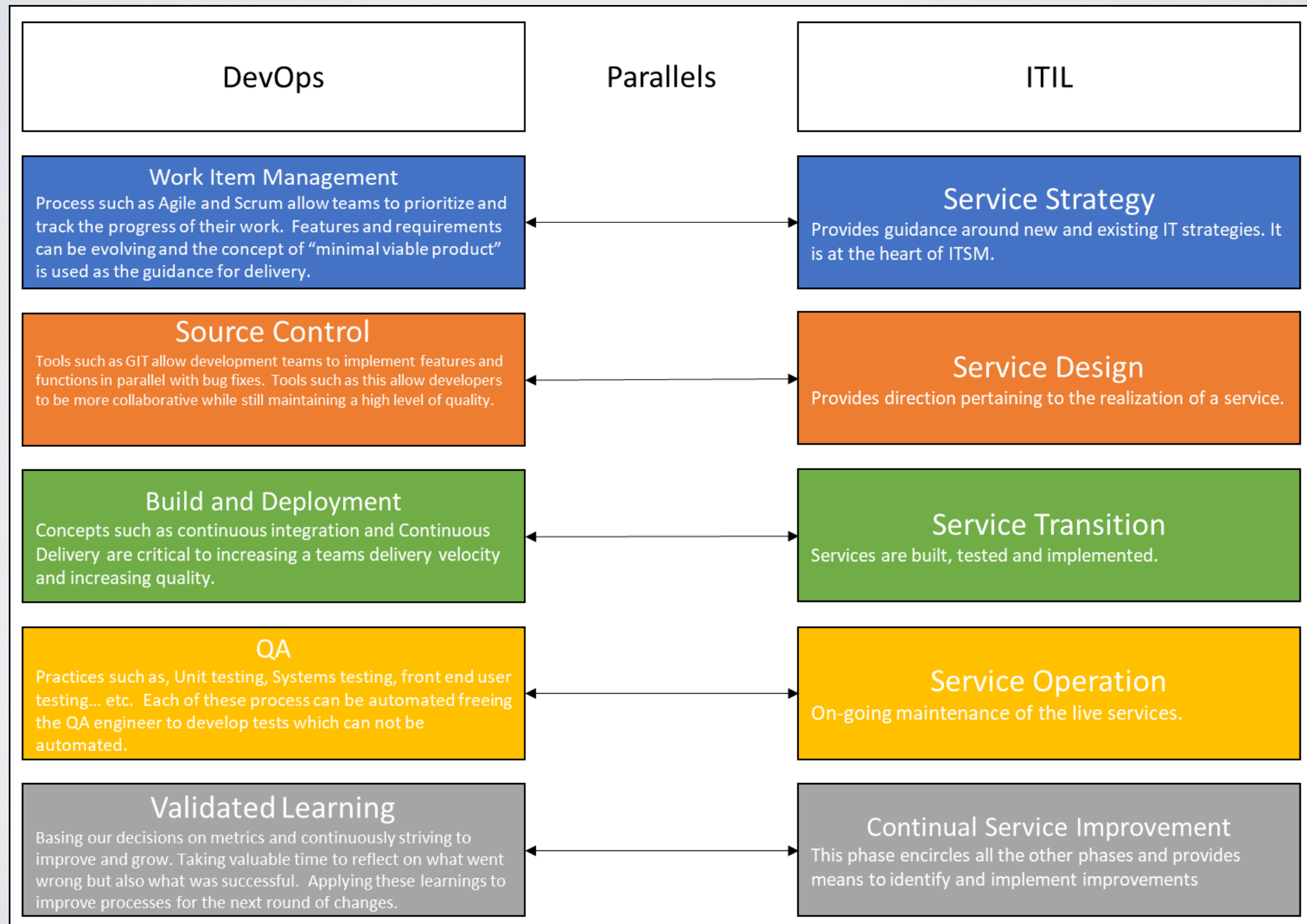
ITIL

- Serve portanto como base ao desenvolvimento de frameworks e estruturas de melhoria contínua no âmbito da tecnologia da informação;
- ITIL4 inclui diversas boas práticas relacionadas a Lean e Agile para se tornar ainda mais aderente ao desenvolvimento usando princípios SOA e Microservices;
- É uma metodologia base que permite a equipes integrar ferramentas e processos em busca da melhor governança e gestão de serviços;

ITIL & DEVOPS

- Se ITIL foca em governança e na entrega do serviço, como uma operação estruturada...
- ... DevOps possui enfoque na automação e agilidade na entrega de software.
- Sendo TI um cubo, ITIL e DevOPS são faces do mesmo e se conectam internamente;

ITIL & DEVOPS



CULTURA DEVOPS

- "A cultura come a estratégia no café da manhã" - Peter Drucker
- Principalmente, DevOPS é sobre comportamento;
- Assim como a arquitetura do sistema está conectada a forma como a empresa organiza seu negócio, DevOPS está intimamente ligado a como a empresa enxerga o atendimento aos seus clientes;
- Devs-on-Phone!!
 - Quem escreve é responsável.

O QUE É?

- “DevOps representa uma mudança cultural e técnica que habilita organizações a desenvolver, entregar e operar software e sistemas com rapidez e segurança.”

Kim, Gene; Humble, Jez; Debois, Patrick; Willis, John.

The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations.

IT Revolution Press, 2016.

1. Colaboração entre equipes interdisciplinares (Dev, Ops, QA, Segurança)
2. Automação de processos: integração contínua (CI), entrega contínua (CD), testes e infraestrutura como código
3. Monitoramento contínuo e feedback rápido
4. Responsabilidade compartilhada pelo ciclo de vida completo do software
5. Melhoria contínua e aprendizado organizacional

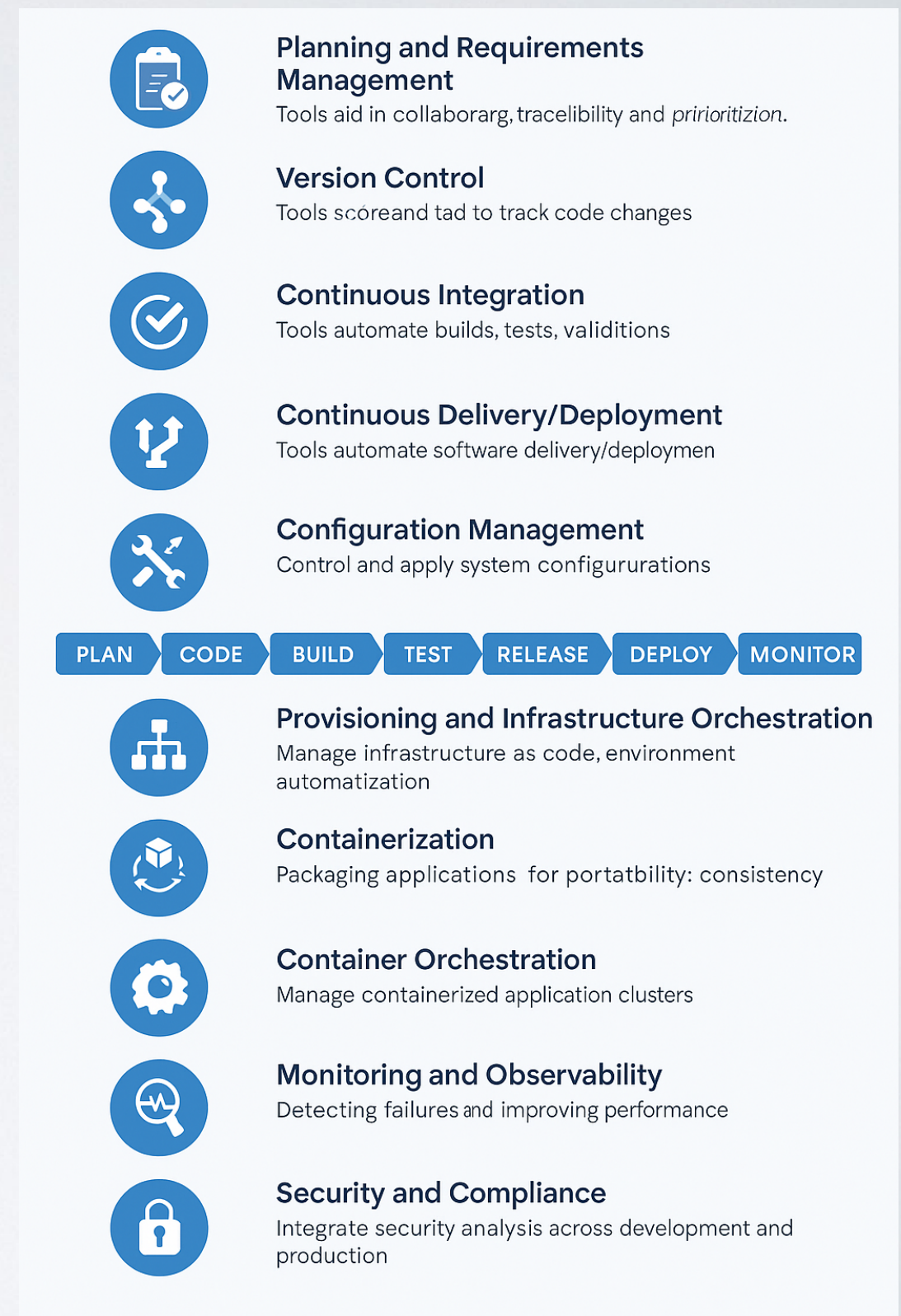
Humble, Jez; Farley, David.

Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation.

Addison-Wesley, 2010.

ENGENHARIA DE SOFTWARE

- DevOPS, como o próprio nome diz, é a conjunção entre Desenvolvimento e Entrega!
- Portanto, se conecta com todas as etapas de qualquer processo de desenvolvimento de software;
- Contudo, se adequa principalmente ao modelo iterativo e incremental;

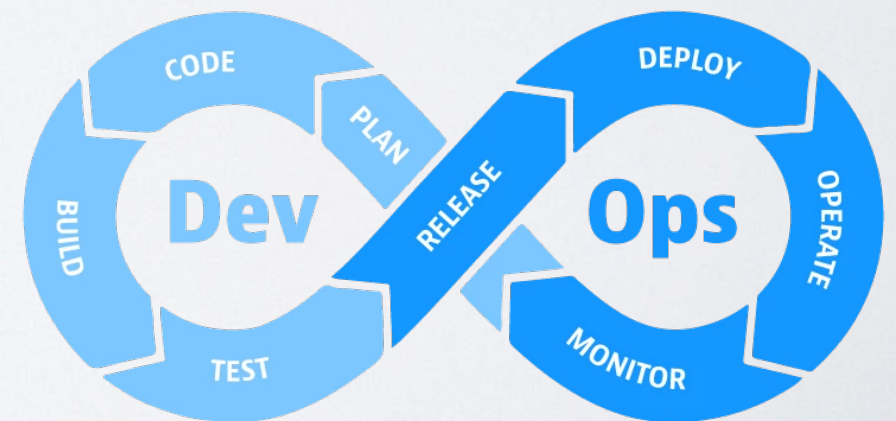


FERRAMENTAS

Categoria	Principais Ferramentas
Planejamento e Gestão de Requisitos	Jira, Trello, Azure Boards, GitHub Projects
Controle de Versão	Git, GitHub, GitLab, Bitbucket, Azure Repos
Integração Contínua (CI)	Jenkins, GitLab CI, CircleCI, Travis CI, Azure Pipelines
Entrega/Implantação Contínua (CD)	Spinnaker, Argo CD, GitHub Actions, Octopus Deploy, Flux CD
Gerenciamento de Configuração	Ansible, Puppet, Chef, SaltStack
Provisionamento e Orquestração de Infraestrutura	Terraform, Pulumi, AWS CloudFormation, Kubernetes
Containerização	Docker, Podman, Buildah
Orquestração de Containers	Kubernetes, OpenShift, Docker Swarm
Monitoramento e Observabilidade	Prometheus, Grafana, ELK Stack, Datadog, New Relic
Segurança e Compliance (DevSecOps)	SonarQube, Snyk, Aqua Security, Trivy, HashiCorp Vault

CI E CD

- Integração contínua (CI):
 - Pipeline baseado em pull-requests, testes e merge automático de código.
- Entrega contínua (CD):
 - Deploy em containers, após os testes, em formato de release contínua, normalmente associado a nuvem.
- São aplicados simultaneamente:
- CI para código e CD para disponibilizar serviço.

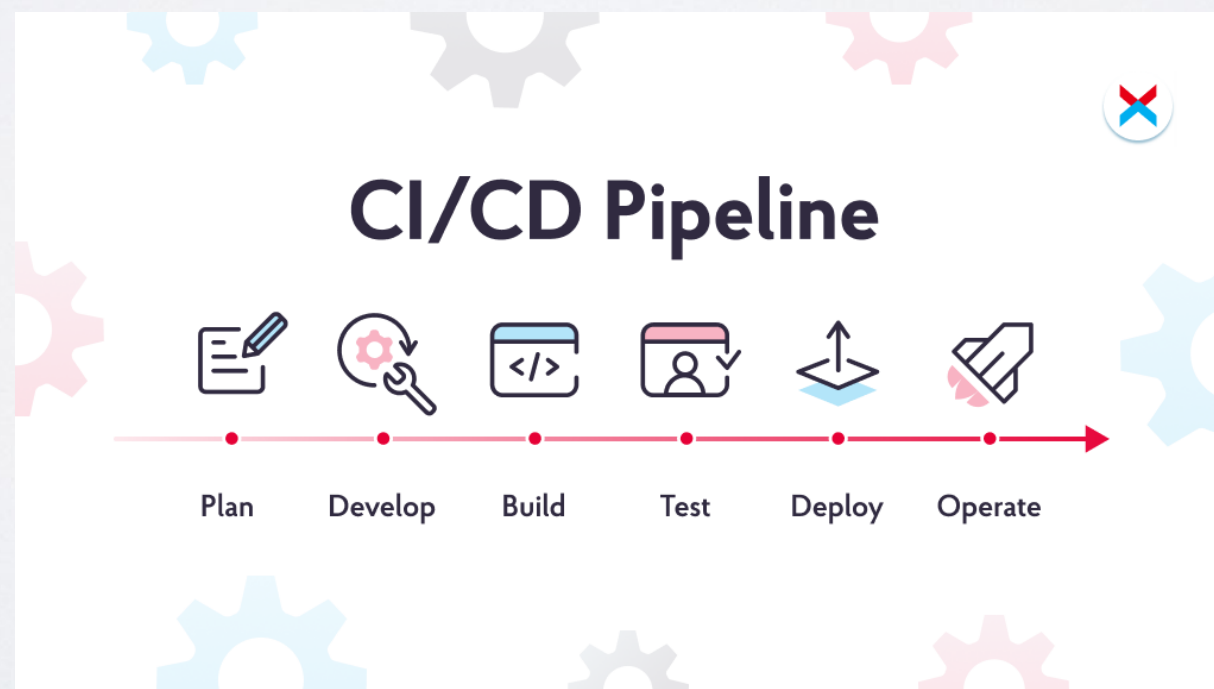


CI/CD EM MICROSERVIÇOS

- Integração Contínua (CI):
 - Cada microservice deve possuir o seu próprio pipeline de build e testes.
 - A integração entre serviços é assegurada por contract tests, que validam compatibilidade e reduzem riscos de falhas em cadeia.
 - O uso de pipelines curtos e objetivos garante feedback rápido e permite corrigir problemas de forma ágil.

CI/CD EM MICROSERVIÇOS

- Escalabilidade: cada microservice evolui de maneira isolada, sem comprometer os demais.
- Resiliência: falhas localizadas podem ser rapidamente revertidas através de rollback automatizado.

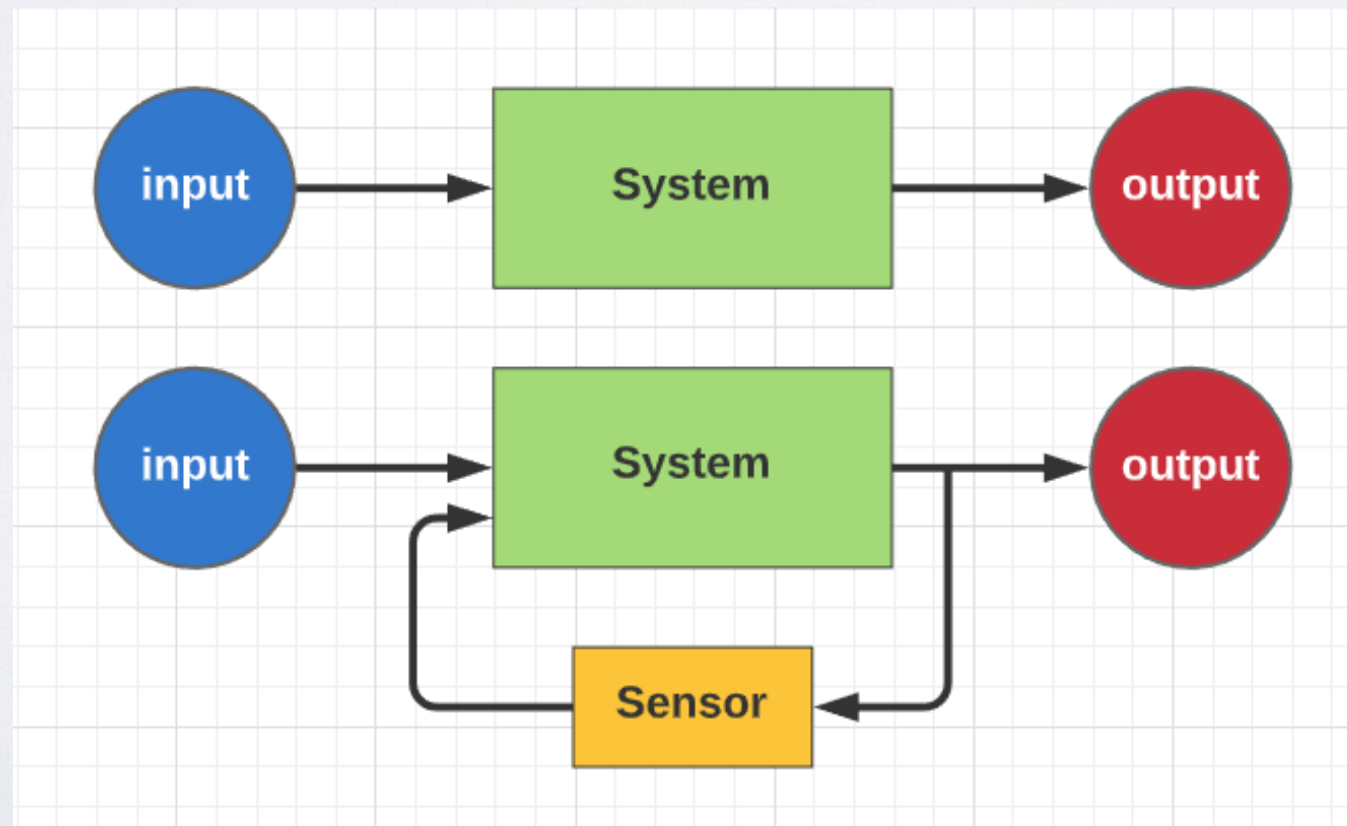


MONITORAMENTO

- O monitoramento é de fundamental importância durante as operações e pode ser combinado com algum controle”.
- O controle em malha aberta (ou seja, o feedback do monitoramento não é levado em consideração) pode ser usado para backups regulares em horários predefinidos.
- No controle em malha fechada, as informações de monitoramento são levadas em consideração ao decidir sobre uma ação, como no exemplo de escalonamento automático.

MONITORAMENTO

- Malha aberta: times em alerta, atuando sobre crises;
- Malha fechada: times de observação e implementação de recuperação de desastres e auto-scaling;



EXERCÍCIO

- Leiam: “An empirical study on principles and practices of continuous delivery and deployment”.
- Implementem utilizando uma ferramenta de monitoramento: Grafana, Prometheus e similares.

BIBLIOGRAFIA

- GALLOTTI, G. M. A. Arquitetura de Software. 1ª Ed. São Paulo: Pearson, 2016. [acervo digital]
- JUNIOR, A. K. Computação em Nuvem. 1ª Ed. Curitiba: Contentus, 2020. [acervo digital]
- ZANETTI, M. Arquitetura de TI e modelos de negócios. 1ª Ed. Curitiba: Contentus, 2020. [acervo digital]
- Baseado nos materiais de aula de Vinícius Garcia - CIn UFPE

DÚVIDAS?

