



A aula interativa do Módulo 4 - Bootcamp Arquiteto Cloud Computing começará em breve!

Atenção:

- 1) Você entrará na aula com o microfone e o vídeo DESABILITADOS.**
- 2) Apenas a nossa equipe poderá habilitar seu microfone e seu vídeo em momentos de interatividade, indicados pelo professor.**
- 3) Utilize o recurso Q&A para dúvidas técnicas. Nossos tutores e monitores estarão prontos para te responder e as perguntas não se perderão no chat.**
- 4) Para garantir a pontuação da aula, no momento em que o professor sinalizar, você deverá ir até o ambiente de aprendizagem e responder a enquete de presença. Não é necessário encerrar a reunião do Zoom, apenas minimize a janela.**

Soluções para Desenvolvimento

Primeira Aula Interativa

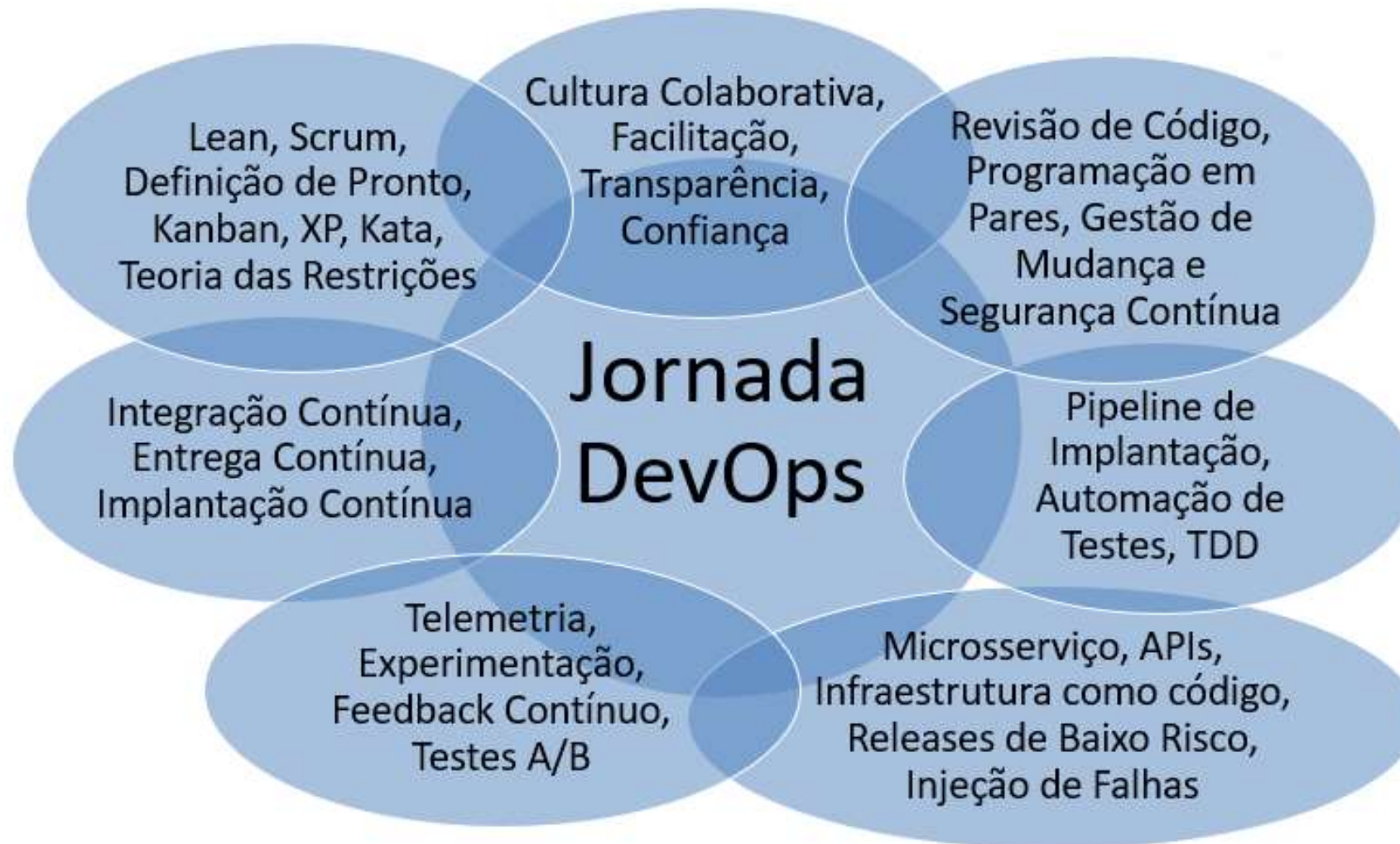
Prof.^a Analía Irigoyen

Soluções para Desenvolvimento

Primeira Aula Interativa – Módulo 4 e Case Prático

Prof.^a Analia Irigoyen

Mindset aceitável no passado	Mindset digital exige DevOps
Filas no check-in para embarcar	Check-in via app
Blockbuster multa por atraso	Netflix entrega vídeo sob demanda
TV domina conteúdo e audiência	Netflix domina conteúdo e audiência
Guia 4 rodas para viagem	Google compra Waze por U\$ 1 Bi
Momento kodak em papel	Facebook compra Instagram por U\$ 1 Bi
Comunicação por e-mail	Facebook compra WhatsApp por U\$ 22 Bi
Filas para transações bancárias	Transações na palma da mão
Filas no orelhão	Qtd de celular supera qtd de pessoas
Táxi acessível para poucos	Uber acessível para muitos
Anúncio caro: TV, jornal e revista	Domínio do Google e mídias sociais
Reembolso médico em papel	Reembolso digital e ágil
Linear e analógico	Exponencial e digital



**Livro Jornada DevOps: MUNIZ; SANTOS; IRIGOYEN; MOUTINHO
(Brasport, 2019)**

Introdução e adoção do Devops – Conceitos Básicos

PERIODIC TABLE OF DEVOPS TOOLS (V3)																															
1 Os Gl GitLab																															2 En Sp Splunk
3 Fm Gh GitHub	4 En Dt Datical																														10 Fm Sg Sumo Logic
11 Os Sv Subversion	12 En Db DBMaestro																														18 Os Fd Fluentd
19 En Cw ISPW	20 En Dp Delphi	21 Os Jn Jenkins	22 Fm Cs Codeship	23 Os Fn FitNesse	24 Fr Ju JUnit	25 Fr Ka Karma	26 Fm Su SoapUI	27 En Ch Chef	28 Fr Tf Terraform	29 En XLd XebiaLabs XL Deploy	30 En Ud UrbanCode Deploy	31 Os Ku Kubernetes	32 Fm Cc CA CD Director	33 En Pr Pivotal Release	34 Pd Al Alibaba Cloud	35 Os Os OpenStack	36 Os Ps Prometheus														
37 Pd At Artifactory	38 Fm Rg Redgate	39 Pd Ba Bamboo	40 Fm Vs VSTS	41 Fr Se Selenium	42 Fr Jm JMeter	43 Os Ja Jasmine	44 Pd Sl Sauce Labs	45 En An Ansible	46 Os Ru Rudder	47 En Oc Octopus Deploy	48 Os Go GoCD	49 Os Ms Mesos	50 Pd Gke GKE	51 Fm Om OpenMake	52 Pd Cp AWS CodePipeline	53 Pd Cy Cloud Foundry	54 En It ITRS														
55 Pd Nx Nexus	56 Os Fw Flyway	57 Os Tr Travis CI	58 Fm Tc TeamCity	59 Os Ga Gatling	60 Fr Tn TestNG	61 Fm Tt Tricentis Tosca	62 Pd Pe Perfecto	63 En Pu Puppet	64 Os Pa Packer	65 Fm Cd AWS CodeDeploy	66 En Ec ElectricCloud	67 Os Ra Rancher	68 Pd Aks AKS	69 Os Rk Rkt	70 Os Sp Spinnaker	71 Pd Ir Iron.io	72 Pd Mg Moogsoft														
73 Fm Bb BitBucket	74 En Pf Perforce	75 Fm Cr Circle CI	76 Pd Cb AWS CodeBuild	77 Fr Cu Cucumber	78 Os Mc Mocha	79 Os Lo Locust.io	80 En Mf Micro Focus UFT	81 Os Sa Salt	82 Os Ce CFEngine	83 En Eb ElasticBox	84 En Ca CA Automate	85 En De Docker Enterprise	86 Pd Ae AWS ECS	87 Fm Cf Codefresh	88 Os Hm Helm	89 Os Aw Apache OpenWhisk	90 Os Ls Logstash														



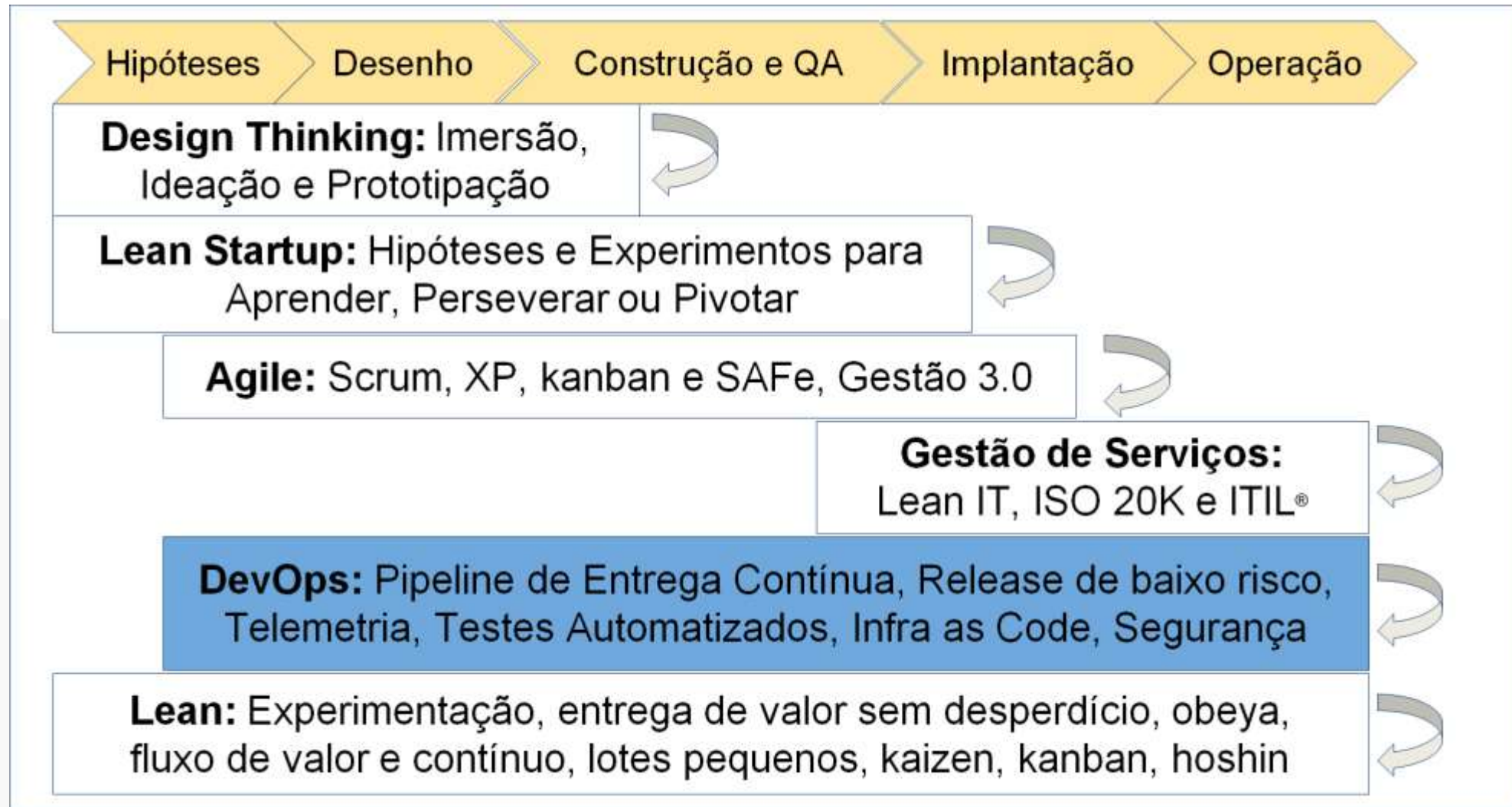
91 En XLi XebiaLabs XL Impact	92 Os Ki Kibana	93 Fm Nr NewRelic	94 En Dt Dynatrace	95 En Dd Datadog	96 Fm Ad AppDynamics	97 Os El ElasticSearch	98 Os Ni Nagios	99 Os Zb Zabbix	100 En Zn Zenoss	101 En Cx Checkmarx SAST	102 En Sg Signal Sciences	103 En Bd BlackDuck	104 Os Sr SonarQube	105 Os Hv HashiCorp Vault
106 En Sw ServiceNow	107 Pd Jr Jira	108 Fm Tl Trello	109 Fm Sk Slack	110 Fm St Stride	111 En Cn CollabNet VersionOne	112 En Ry Remedy	113 En Ac Agile Central	114 Pd Og OpsGenie	115 Pd Pd Pagerduty	116 Os Sn Snort	117 Fm Tw Tripwire	118 En Ck CyberArk	119 En Vc Veracode	120 En Ff Fortify SCA

<https://softwarezen.me/campaigns/warmup/gravacao/>

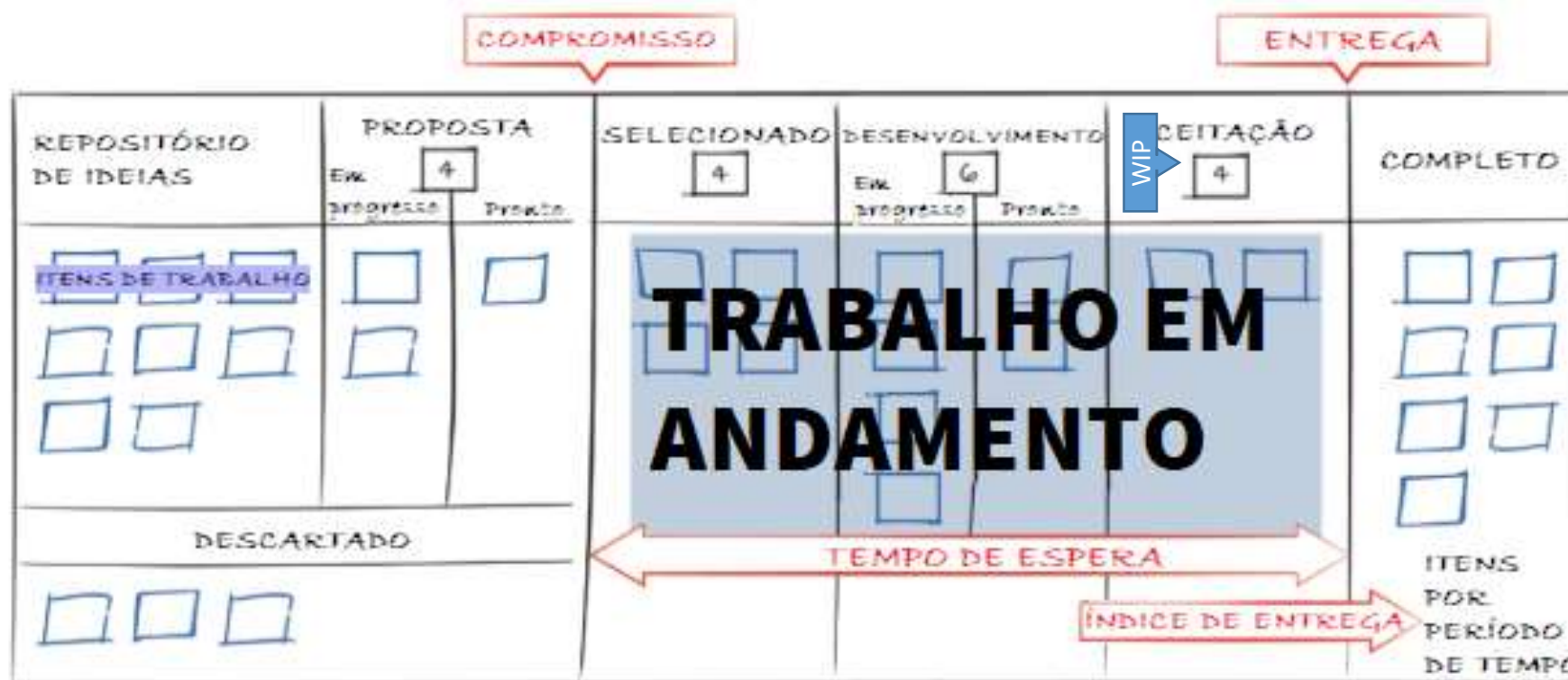
A má implementação ou excesso de customização pode destruir uma ótima ferramenta...

**Incentive a opinião das equipes no chão de fábrica!
Vá ao gemba !**





KANBAN – Produção Puxada



Dívida técnica (Débito técnico)

- ✓ **Representa o resultado das decisões do curto prazo que geram problemas que se tornam cada vez mais difíceis de resolver com o passar do tempo**
- ✓ Reduz a performance da equipe e aumenta o TCO
- ✓ Os objetivos conflitantes das áreas Dev e Ops contribuem para o aumento do débito técnico

Algumas causas da dívida técnica

- ✓ Ausência de testes
- ✓ Requisitos não claros
- ✓ Documentação pobre
- ✓ Pouco foco em refatoração
- ✓ Equipes não colaborativas
- ✓ Conflitos de interesse
- ✓ Pressão de chefes e áreas de negócio

Princípios Lean



Fluxo de valor: É o processo que concretiza uma necessidade de negócio em um produto ou serviço para entrega de valor ao cliente.

Mapeamento do fluxo de valor: Visa entender como o processo funciona com foco na entrega de valor ao cliente e identifica gargalos ou desperdícios.

Gemba: É o local onde as coisas acontecem e todos deveriam ir ao gemba com frequência para conhecer o “chão de fábrica” e evitar suposições sem dados e fatos.

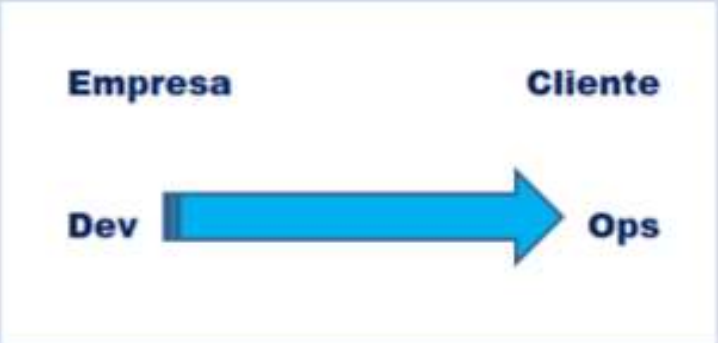
Obeya: Também conhecida nas organizações como “sala de guerra”, o objetivo é facilitar a gestão visual e a coordenação para solução de problemas sem os entraves das estruturas organizações clássicas.

Desperdício

Existem 8 tipos de desperdícios identificados no Lean:

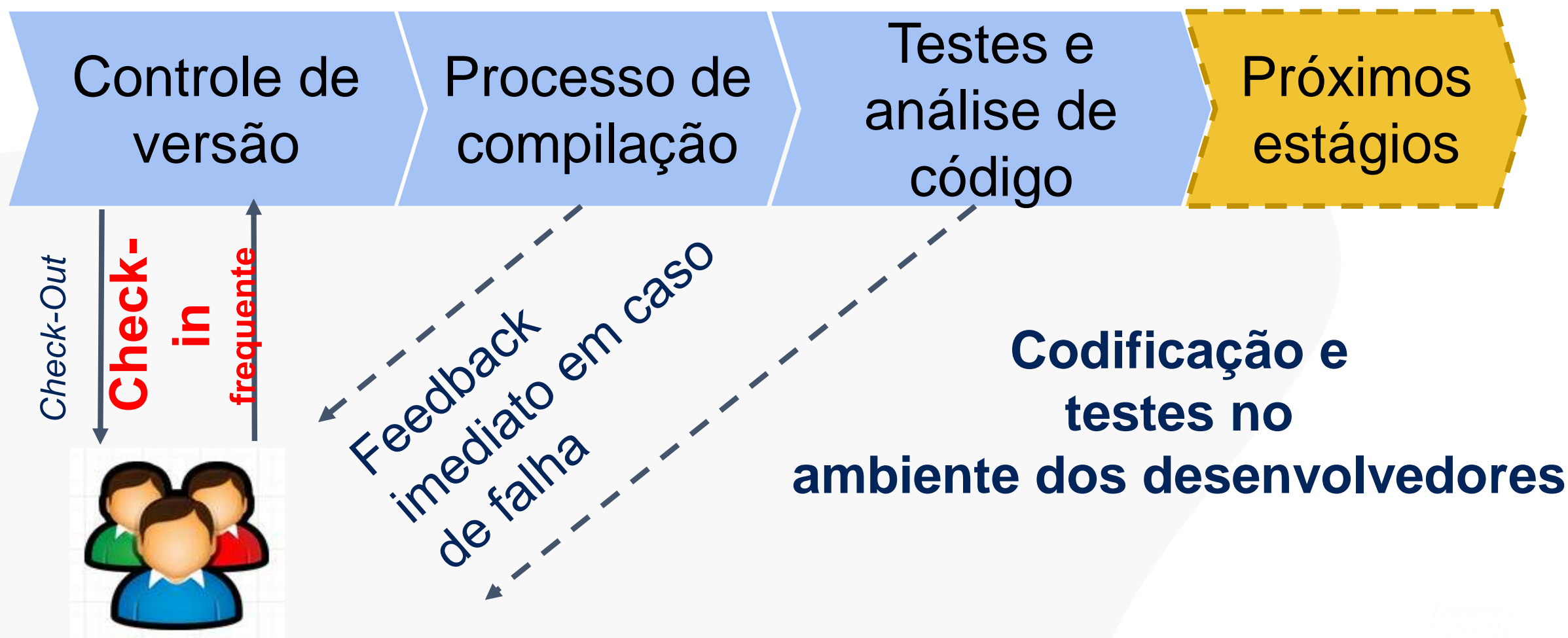
- Defeito e retrabalho: Desfazer, refazer algo.
- Movimentação: Caminhadas, deslocamentos, viagens.
- Espera: Pessoas aguardando informações, materiais ou outras equipes.
- Transporte: Transferências desnecessárias de materiais ou informações.
- Estoques: Informações ou materiais sem uso.
- Processamento: Etapa redundante ou desnecessária.
- Desconexão ou superprodução: Fluxo deficiente ou falta de sincronismo entre etapas (antes ou depois do necessário).
- Conhecimento: Não aproveitar as habilidades das pessoas adequadamente.

Corda de Andon: Dispositivo que existe nas fábricas da Toyota para interromper a linha de produção quando é encontrado algum defeito nos produtos. O objetivo é aglomerar imediatamente todas as equipes e líderes que podem ajudar a resolver o problema na origem, podendo mobilizar os executivos e a alta administração.

Objetivo da Primeira Maneira	Princípios e Práticas
<p>Acelerar o fluxo dos desenvolvedores (Esquerda) para operação e clientes (Direita)</p> 	<ol style="list-style-type: none">1. Tornar o trabalho visível2. Reduzir o tamanho dos lotes e intervalos3. Aplicar teoria das restrições e otimizar o fluxo4. Remover desperdícios e foco no cliente5. Reduzir o número de transferências (handoff)6. Incorporar qualidade na origem7. Limitar o trabalho em andamento (WIP)8. Infraestrutura como código e self service9. Integração, entrega e implantação contínua10. Testes automatizados e TDD11. Arquitetura e releases de baixo risco

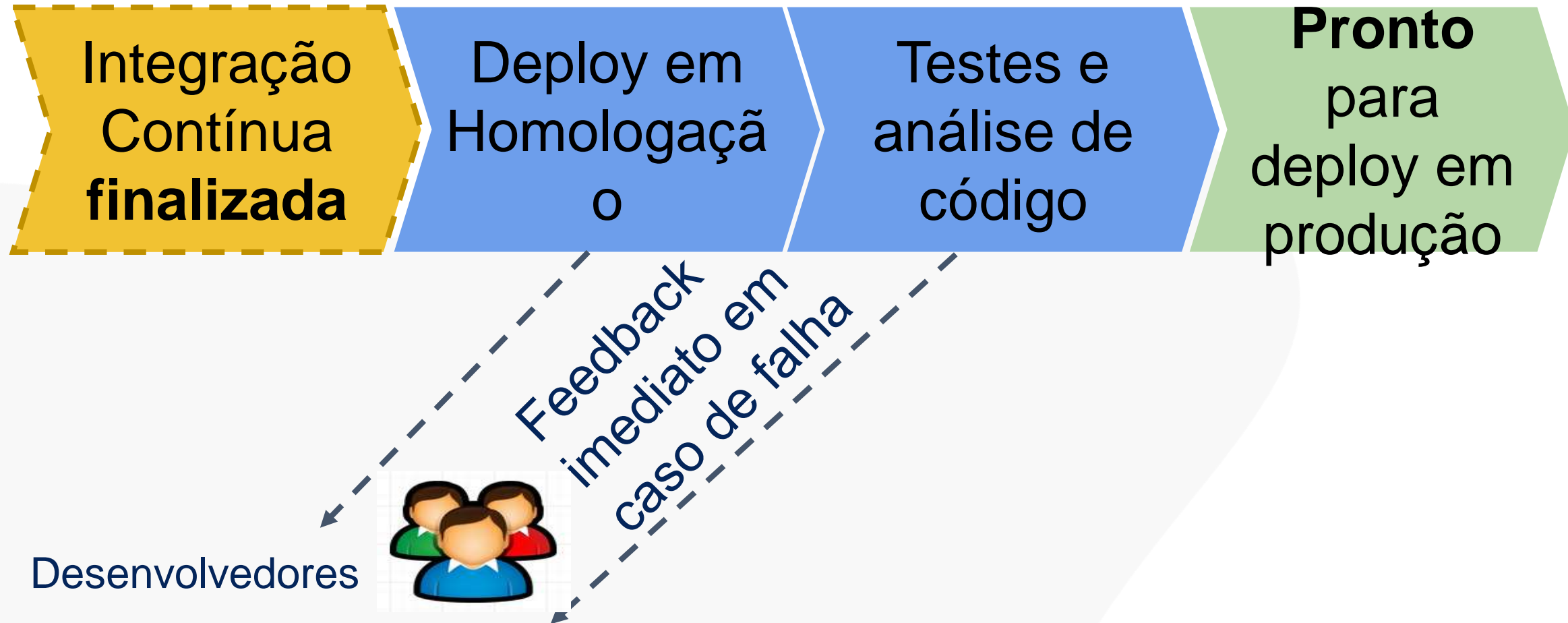
Livro Jornada DevOps: MUNIZ; SANTOS; IRIGOYEN; MOUTINHO (Brasport, 2019)

Resumo da Integração contínua

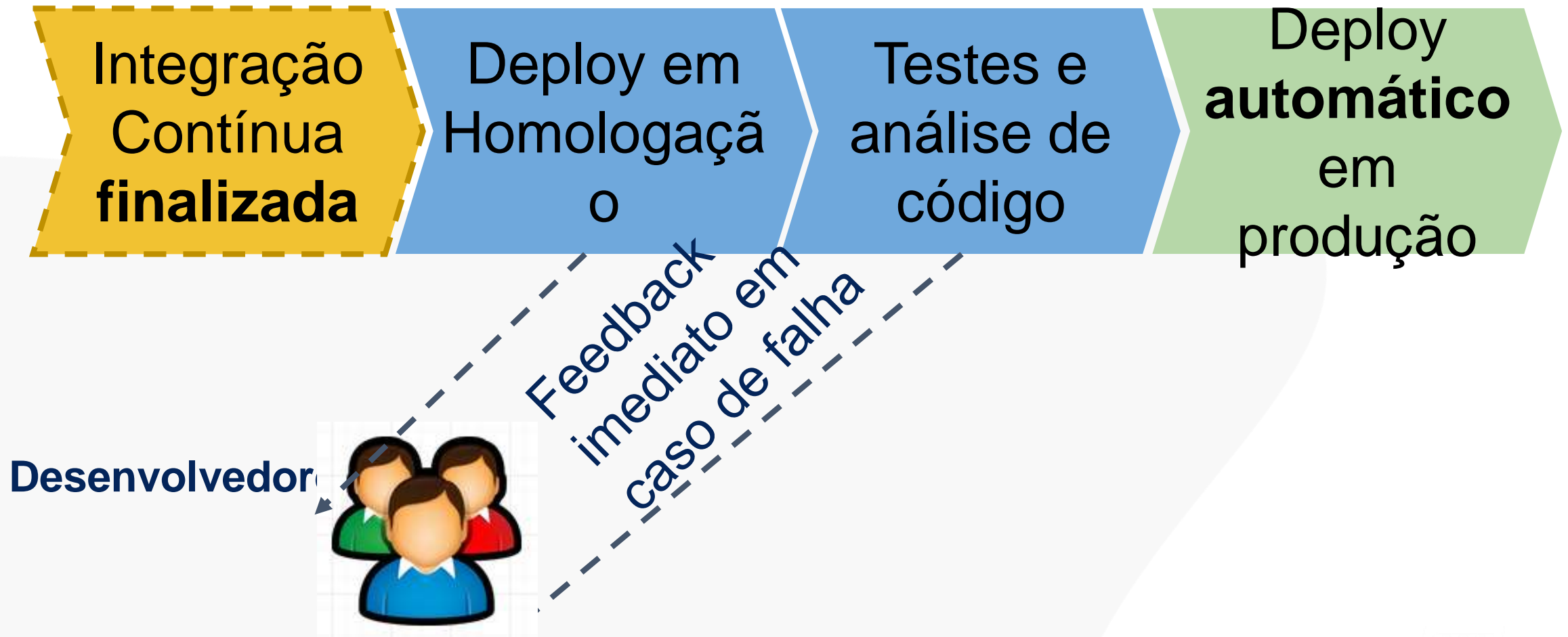


Resumo da Entrega contínua

IGTI



Resumo da Implantação contínua



Infraestrutura ágil



É a aplicação dos princípios ágeis na infraestrutura

Foco principal é a infraestrutura como código:

Gerência e provisão de ambientes automaticamente

Tudo fica centralizada no controle de versões

Nuvem/virtualização é a base e existem ferramentas para otimizar (Puppet, Terraform, Chef, Ansible, etc.)

Infraestrutura como código (IaC)



- ✓ Código e configurações dentro do controle de versão
- ✓ Criação automatizada e sob demanda (self-service) em todos os ambientes, evitando trabalho manual
- ✓ Todos os estágios do fluxo de valor com ambientes iguais ou semelhantes ao de produção
- ✓ **Infraestrutura imutável:** Foco em recriar todo o ambiente de produção de forma rápida em vez de realizar alterações

IaaS, SaaS e PaaS



Fonte: <https://azure.microsoft.com/>

PaaS AzureDevOps + Azure Services



[Azure DevOps](#)

O Azure DevOps gerencia o processo de desenvolvimento.



[Microsoft
Release
Management](#)

Os agentes de build e lançamento do Microsoft Release Management implantam o modelo do Azure Resource Manager e o código associado nos vários ambientes.



[Grupos de
recursos](#)

Os grupos de recursos do AzureDevOps são usados para definir todos os serviços necessários para implantar a solução em um ambiente de desenvolvimento e teste ou de produção.



[Aplicativos Web](#)

Um aplicativo Web executa o site e é implantado em todos os ambientes. Slots de preparo são usados para alternar entre versões de pré-produção e de produção.

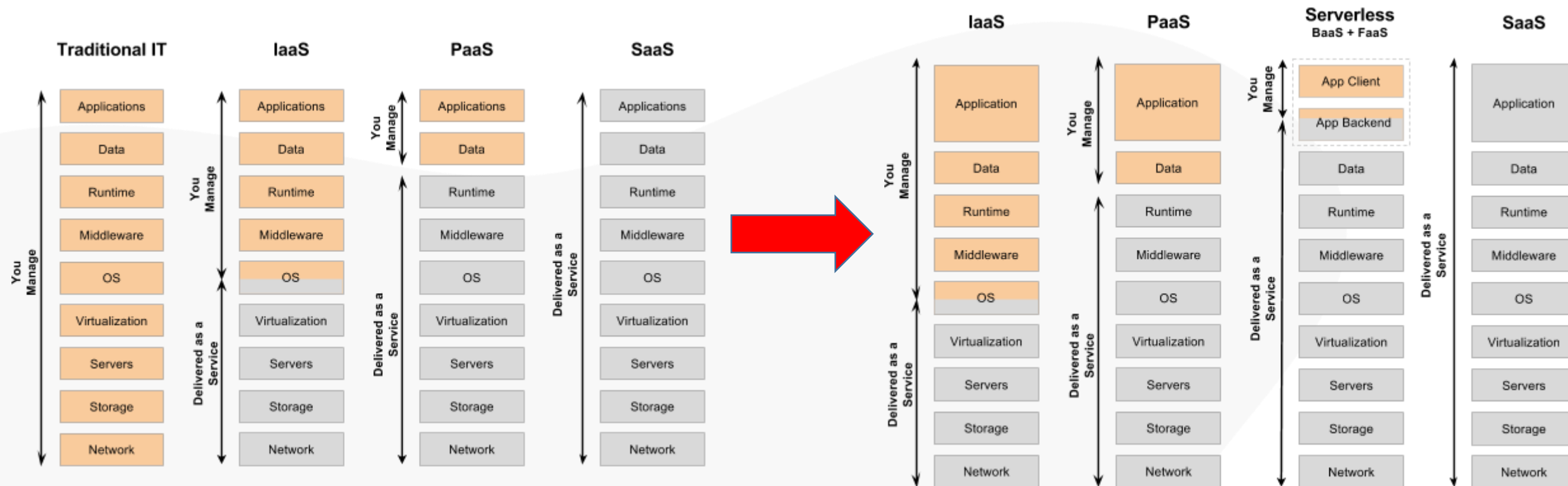
Ciclo de Desenvolvimento

Criação de Releases

Criação de Pipelines

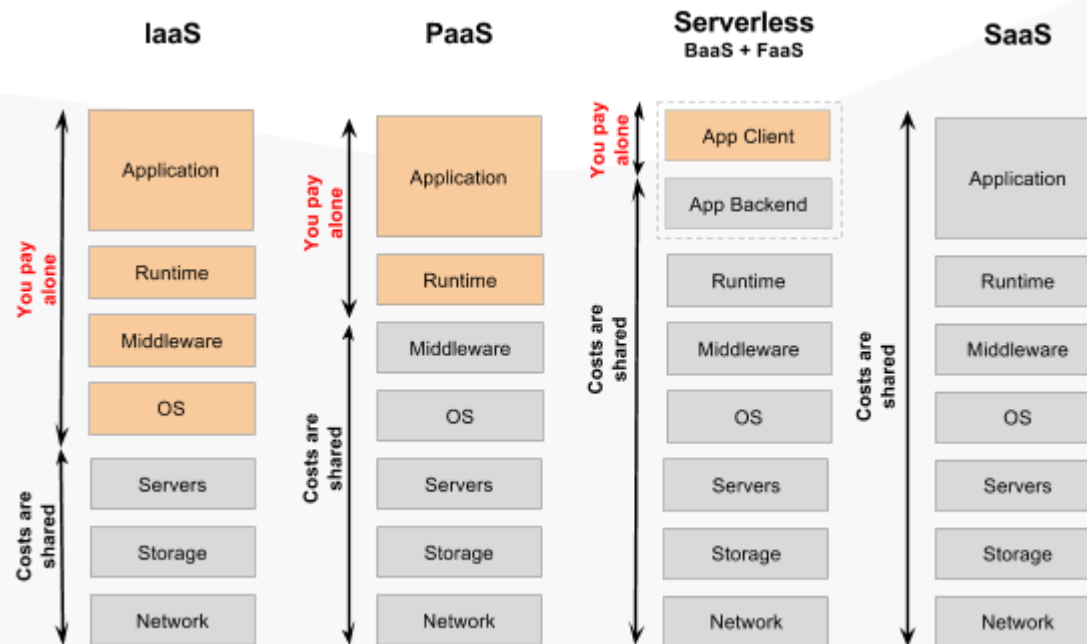
Criação de Ambiente
para hospedar um site

Serverless – Backend as a Service (BaaS) and Function as a Service (FaaS).



Fonte: <https://specify.io/concepts/serverless-baas-faas>

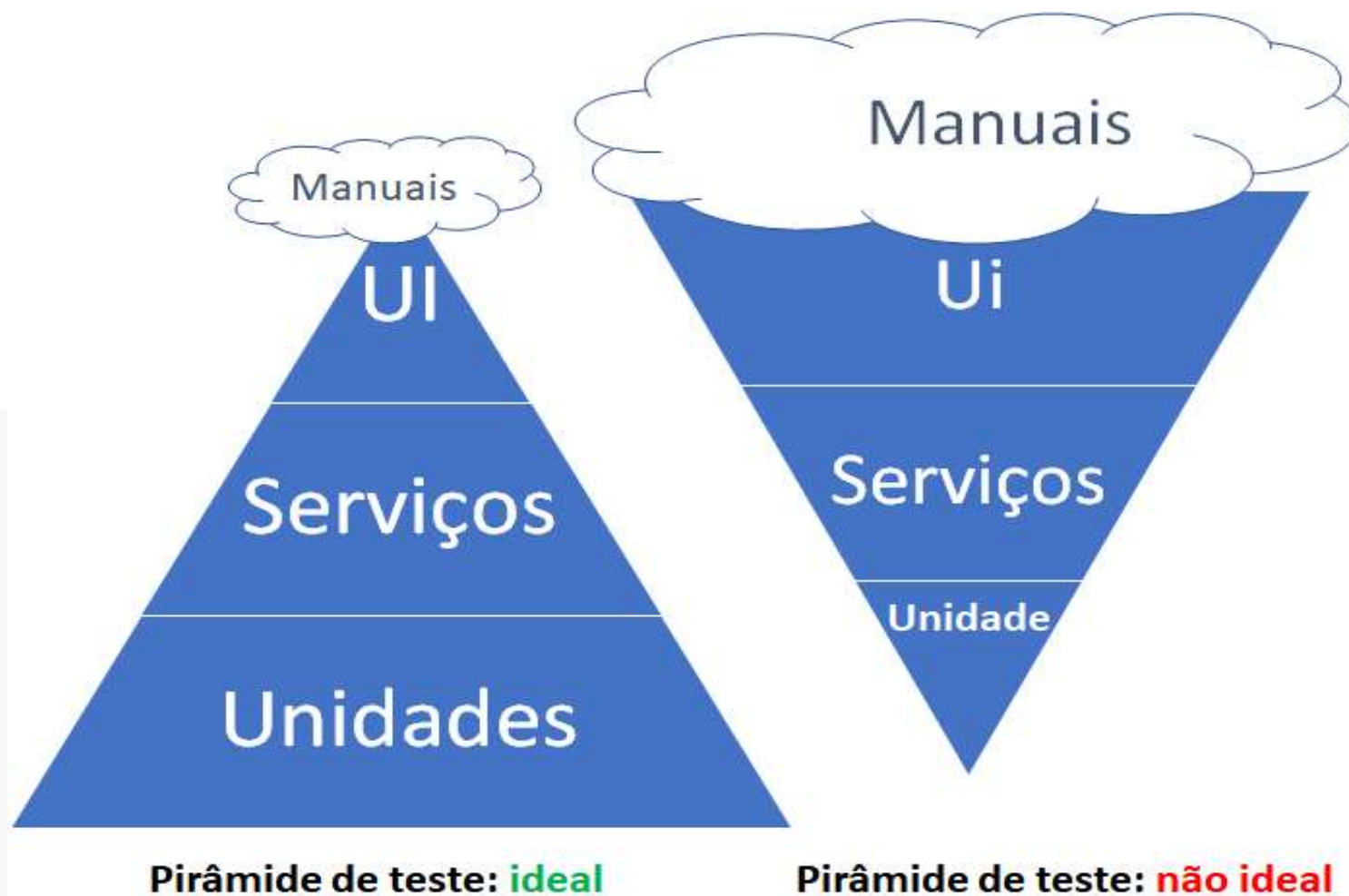
Serverless – Backend as a Service (BaaS) and Function as a Service (FaaS).



Quando você está usando FaaS, você compartilha o tempo de execução com outras pessoas. Por exemplo: quando sua "função" é escrita em JavaScript, esta parte do código será executada no mesmo servidor node.js que as "funções" de outros usuários deste FaaS (isso pode ser diferente para alguns fornecedores).

Ao usar o BaaS, você compartilha o mesmo BaaS com outros usuários deste BaaS (isso pode ser diferente para alguns fornecedores).

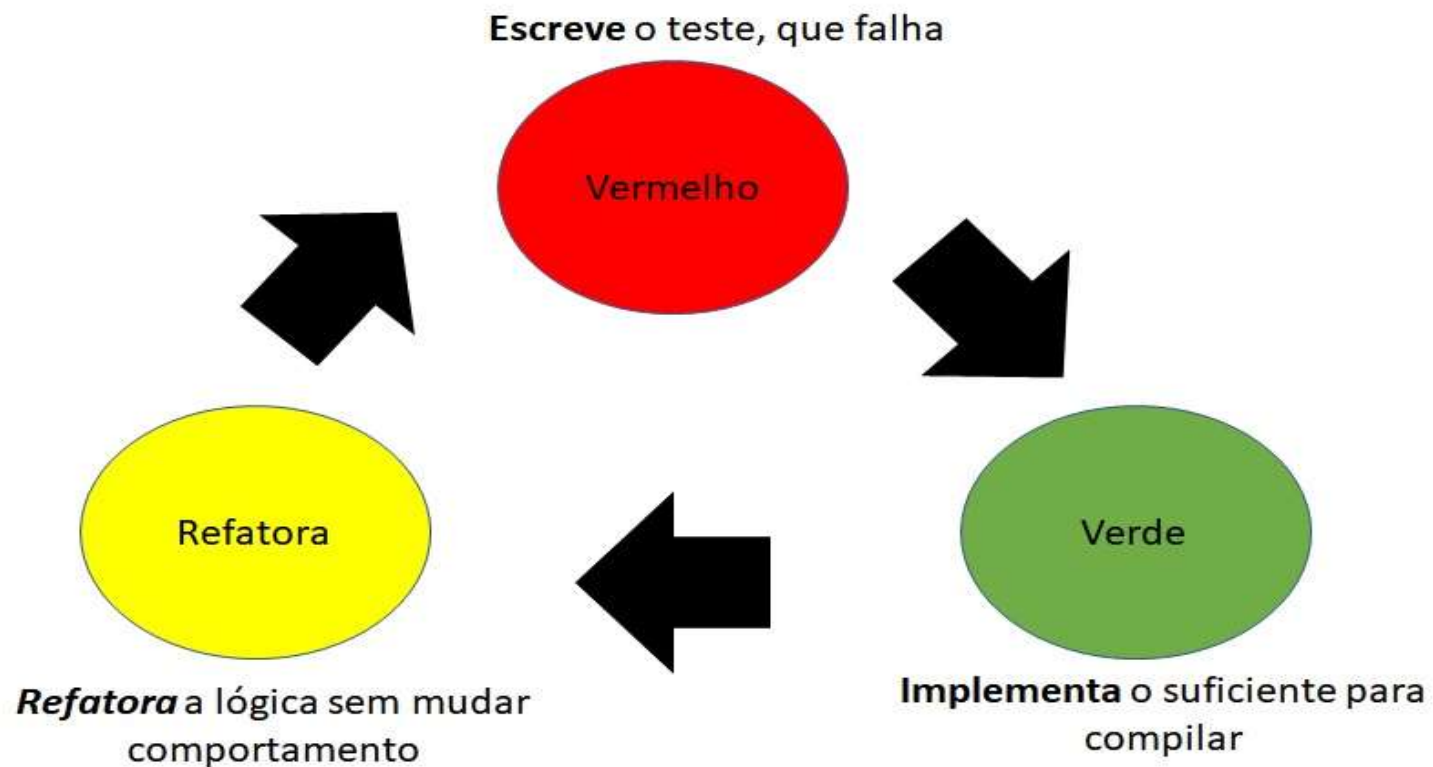
Fonte: <https://specify.io/concepts/serverless-baas-faas>



iGTi



Sequência básica do TDD



Livro Jornada DevOps: MUNIZ; SANTOS; IRIGOYEN; MOUTINHO (Brasport, 2019)

Estratégias de branching (ramificação)

Estratégia	Vantagens	Desvantagens
Produtividade individual	Projeto privado que não atrapalha outras equipes	Merge do código ocorre no final do projeto e gera muitos problemas
Produtividade da equipe (Desenv. Baseado no trunk)	Fila única com todos trabalhando no trunk e commit frequente Não há o estresse de merge no final do projeto	Difícil de implantar e cada commit pode quebrar o projeto inteiro Deve-se puxar a corda de andon para corrigir

Categorias de liberação (Release)

Baseado no Ambiente

Há 2 ou mais ambientes e apenas um fica ativo para os clientes (Ex.: balanceadores)

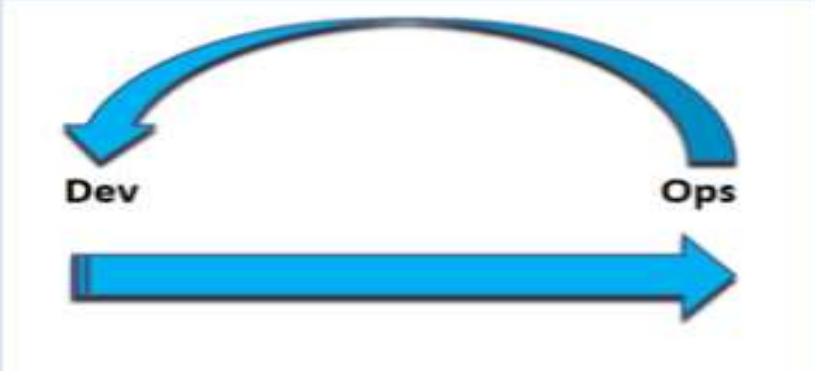
Azul verde (Blue-green) e
Canário (Canary)

Baseado no aplicativo

Novas funcionalidades de forma seletiva usando configurações simples (não precisa fazer deploy)

Alternância de recursos (Feature toggles) e Lançamento escuro

Arquitetura	Vantagens	Desvantagens
Monolítica	<ul style="list-style-type: none">Inicialmente simplesBaixa latência entre processosEficiente para recursos em pequena escala	<ul style="list-style-type: none">Fraca escalabilidade e redundânciaImplantação big bangLongo período de build
Mirosserviço	<ul style="list-style-type: none">Cada unidade é simplesEscalonamento independenteTeste e implantação independenteFacilita visão por produto	<ul style="list-style-type: none">Latência de redePrecisa de ferramentas para gerenciar dependências

Objetivo da Segunda Maneira	Princípios e Práticas
<p>Rápido feedback em todos os estágios do fluxo de valor (Direita para a Esquerda)</p> 	<ol style="list-style-type: none">1. Ver problemas quando ocorrem (“ir ao gembu”)2. Aglomerar quando problema aparece (Andon)3. Qualidade próxima da fonte (menos aprovações)4. Telemetria self service e irradiadores de informação disponível para todos5. Desenvolvimento por hipóteses e Testes A/B6. Equipes Dev e Ops compartilham o trabalho diário e plantões de suporte 24 x77. Revisão de código usando as técnicas: Programação em pares, sobre os ombros, divulgação por email, assistida por ferramentas

Telemetria



- ✓ Coleta remota e automática de dados para monitorar a saúde das aplicações
- ✓ Visão de todo o fluxo de valor permite otimizar e comunicar melhor

Fornece feedback efetivo:

- ✓ **Experiência do cliente**
- ✓ Erros no pipeline de implantação e após deploy
- ✓ Eventos proativos
- ✓ **Funcionalidades mais usadas pelos clientes**

Feedback

Visão clara do fluxo de valor em todos os estágios do ciclo de vida do serviço, produto, desenvolvimento e implementação, entrada em operação e retirada

Dessa forma, todos os participantes do fluxo de valor tem a oportunidade de aprender desde os primeiros estágios do projeto

Opções para resolver problemas



Correção: Fix Forward

É uma mudança no código e deve ser executada em ambientes com grande maturidade:

Teste automatizado,
implantação rápida e
telemetria

Reversão: Rollback

Usuário deixa de receber a implantação e há 2 formas:

1. Implantar a versão anterior (Ex.: Canário)
1. Alternância de recursos:
Mais fácil de reverter e
menos arriscada

Suporte compartilhado



A implantação de um pipeline com testes automatizado não garante zero erro, em função da complexidade dos sistemas

Erro de software depende que a equipe Dev priorize a solução para não impactar a operação por muito tempo

Todos os participantes do Fluxo de Valor devem compartilhar as responsabilidades para resolver os incidentes em produção

Eficácia do Pull Request

Recomendação
é que a revisão
seja realizada
pelo colega
antes de efetivar
o código no trunk

Revisão deve ser em pequenos lotes.
Shoup diz: “Quando aumentamos o
código de 10 linhas para 100 linhas, a
chance de erro aumenta 10 vezes”

Mudanças críticas podem ter a
revisão do especialista no assunto
(segurança, BD, redes)

Programação em pares



O código é criado por duas pessoas trabalhando juntas em um único computador

A melhor maneira de parear o programa é sentar lado a lado na frente do monitor para revezar as atividades de codificação (piloto) e revisão (navegador)

Tempo gasto a mais é 15% e aumenta qualidade do software de 70% para 85%