

Engenharia de Produto

Prof. Carla Rocha

rocha.carla@gmail.com

<https://github.com/fga-gpp-mds>



1.

Engenharia de Software

Como é vista por MDS

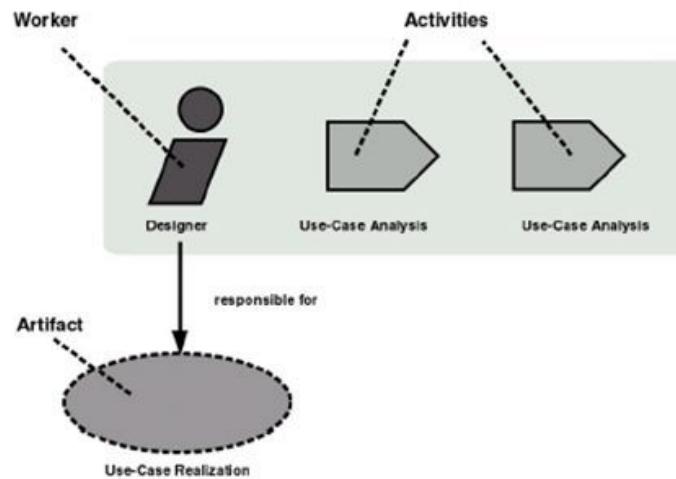
Estrutura do processo:

Intervenientes (*Workers*) - Quem? (*who*)

Atividades (*Activities*) - Como? (*how*)

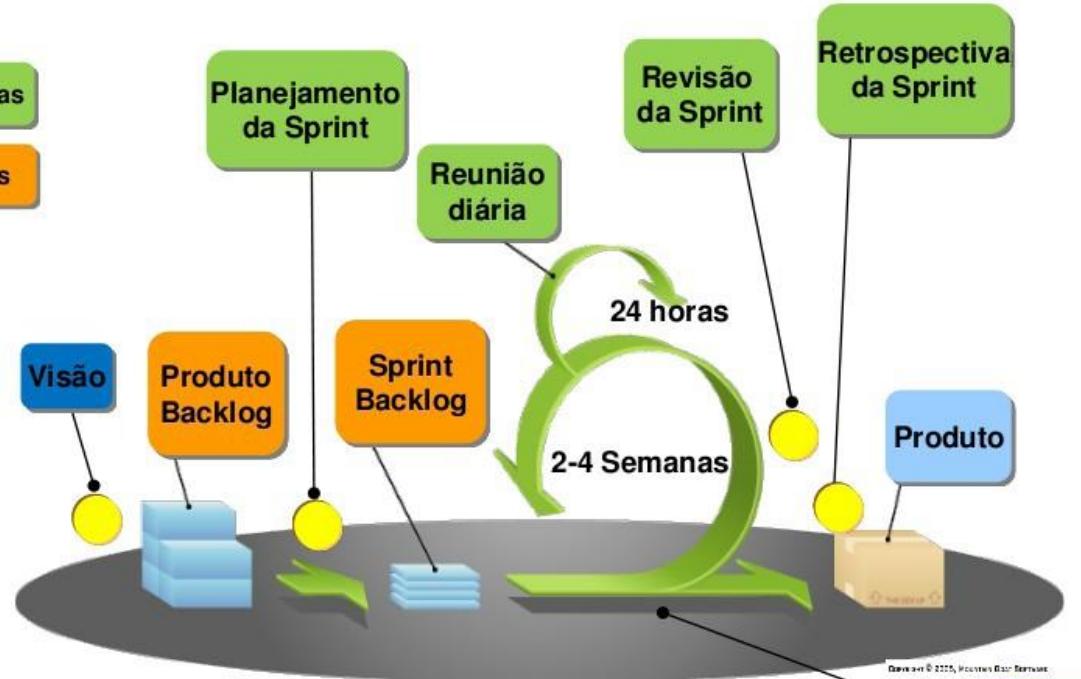
Artefatos (*Artifacts*) - O Que? (*what*)

Fluxo de Trabalho (*Workflows*) - Quando? (*when*)



Legenda:

- Cerimônias
- artefatos



Dados em © 2015, Moisés Ribeiro Barreto

Papéis

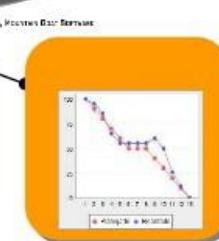
- Product Owner (PO)
- ScrumMaster (SM)
- Equipe Scrum

Cerimônias

- Planejamento da Sprint
- Reunião Diária
- Revisão da Sprint
- Retrospectiva da Sprint

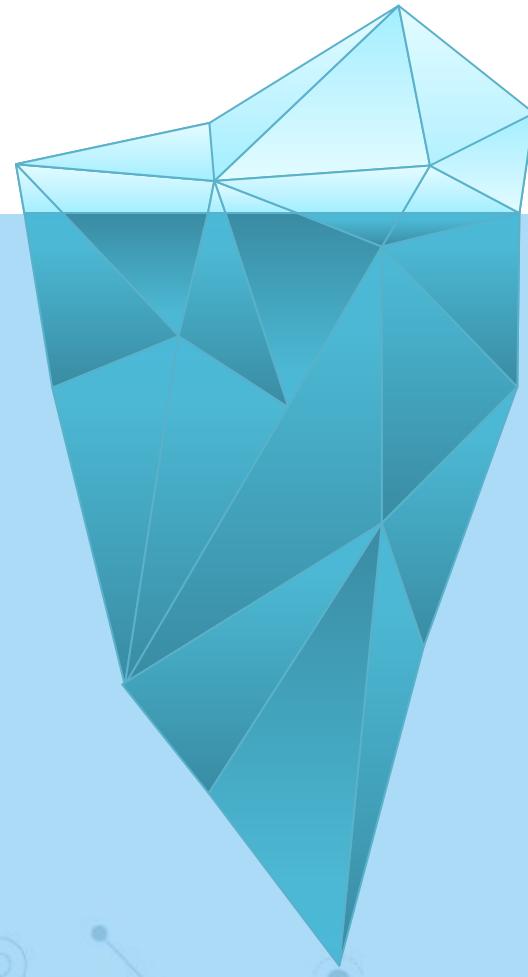
Artefatos

- Product Backlog
- Sprint Backlog
- Burndown (gráfico)



Burndown

Desenvolver funcionalidades é somente a “ponta do Iceberg”
de um produto de software

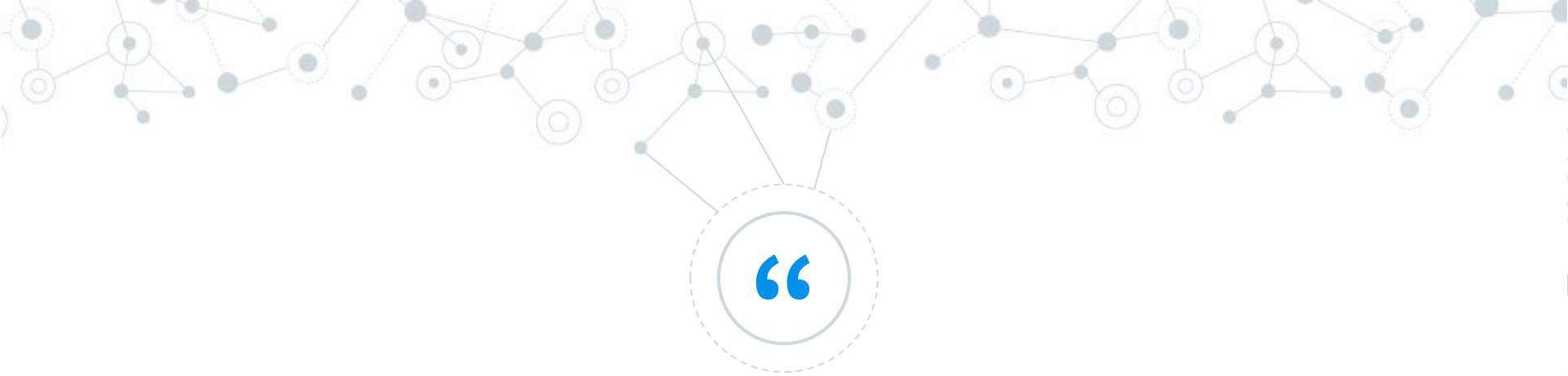


A faint, light-gray network diagram serves as the background for the entire slide. It consists of numerous small, semi-transparent circles of varying sizes connected by thin gray lines, creating a complex web-like pattern.

2.

Engenharia de Software

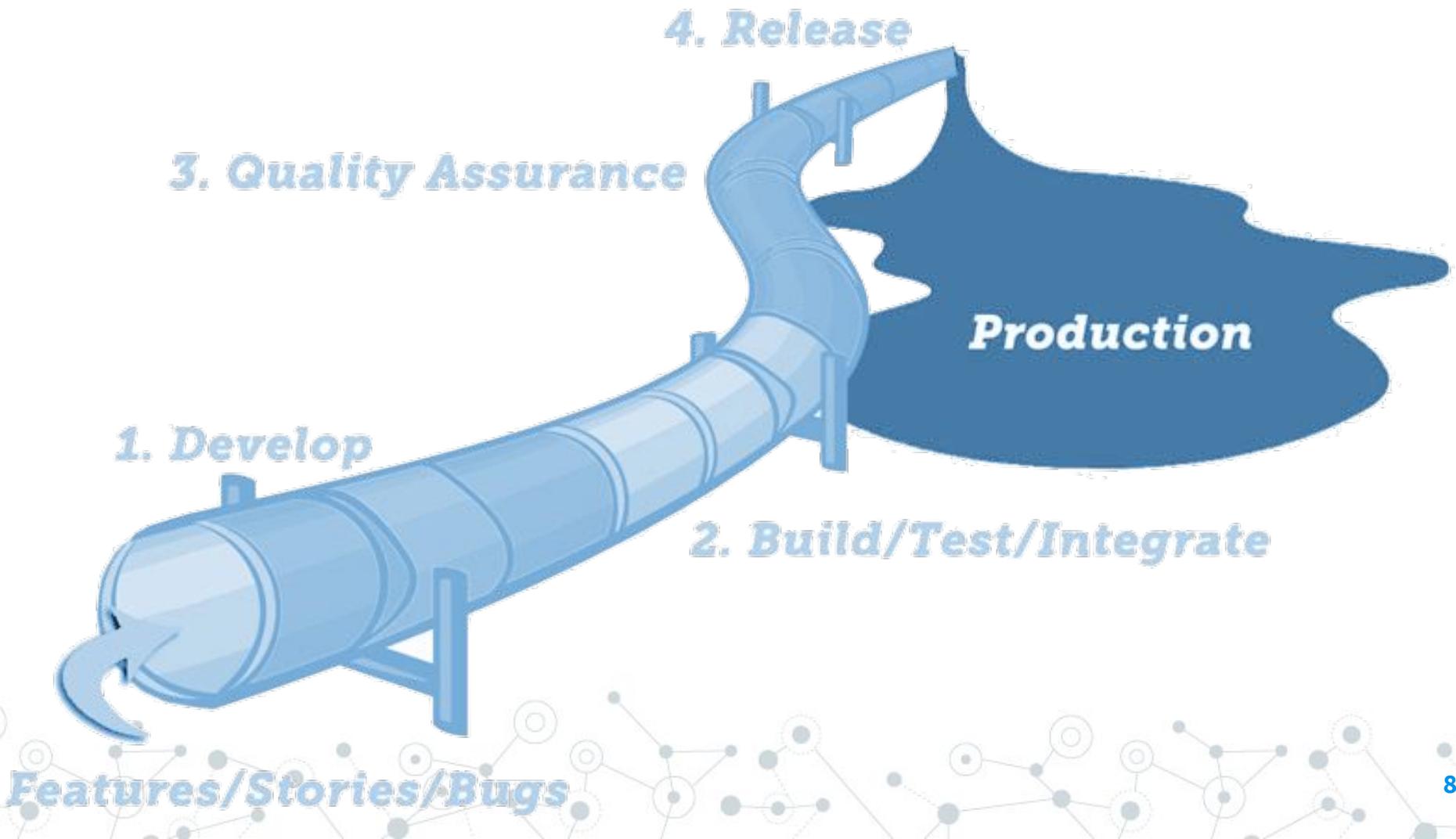
Como é vista por Engenharia de
Produto



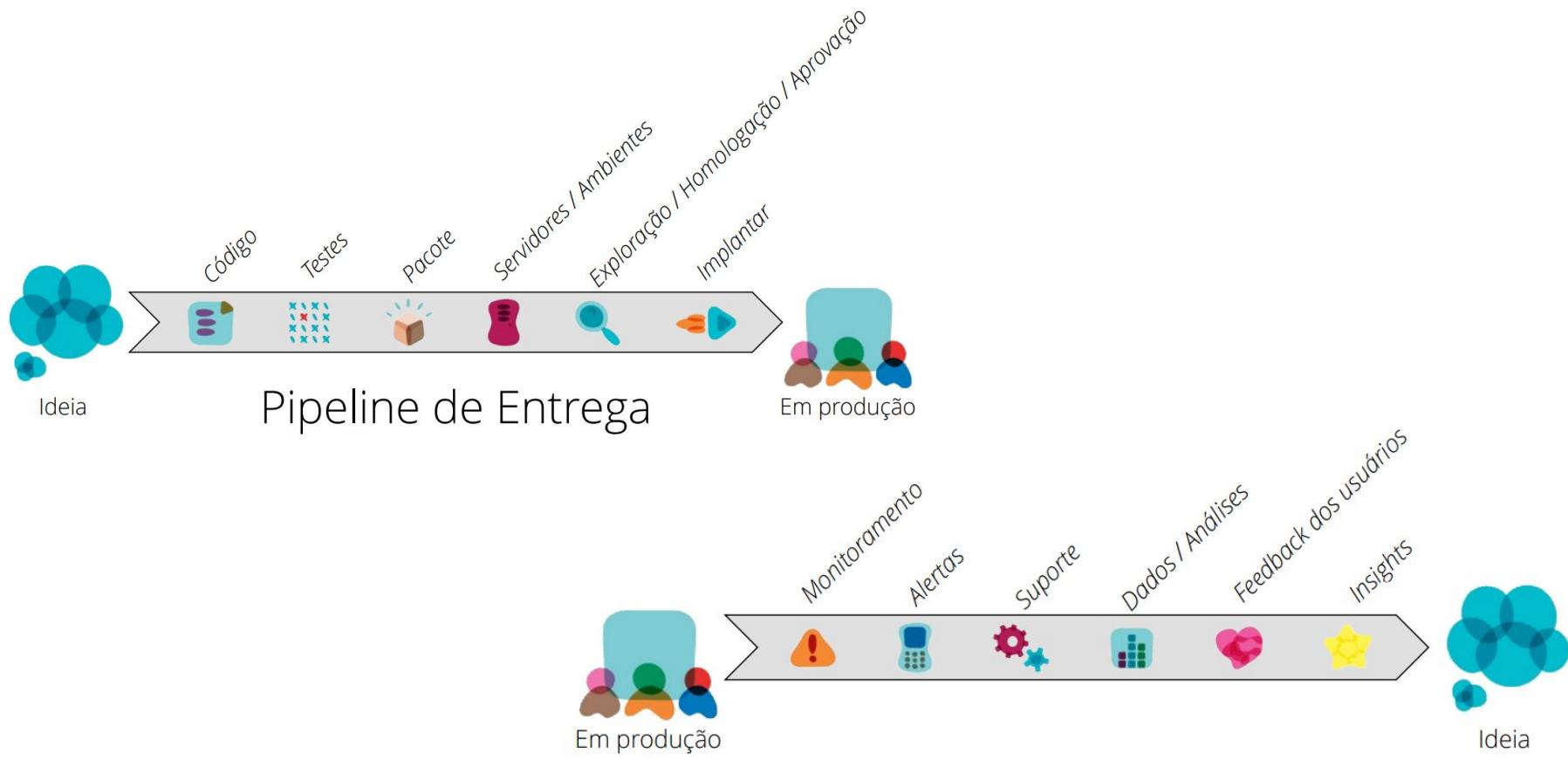
“

Engenharia de produto é
responsável por **desenvolver** o
produto e **mantê-lo operando**

Engenharia de Produto - Pipeline Completo de Entrega de Software



Engenharia do Produto - Da ideia à entrega





2,000,000,000

Linhas de Código

40,000

Commits por dia

Código fonte do Google

250,000

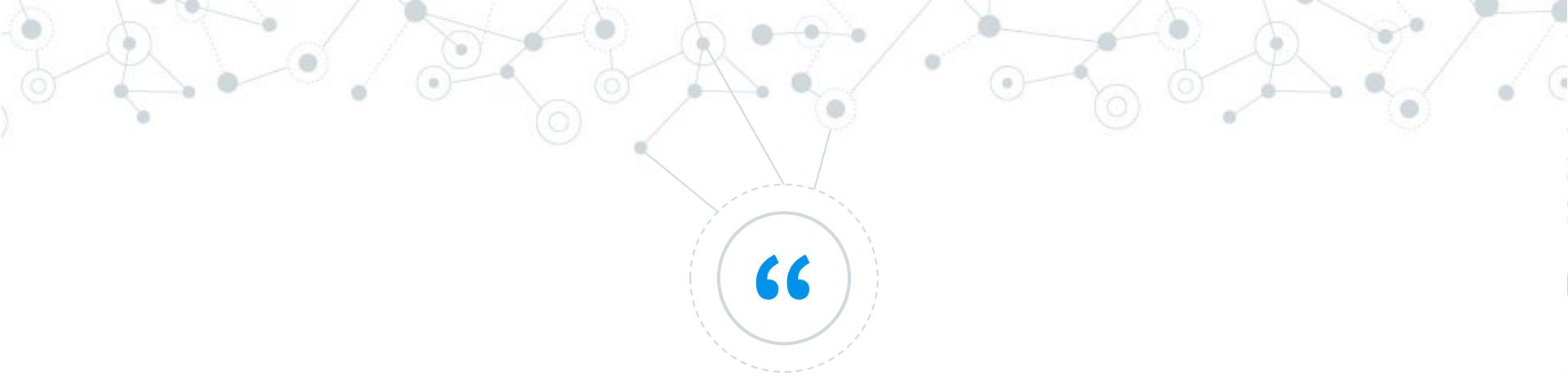
Arquivos modificados por semana!

Engenharia de Produto - Caso Google

“Google’s source code is big, on the order of 2 billion lines of code (LOC) and it evolves rapidly [1], [2]. On average, 40.000 code changes are submitted to the Google code repository every day, and 15 million lines of code in 250.000 files are modified every week [2].” (Who Broke the Build?, 2017)

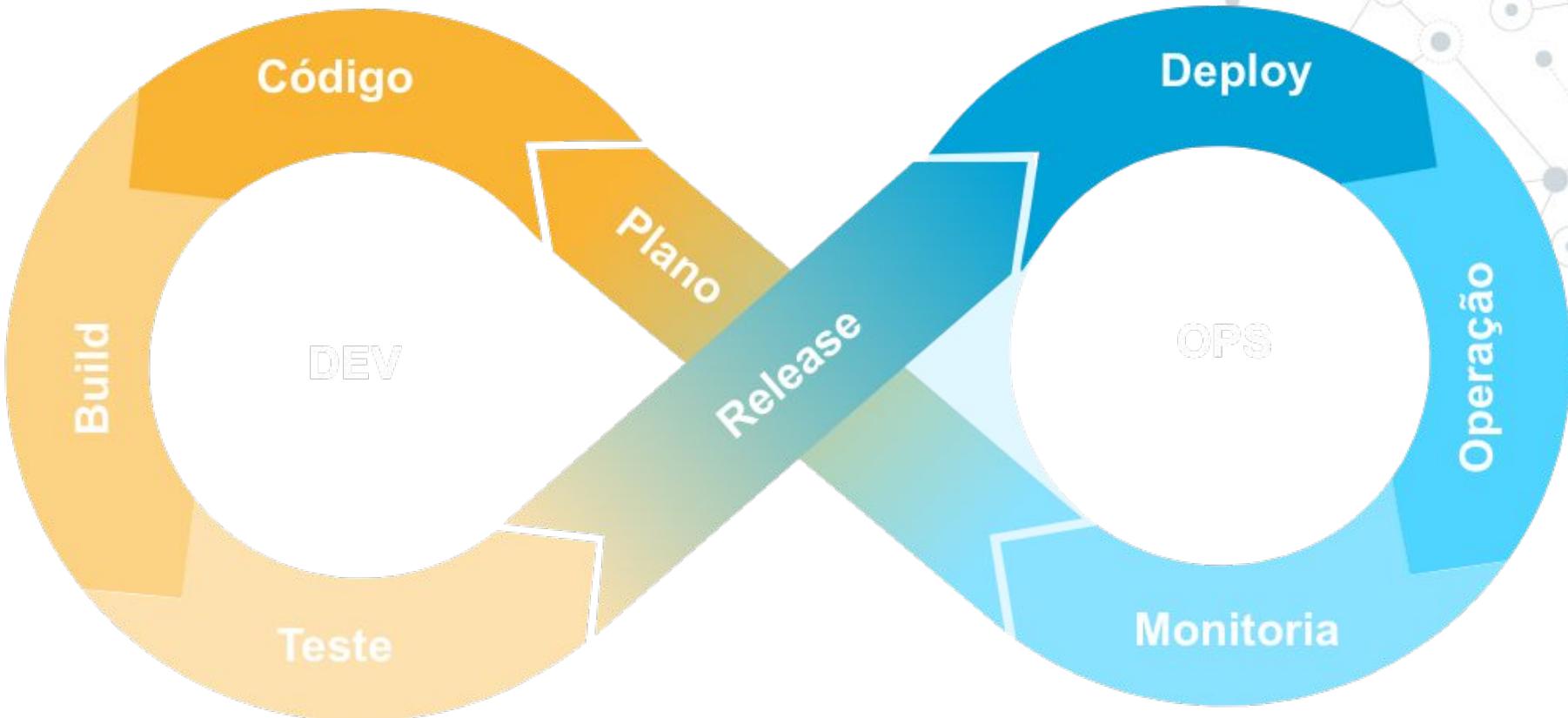


Entrega Contínua é uma **disciplina** de desenvolvimento de software onde você desenvolve software de uma maneira que ele possa ser **entregue em produção a qualquer momento.**



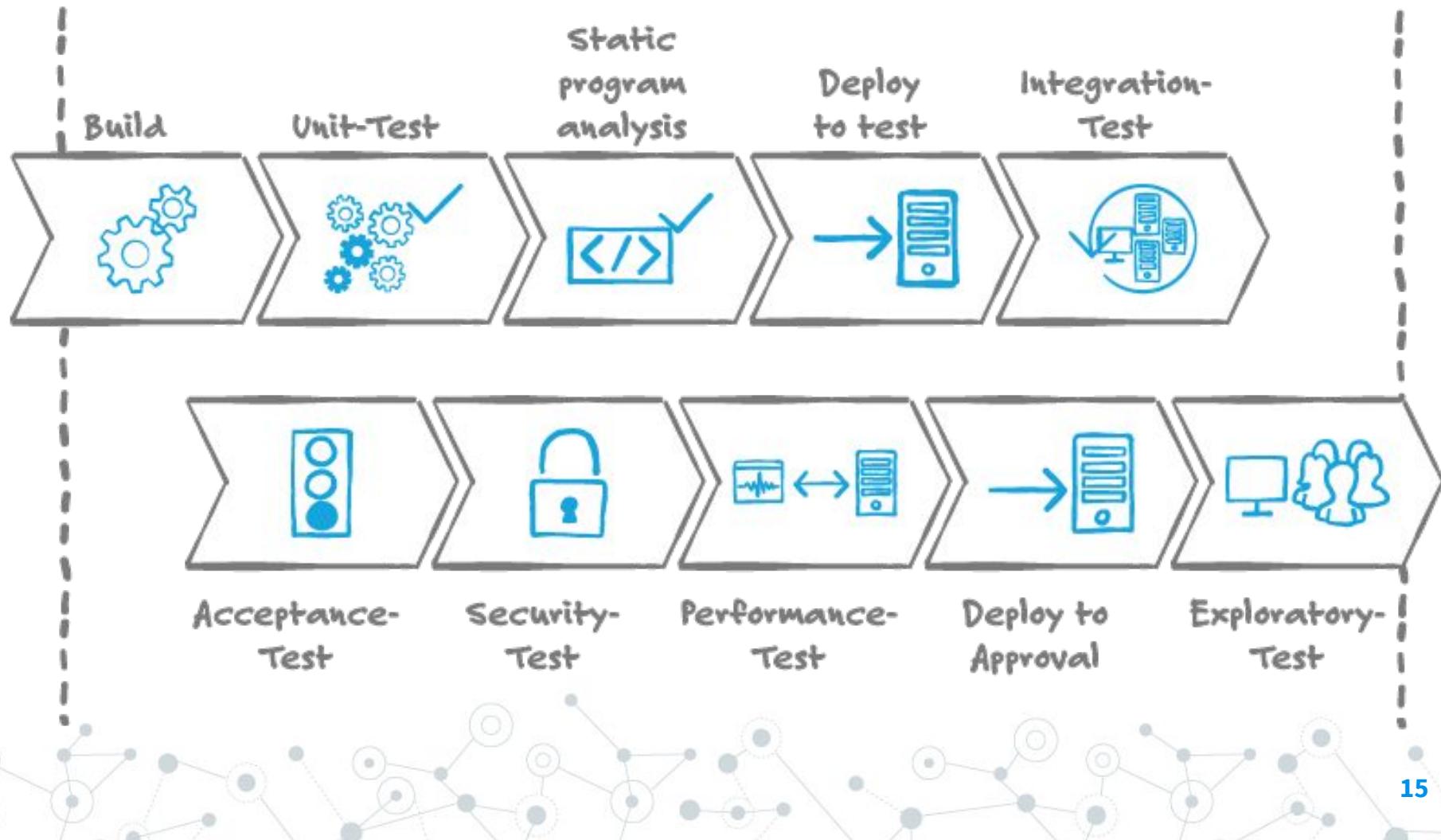
“

Deploy Contínuo é a prática de Engenharia de Software de gerar várias atualizações incrementais de software em produção em fluxo contínuo de 10 à 1000 deployments por dia



Maturidade

Entrega/Deploy Contínuos - Etapas do pipeline customizadas

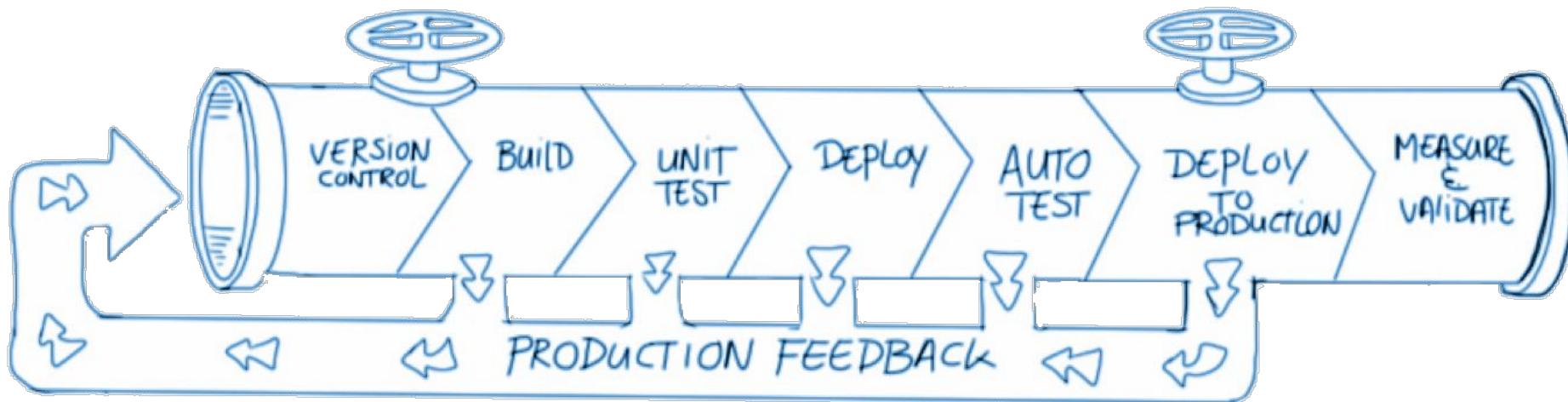


Entrega Contínua

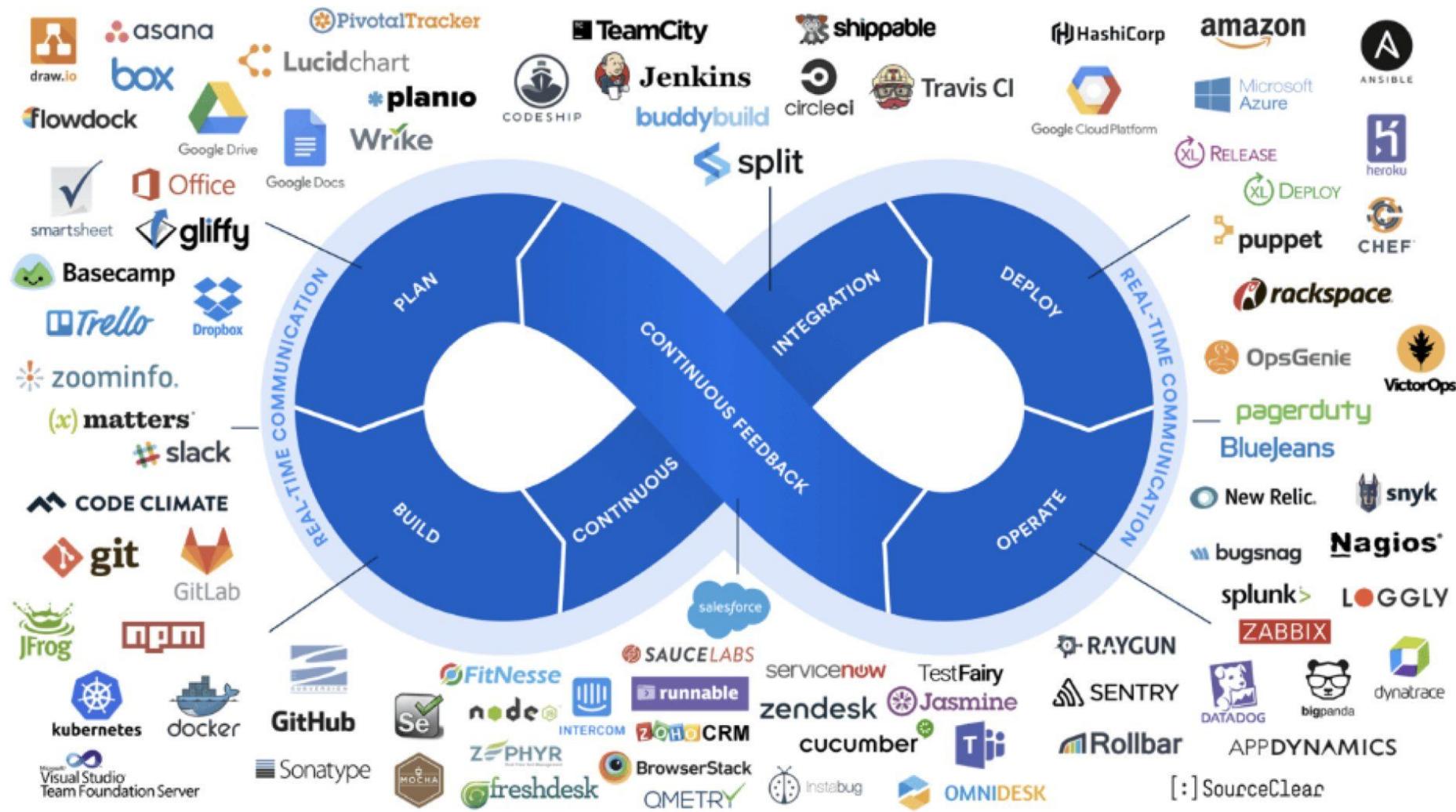
Como desenvolver um
projeto de software com
entrega/deploy
contínuos?

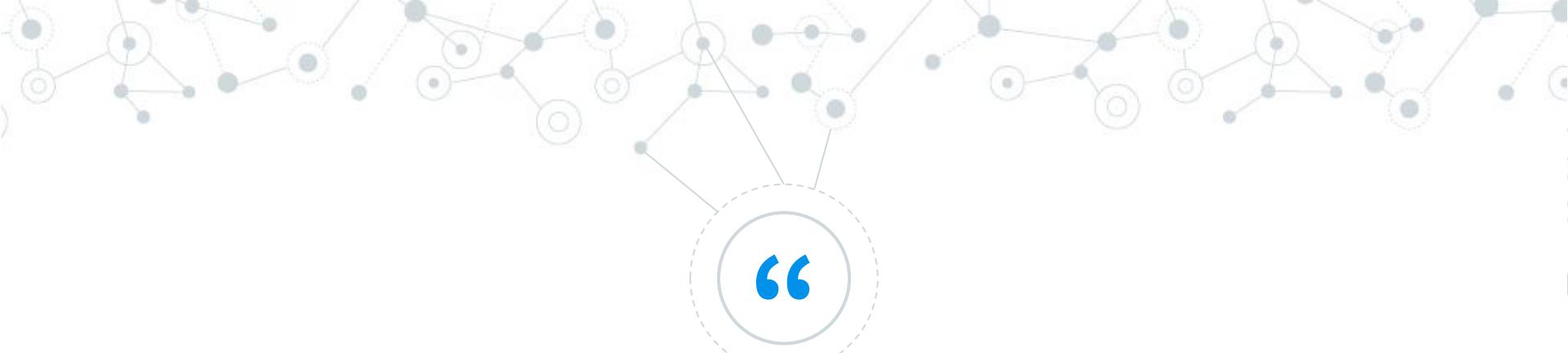


Entrega/Deploy Contínuos - Importância do feedback/automação



Ferramentas de Automação





“

Devops é um conjunto de práticas para integração entre as equipes de **desenvolvimento de softwares, operações** (infraestrutura ou sysadmin) e de **apoio envolvidas** (como controle de qualidade) e a adoção de **processos automatizados** para produção rápida e segura de aplicações e serviço



PERIODIC TABLE OF DEVOPS TOOLS (V2)

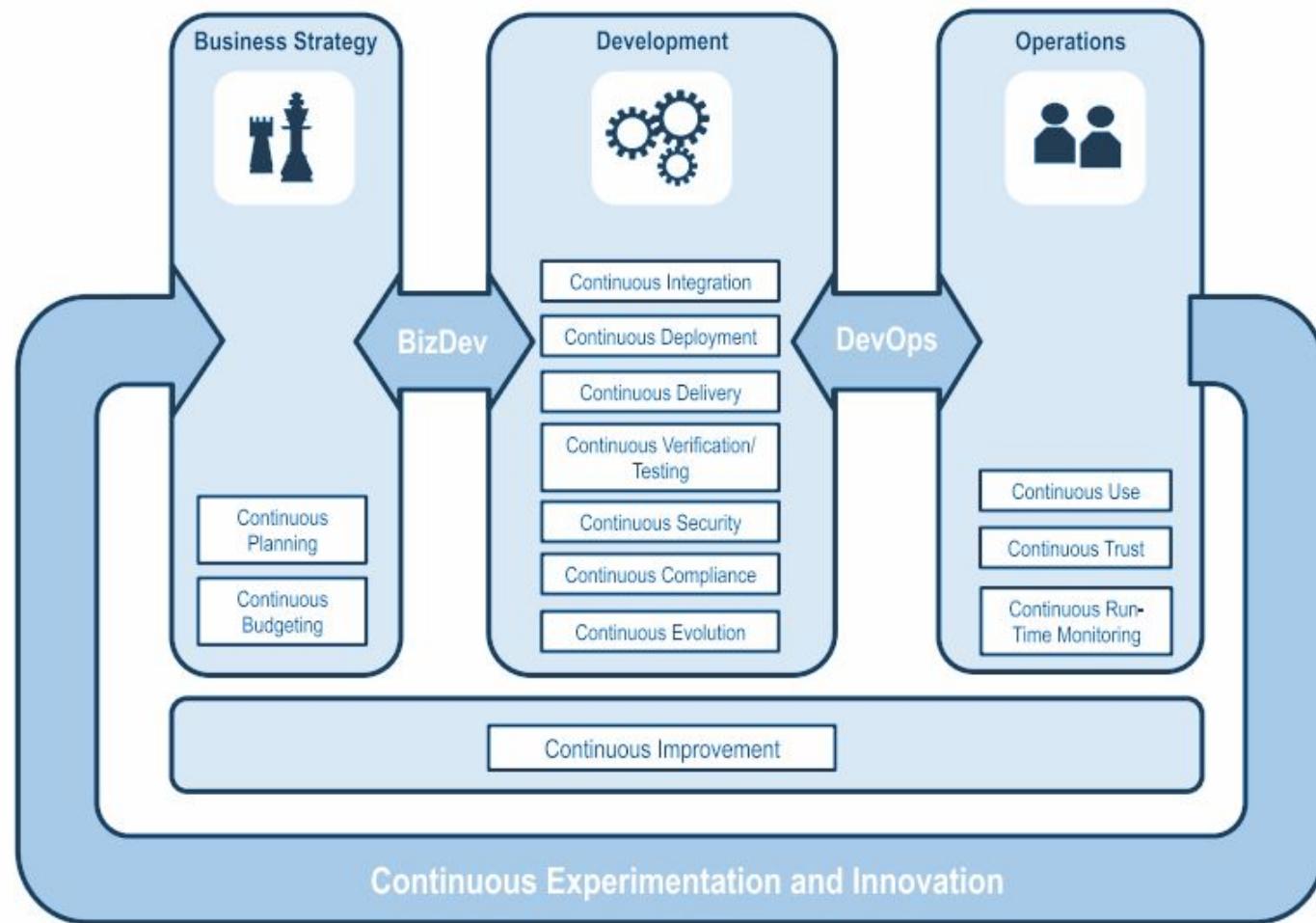
EMBED DOWNLOAD ADD

1 Fm Gh Github	2 Fm Aws Amazon Web Services
3 Os Gt Git DBmaestro	4 Pd Dm Bitbucket Liquibase
5 Fr Fm Paid Enterprise	6 En An Ansible
7 Os Bb Bitbucket	8 Os Sl Salt
9 Os En Os Lb Liquibase	10 Os Dk Docker
11 Fm En	12 Os En
13 Os Ot Otto	14 En Bl BladeLogic
15 Os Va Vagrant	16 Fr Tf Terraform
17 Os Rk Rkt	18 En Gc Google Cloud Platform
19 Os Gl GitLab	20 En Rg Redgate
21 Os Mv Maven	22 Os Gr Gradle
23 Os At ANT	24 Os Fn FitNesse
25 Fr Se Selenium	26 Os Ga Gatling
27 Fr Dh Docker Hub	28 Os Jn Jenkins
29 Pd Ba Bamboo	30 Os Tr Travis CI
31 Os Gd Deployment Manager	32 Os Sf SmartFrog
33 Os Cn Consul	34 Os Bc Bcfg2
35 Os Mo Mesos	36 En Rs Rackspace
37 Os Sv Subversion	38 En Dt Datical
39 Os Gt Grunt	40 Os Gp Gulp
41 Os Br Broccoli	42 Fr Cu Cucumber
43 Os Cj Cucumber.js	44 Fr Qu Qunit
45 Os Npm npm	46 Fm Cs Codeship
47 Pd Vs Visual Studio	48 Fr Cr CircleCI
49 Fr Cp Capistrano	50 Fr Ju JuJu
51 Os Rd Rundeck	52 Os Cf CFEngine
53 Fr Ds Swarm	54 Os Op OpenStack
55 Os Hg Mercurial	56 En Dp Delphix
57 Fr Sb sbt	58 Os Mk Make
59 Os Ck CMake	60 Fr Ju JUnit
61 Fr Jm JMeter	62 Fr Tn TestNG
63 Os Ay Artifactory	64 Fm Tc TeamCity
65 Fr Sh Shipable	66 Os Cc CruiseControl
67 En Ry RapidDeploy	68 Fr Cy CodeDeploy
69 En Oc Octopus Deploy	70 En No CA Nolio
71 Os Kb Kubernetes	72 Fm Hr Heroku
73 En Cw ISPW	74 En Id Idera
75 Os Msb MSBuild	76 Os Rk Rake
77 Fr Pk Packer	78 Os Mc Mocha
79 En Xltv XL TestView	80 Os Jm Jasmine
81 Os Nx Nexus	82 Os Co Continuum
83 Fm Ca Continua CI	84 Pd So Solano CI
85 En Xld XL Deploy	86 En EB ElectricBox
87 Fr Dp Deploybot	88 En UD UrbanCode Deploy
89 Os Nm Nomad	90 En Os OpenShift
91 En Xlr XL Release	92 En Ur UrbanCode Release
93 En Bm BMC Release Process	94 En Hp HP Cedar
95 En Au Automic	96 En Pl Plutora Release
97 En Sr Serena Release	98 Pd Tfs Team Foundation
99 Fm Tr Trello	100 Pd Jr Jira
101 Fm Rf HipChat	102 Fm Sl Slack
103 Fm Fd Flowdock	104 Pd Pv Pivotal Tracker
105 En Sn ServiceNow	106 Os Ki Kibana
107 Fm Nr New Relic	108 Os Ni Nagios
109 Os Zb Zabbix	110 En Dd Datadog
111 Os El Elasticsearch	112 Os Ss StackState
113 En Sp Splunk	114 Fm Le Logentries
115 Fm Sl Sumo Logic	116 Os Ls Logstash
117 Os Gr Graylog	118 Os Sn Snort
119 Os Tr Tripwire	120 En Ff Fortify

XebiaLabs
Deliver Faster

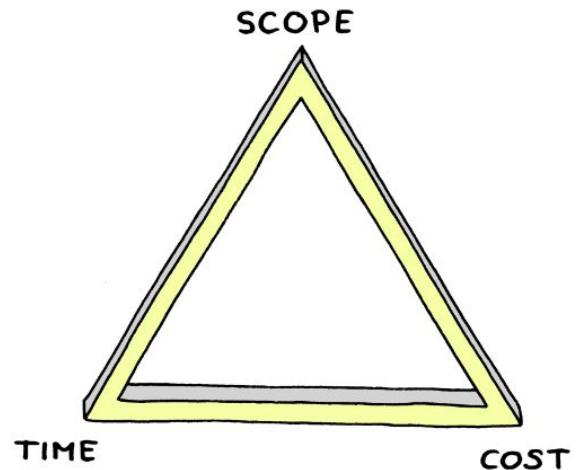
Follow @xebialabs

Contínuos - Visão Holística de atividades do negócio, desenvolvimento, operação e inovação



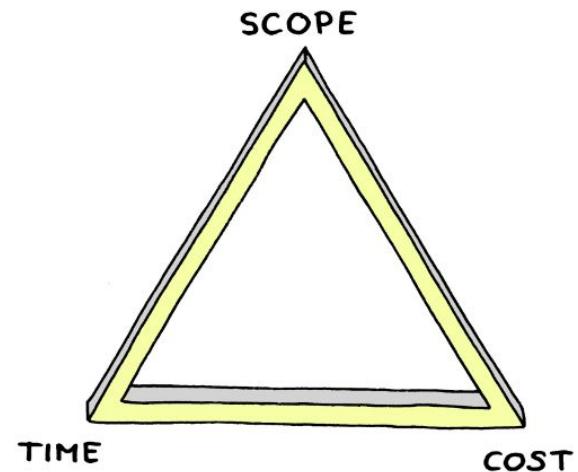
Entrega Contínua

Como garantir a entrega
contínua a partir das
restrições do projeto?



“

Gestão de Projeto deve atender à
três restrições: **escopo**, **custo** e
tempo





Gestão de projeto inclui o
planejamento, organização,
monitoramento, controle, a
formação pessoal de operações
relacionadas a um projeto



“

Projeto é um conjunto **único** de atividades planejados para gerar um **produto específico** dentro de um **intervalo de tempo** e **recursos limitados**

AGILE SERVICE DELIVERY

VISION AND GOALS

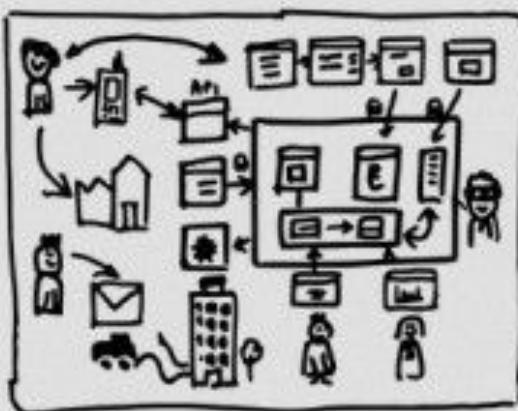
A GOOD VISION FIRES YOU UP!
[INSERT GOOD LEADERS HERE]

SET GOALS THAT DESCRIBE OUTCOMES

- ➡ ACHIEVE THIS OUTCOME
- ➡ AND THIS OUTCOME
- ➡ AND THIS OUTCOME...

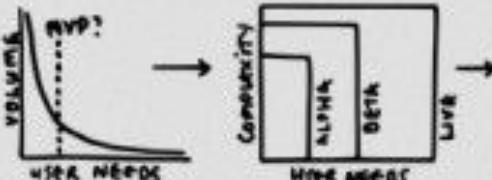
DRAW THE THING!

A PICTURE PAINTS A THOUSAND WORDS...



SCOPE

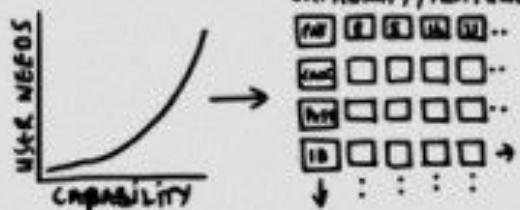
... TYPICALLY 80% OF USERS
NEED 20% OF SERVICE FEATURES



... START WITH THE HARD STUFF

CAPABILITIES

... DELIVER USER NEEDS AND ARE
THE BUILDING BLOCKS OF THE
SERVICE.



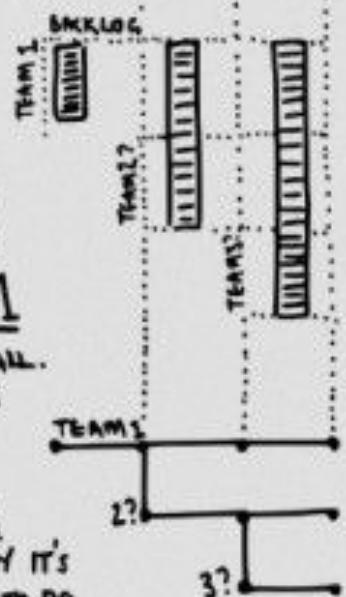
CC BY http://jamiearnold.com

ROADMAP

... ASK 7 QUESTIONS

3m 6m 12m...

1 TRYING TO PROVE?	?	?	?
2 WHICH USERS?	?	?	?
3 WHAT OPERATING?	?	?	?
4 WHAT SAYING?	?	?	?
5 ASSUMPTIONS?	?	?	?
6 DEPENDENCIES?	?	?	?
7 WHAT CAPABILITIES? (... NOT JUST THE SOFTWARE)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



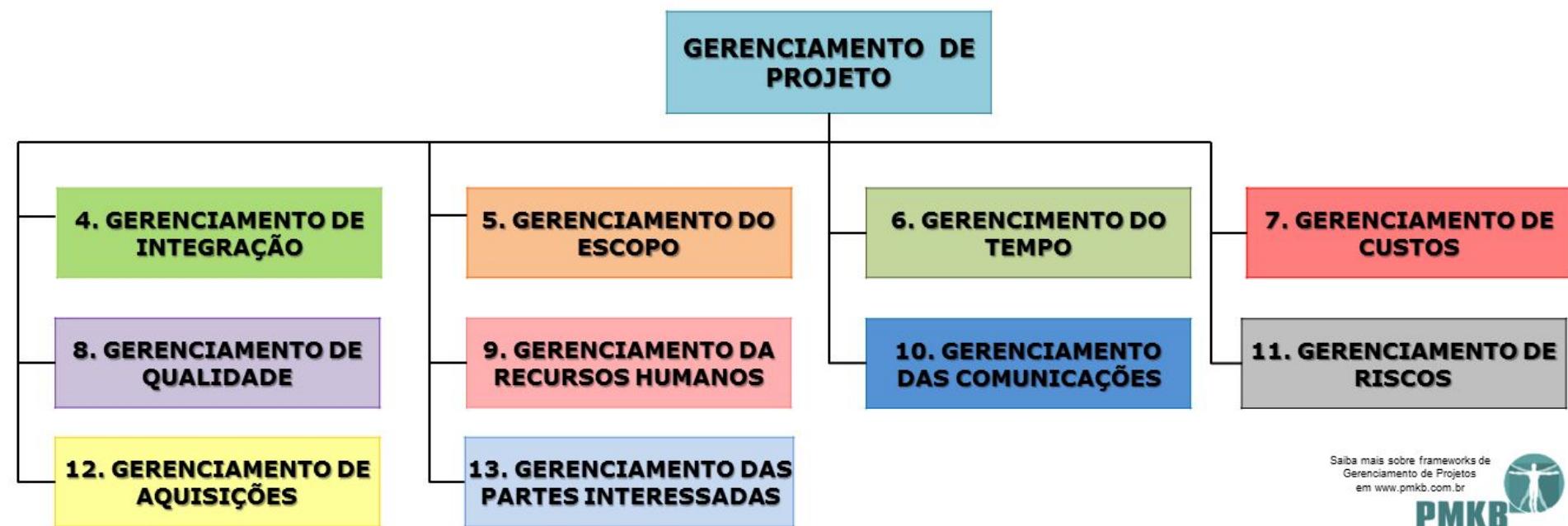
GROWING

THE TEAM

ALWAYS START SMALL.
TEAM CULTURE IS
PRECIOUS.

... ONLY SCALE
WHEN THE TEAM
AND BACKLOG SAY IT'S
THE RIGHT THING TO DO

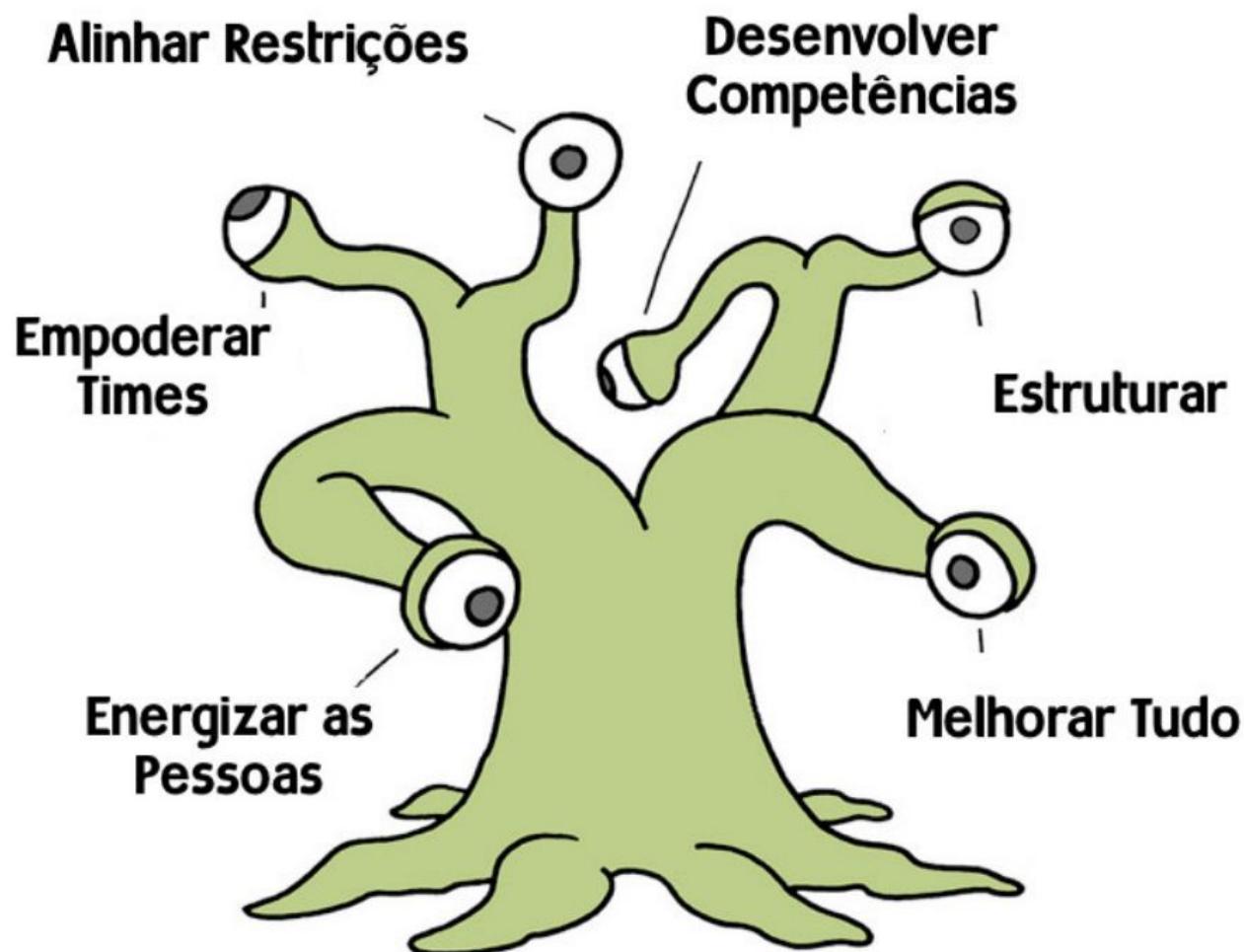
Modelo de Gestão PMBOK



Saiba mais sobre frameworks de
Gerenciamento de Projetos
em www.pmbok.com.br



O modelo de Gestão 3.0





“

“O que é medido é gerido” – Peter Drucker

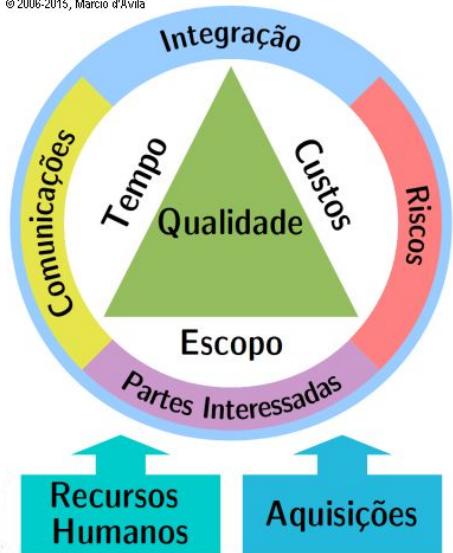
Gestão de Projeto - Algumas modificações para projeto de software

**Dividir Custo
(recursos) em
Pessoas e
Ferramentas**

**Dividir escopo
em
funcionalidade
e qualidade**

**Incluir
dimensões para
Processo e Valor
(de negócio)**

© 2006-2015, Márcio d'Ávila



As 7 Dimensões dos Projetos de Software

1. Pessoas
2. Funcionalidades
3. Qualidade
4. Ferramentas
5. Tempo
6. Valor
7. Processo

Pessoas

cross-functional

colocation

interações

respeito

times pequenos

colaboração

responsabilidade

confiança

auto-organização

Funcionalidade

- minimal marketable features
- envolvimento do cliente
 - estórias de usuário
 - demonstrações
 - backlogs
 - “inch-deep, mile-wide”
 - critérios de aceitação

Qualidade

test-driven development

excelência técnica

definição de pronto

design emergente

refactoring

simplicidade

programação em par

Ferramentas

quadros brancos

sticky notes

builds diários

controle de versão

integração contínua

testes automatizados

burn charts

Tempo

rolling wave planning

timeboxes

potentially shippable products

planejamento de release iterações

sprints

ritmo sustentável

Valor

increments

priorização

feedback

value streams

embracing change

mapeamento de valor

entregas frequentes

Processo

sprint planning

stand-up meetings

código fonte coletivo

retrospectivas

velocidade

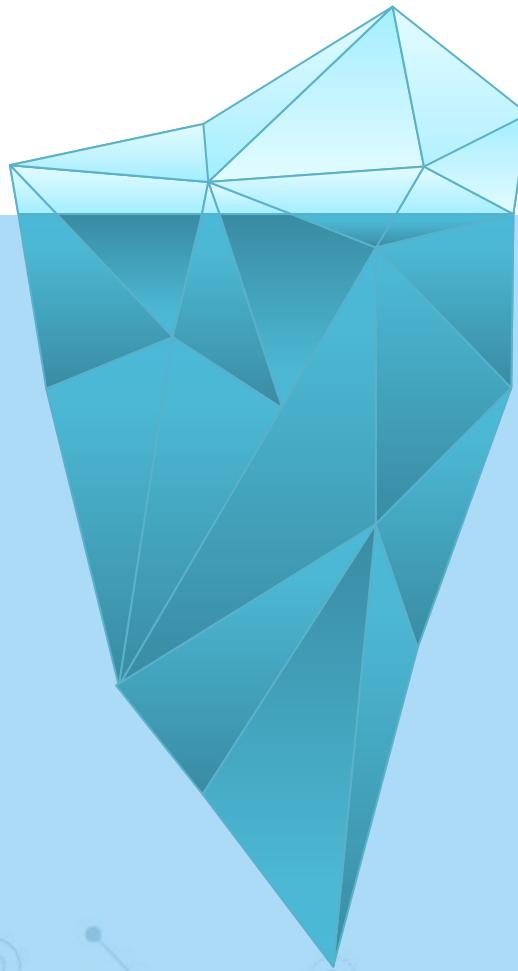
planning poker

spikes

Princípio para Entregas de Baixo Risco

- Incremental é melhor que “Big Bang”
- Implantação != Entrega
- Foque em entregar lotes pequenos
- Qualidade deve estar embutida no processo

Desenvolver funcionalidades é somente a “ponta do Iceberg” de um produto de software



Devops/BizDev

Práticas e ferramentas para viabilizar a entrega, deploy contínuo de software

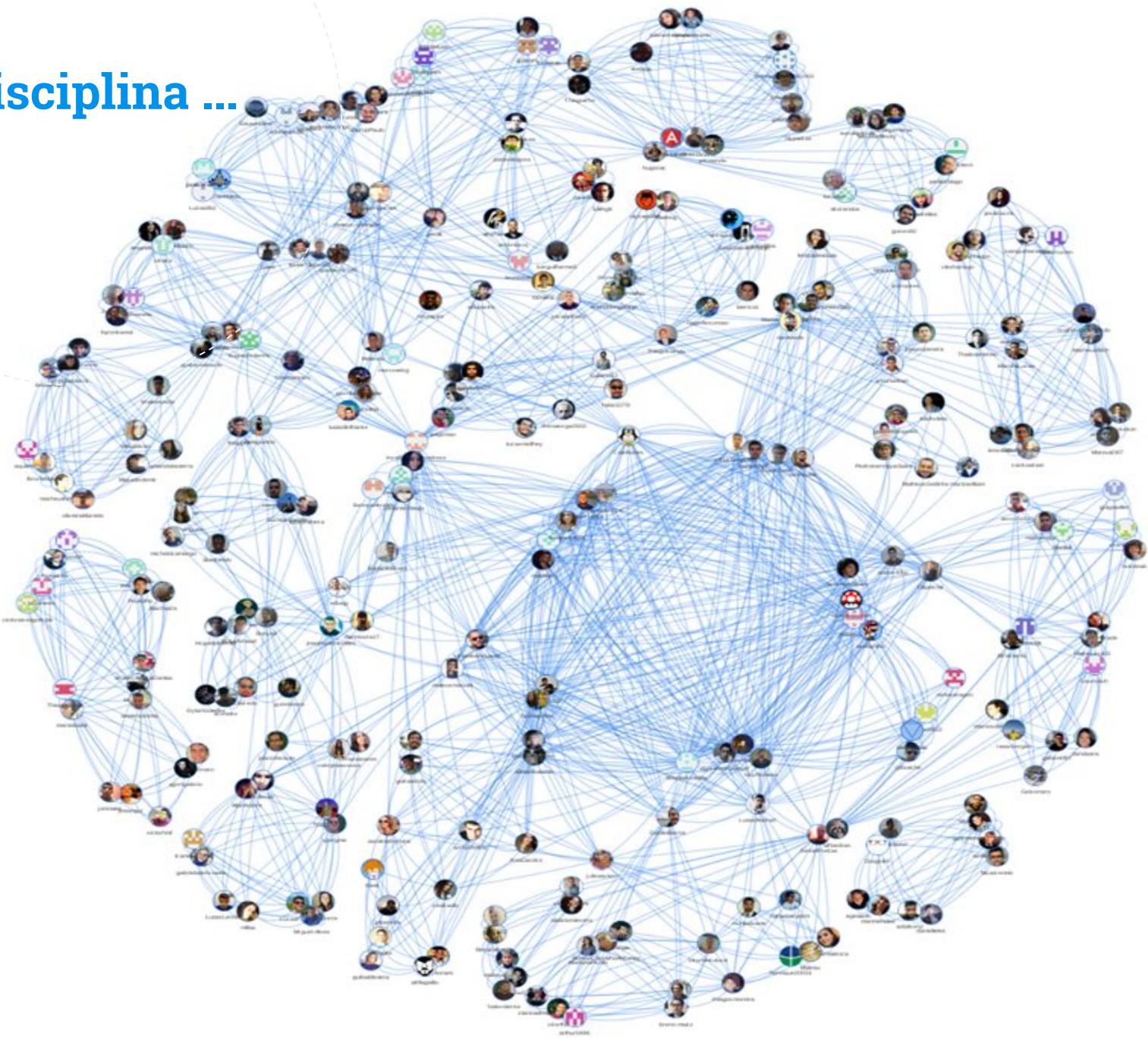
Dev

Planejamento/Desenvolvimento de novas funcionalidades de acordo com as práticas ágeis (qualidade, testes)

Gestão de Projeto de Software

Gestão do escopo, tempo, custos, qualidade, recursos materiais, pessoas, comunicações, riscos para viabilizar a entrega contínua de software sempre agregando o máximo de valor, e dentro das restrições impostas.

A disciplina ...



Lição #3:

“Sejamos claros: **Sua carreira** é **sua**
responsabilidade, seu empregador **não é sua**
Mãe” – Robert C. Martin

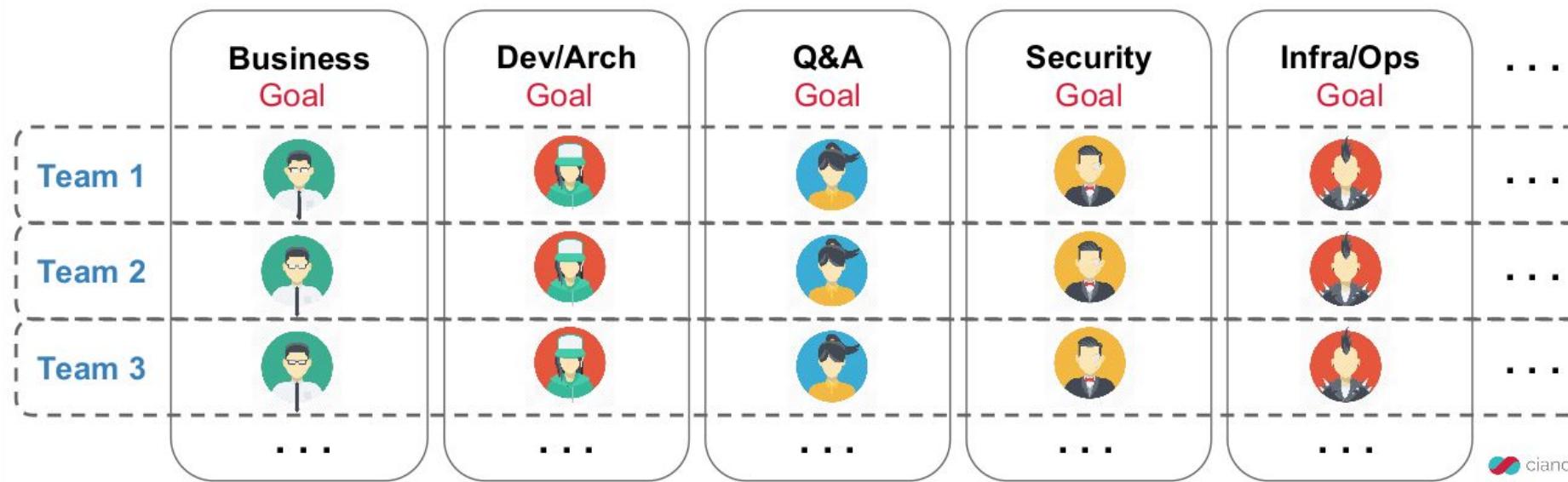


#3 Time



“

Times multifuncionais (máximo de 10 pessoas por time)



Times

- PM* - Product Manager (Antigo PO)
- TL* - Tech Leader (Antigo Scrum Master)
- UX* - Designer
- QA* - Qualidade
- DV* - Devops
- DS* - Data Science
- SE** - Engenheiros de Software

O time deve ter
Conhecimento e autonomia
sobre determinada temática
do projeto

* Desempenhado por EP

** Desempenhado por MDS



Mudança #1:

Coach/monitor faz parte do time com o papel de
PO

Coach não influencia mais na **avaliação** dos grupos

Papel dos coaches: potencializar o desempenho dos times





“

Clientes Técnicos



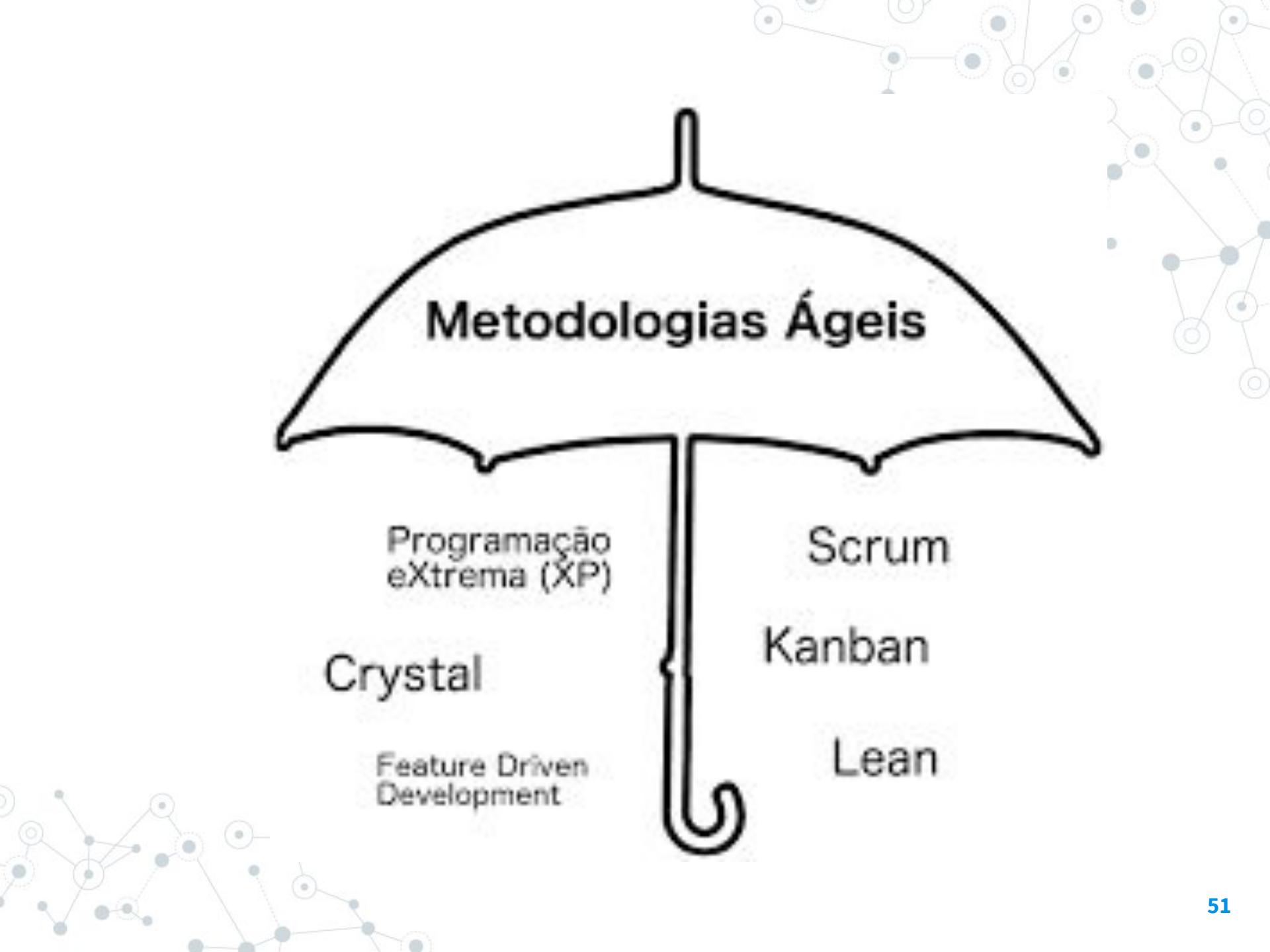
Ter o cliente por perto agiliza o processo de decisão



É hora de rever o método

Modelo de Processo

- Vamos aplicar métodos e práticas ágeis
- Vamos adaptar lean startup, scrum, XP, Kanban
- Vamos aplicar práticas XP, práticas Devops
- Vamos aderir a alguns artefatos do RUP/PMBOK (que se adequa ao problema)
- Vamos aplicar arquitetura de Microsserviços



Metodologias Ágeis

Programação
eXtrema (XP)

Crystal

Feature Driven
Development

Scrum

Kanban

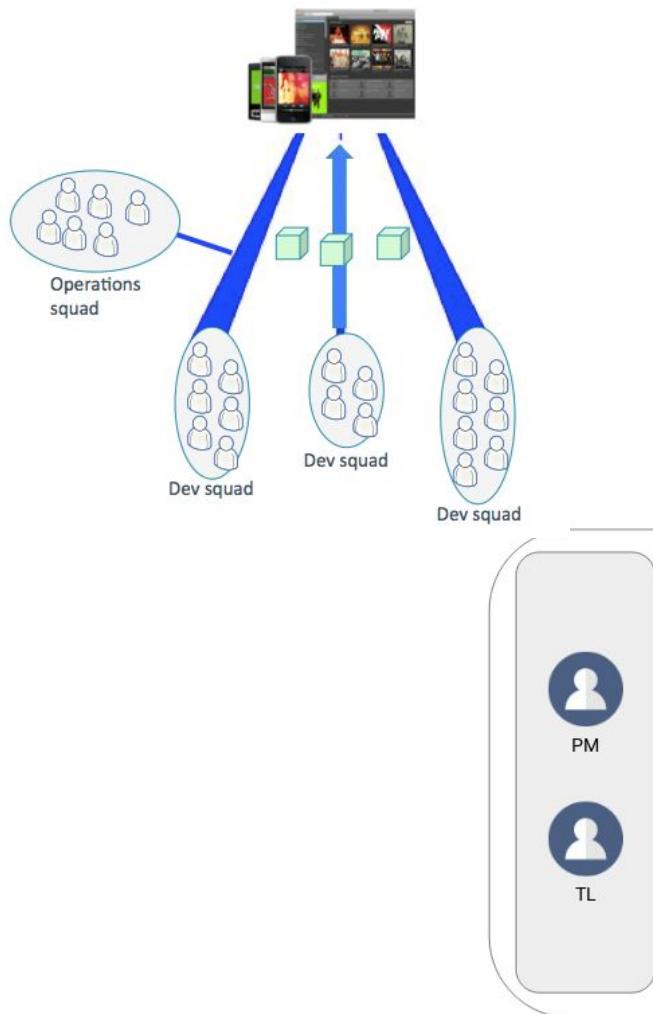
Lean

Mudança #2:

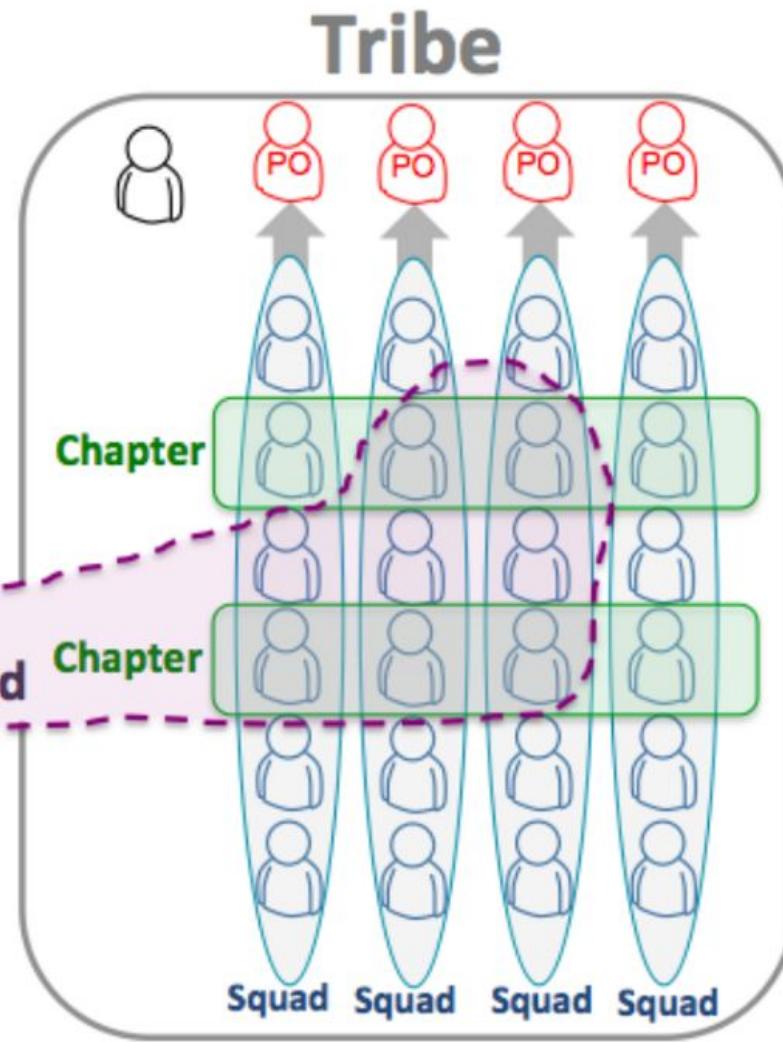
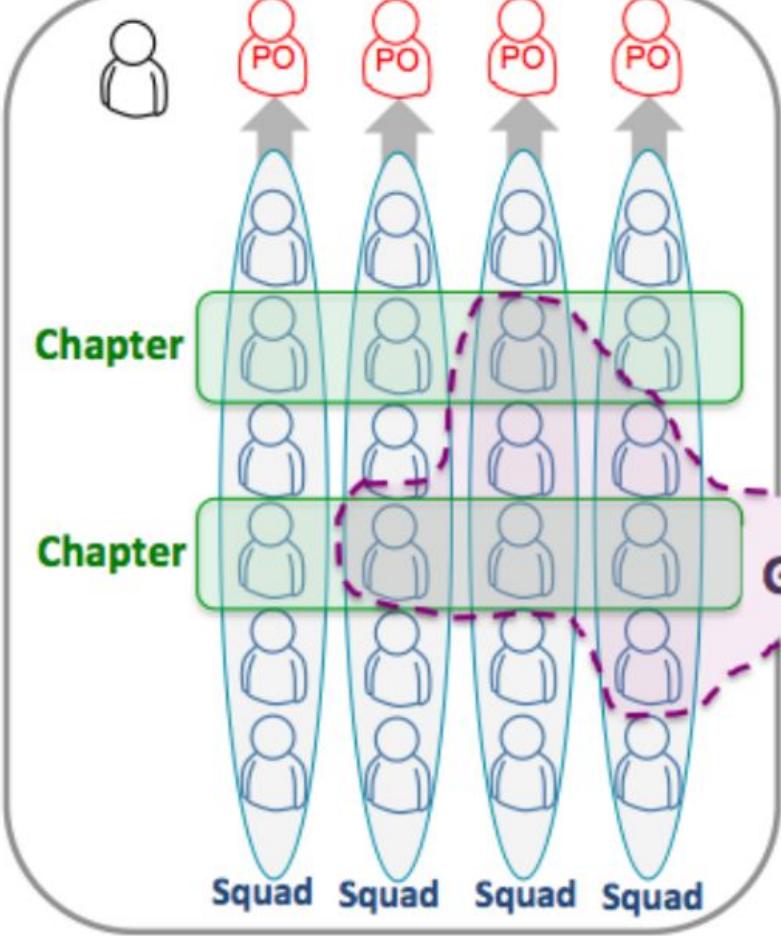
Os métodos ágeis e práticas ágeis serão exercitadas por toda a disciplina

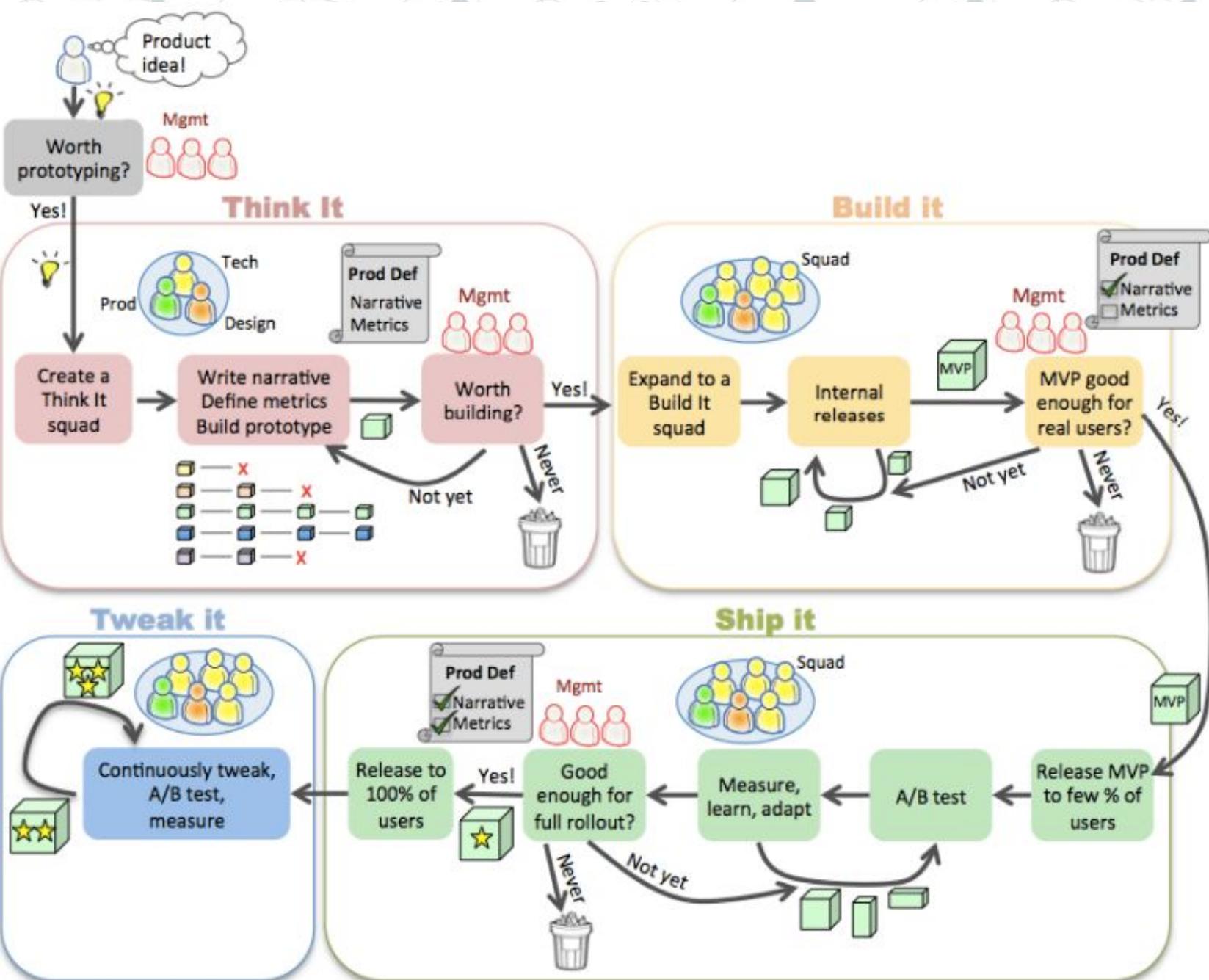
Mais importante que os métodos, são as práticas!



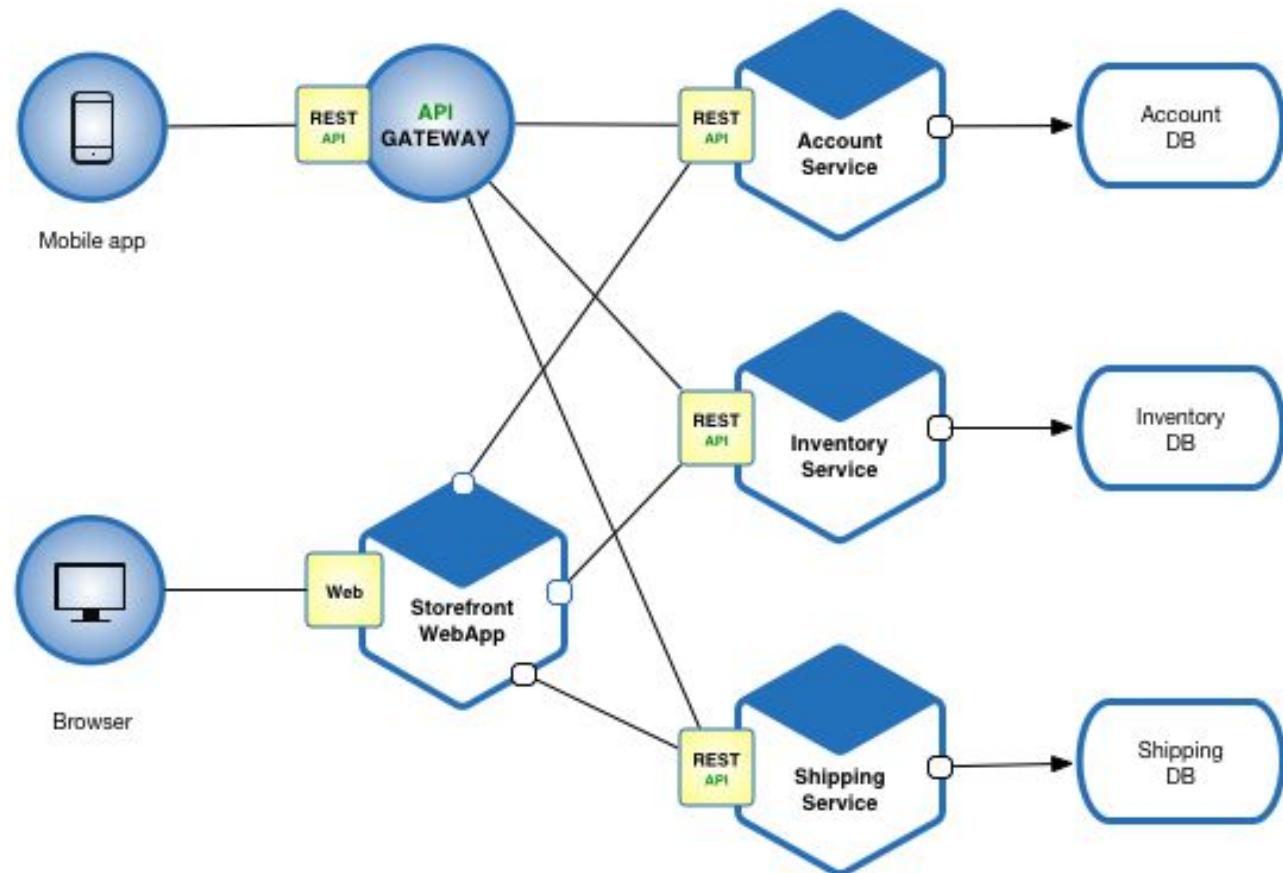


Squad





Arquitetura MicroServiços



Regras Gerais GPP/MDS 2018.1



Quantidade de Sprints

Serão ao total X sprints



Artefatos

- Termo de Abertura de Projeto (TAP)
- Estrutura Analítica de Projeto (EAP)
- Plano de Ger. Comunicação
- Plano GCS
- Plano Qualidade
- Plano de Riscos



Horas trabalhadas

Deve-se planejar, no mínimo, 10 horas por semana por integrante. Durante TODA a disciplina, deve-se manter o monitoramento das horas trabalhadas por integrante

<https://www.bitrix24.com.br/features/tasks.php>



Releases

Serão realiadas 3 Releases

- R1:
- R2:
- R3:



Registro das Sprints

O registro de acompanhamento das sprints devem ser disponibilizados na wiki do projeto e nas paredes da sala de aula



Critérios de Avaliação

- Provas
- Projeto
- Contribuição na Wiki da Disciplina

Maiores detalhes, olhar o plano de ensino no repositório da disciplina

Principais Problemas/Riscos na Disciplina (relatadas por grupos de outros semestres)

1. Falta de compromisso de membros (ou trabalha 10 horas por semana ou não dá para entregar o projeto)
2. Falhas na comunicação
3. Não ouvir a professora
4. Não assumir riscos quando necessário (reagir tarde)

Mandamentos GPP/MDS

1. Faça o que GPP sugerir
2. GPP: acredite na Carla!
3. Não se apegue à linguagens (é só mais uma ferramenta)
4. As 10 horas semanais são sagradas!
5. Tenha calma^{*5}
6. Divirta-se



software engineering

engineering quality generic although engineers computer practice actual fields classical improve definition sufficient organically certification undergraduate becoming sister future licensing quantifiable purpose science field's many contentious affordable degrees started international approach operation experience disciplined bright standard due post-graduate issue

precision object-oriented still professional young much lack just development build however new conforms grown see pre-requisite systematic compared necessarily especially magazine

higher paradigm subsume term application field study maintainable maintenance knowledge debate

technology system questionable approaches programs defined quicker years public technical around protect profession introduction continued

education tool study maintainable maintenance knowledge debate

programming field creating testing according viewing money

planning since looks curriculum impact main dedicated engineer change



Agile Development

clients flow
team growth quality finishing technology
developer requirements fast testing release cycle
programmer work change test system information
organization integration plan process collaboration business
project review computer code innovation strategy
leadership management web implementation continuous data response
analysis backlog deployment brainstorming lifecycle
model control iterative method improvement methodology
speed progress analytics engineering
division master coding

OPEN SOURCE

A word cloud centered around the concepts of Open Source and Open Software. The words are arranged in two main columns: 'OPEN SOURCE' on the left and 'OPEN SOFTWARE' on the right. Various terms related to these concepts are scattered throughout the cloud.

OPEN SOURCE:

- COST
- PUBLIC
- COPYRIGHT
- ACCESS
- DESIGN
- INCLUDING
- SCIENTIFIC
- WORLD
- PROJECTS
- INDIVIDUALS
- LINUX
- BLOGS
- OPERATING
- COMMONS
- CENTURY
- PROJECT
- NETWORK
- BUSINESS
- CREATIVE
- SHARING
- USERS
- RAYMOND
- PRODUCTION
- ONE
- CULTURAL
- HARDWARE
- INNOVATION
- AVAILABLE
- RESEARCH
- CODE
- EXAMPLES
- INTELLECTUAL
- PROPERTY
- GENERAL
- FIRST
- CULTURE
- DEVELOPMENT
- IDEA
- COMPANIES
- COMMUNITY
- MEDIA
- INTERNET
- TERM
- DATA
- ONLINE
- LICENSE
- DIGITAL
- SOCIAL
- USING

OPEN SOFTWARE:

- POTENTIAL
- COMMUNITIES
- DISTRIBUTED
- PRODUCT
- MANUFACTURERS
- SIMILAR
- TECHNOLOGY
- COSTS
- EVEN
- MOVEMENT
- MESSAGEBOARDS
- NEW
- ARTICLE
- MAKING
- CITATION
- WORK
- COMPUTER
- CASE
- POLITICAL
- IDEA
- COMPANIES
- WELL
- SCIENCE
- FIRST
- TECHNOLOGIES
- MANY
- MADE
- SCIENCE
- FIRST
- CULTURE
- DEVELOPMENT
- IDEA
- COMPANIES
- COMMUNITY
- MEDIA
- INTERNET
- TERM
- DATA
- ONLINE
- LICENSE
- DIGITAL
- SOCIAL
- USING

Sprint 0

- Planejamento dos Treinamentos
 - Comecem por git/revisão OO/Scrum/Kanban
 - Docker, Docker Composer, gitflow
- Plano de comunicação (comece pelo principal risco na disciplina!)
 - Canais de comunicação, bots de integração com as ferramentas utilizadas, Rituais de comunicação (standups)
- Metodologia ágil
 - Quando serão os rituais, onde, quem é responsável

Sprint 0

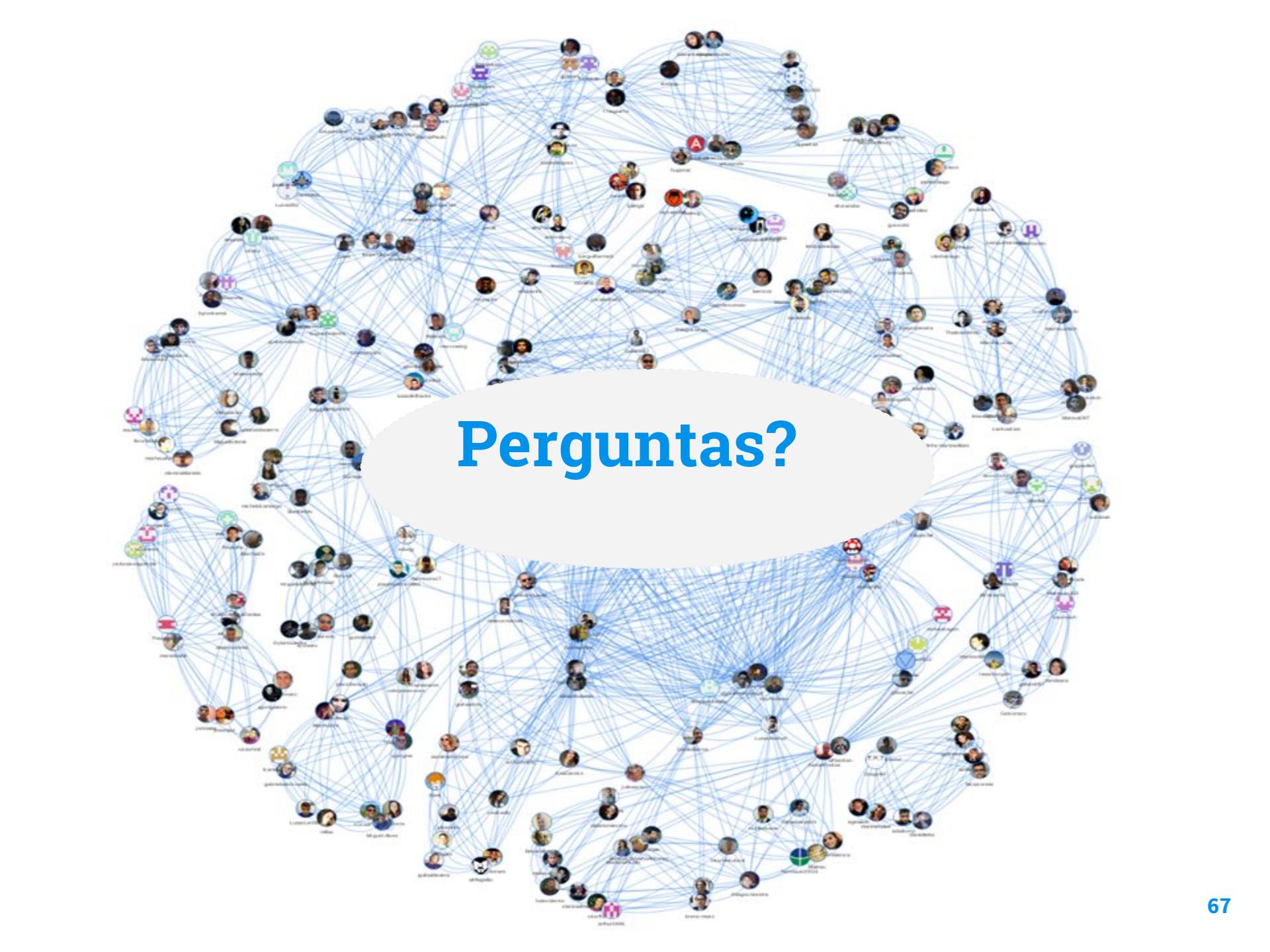


Comunidade de software Livre

- Como começar:
 - <https://opensource.guide/>
 - Nesse guia, há templates para os principais artefatos (README, código de conduta, guia de contribuição, etc)
- Procurem bons repositórios de software livre
 - <https://github.com/openmrs/openmrs-core>
 - <https://github.com/RasaHQ/>
- Checklist de projetos de software livre
 - <https://github.com/fga-gpp-mds/Falko-2017.2-FrontEnd/community>

Leituras sugeridas

- <https://github.com/fga-gpp-mds/00-Disciplina/wiki>
- Livro - "Construindo Software como Serviço (SaaS): Uma Abordagem Ágil Usando Computação em Nuvem"
- http://www.huffpostbrasil.com/2017/07/04/por-tras-da-cultura-empreendedora-dos-eua_a_23015575/?utm_hp_ref=br-homepage



Perguntas?

Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by [SlidesCarnival](#)
- Photographs by [Unsplash](#) & [Death to the Stock Photo](#)
[\(license\)](#)