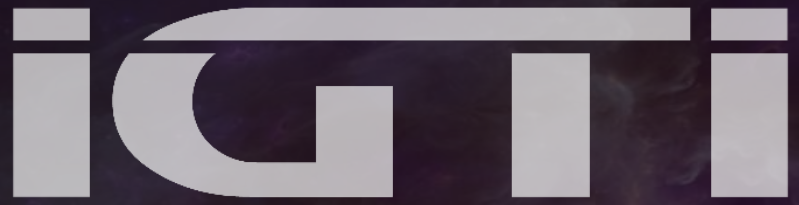


A aula interativa do **Módulo1 - Bootcamp Arquiteto de Software** começará em breve!

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.

Fundamentos

Capítulo 3: Aula Interativa 2

Prof: Junilson Pereira Souza

Fundamentos

Aula Interativa 2 – parte 1: Arquitetura Ágil

Prof: Junilson Pereira Souza

Nesta aula

- Arquiteto da Torre de Marfim vs Arquitetura Colaborativa.
- Detalhar antecipadamente vs abordagem Iterativa e Incremental.
- Arquitetura não mensurável vs mensurável

Abordagem Tradicional

Arquiteto da Torre de Marfim.

Detalhar tudo antecipadamente.

Arquitetura não mensurável.

Arquiteto da Torre de Marfim

Arquitetos que desenvolvem suas atividades isolados do dia a dia do time de desenvolvimento.

https://medium.com/it-dead-inside/knocking-down-the-ivory-tower-72fd249a8db7

Arquitetura Colaborativa

Times envolvidos nas decisões arquiteturais.

Engajamento, propriedade e autoridade.

Envolvimento também nas evoluções.

Cobertura de áreas como desempenho, monitoração, registro de eventos, deployment.

Detalhar Tudo Antecipadamente

BPUF: Big Planning Up Front.

BRUP: Big Requirements Up Front.

BDUF: Big Design Up Front.

Abordagem Iterativa e Incremental

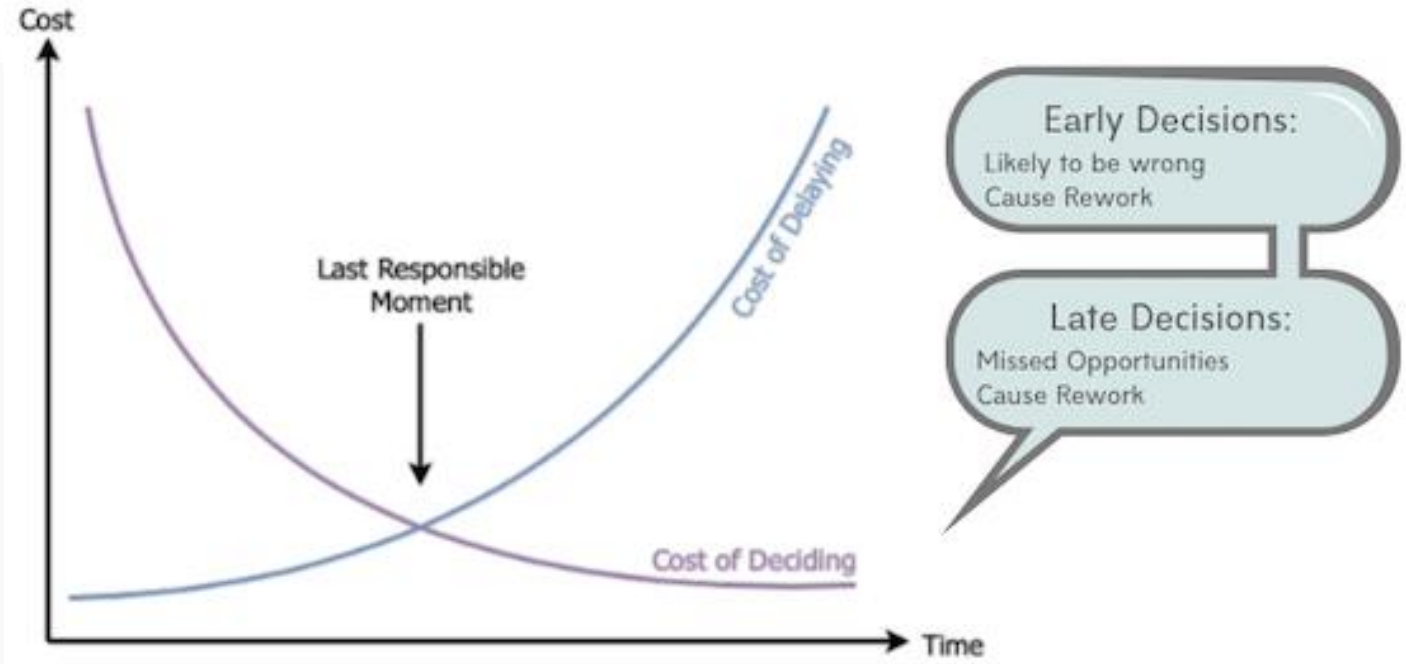
Backlog Técnico.

Last Responsible Moment.

YAGNI: You Ain't Gonna Need It

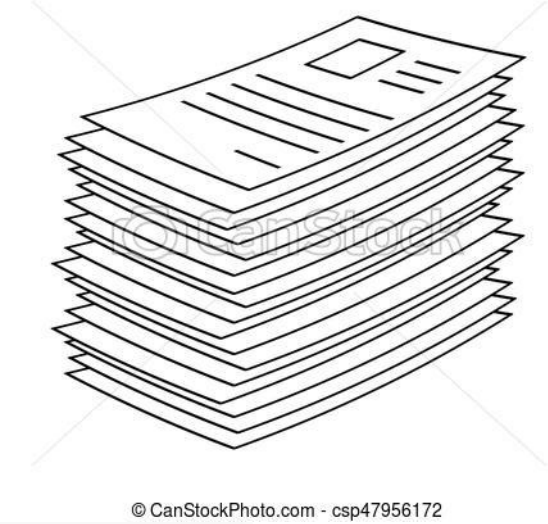
KISS: Keep It Simple, Stupid.

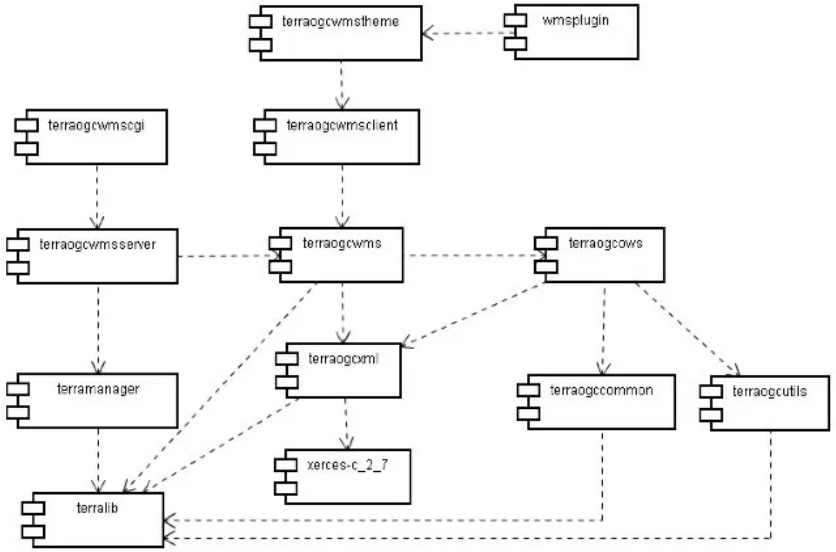
Last Responsible Moment



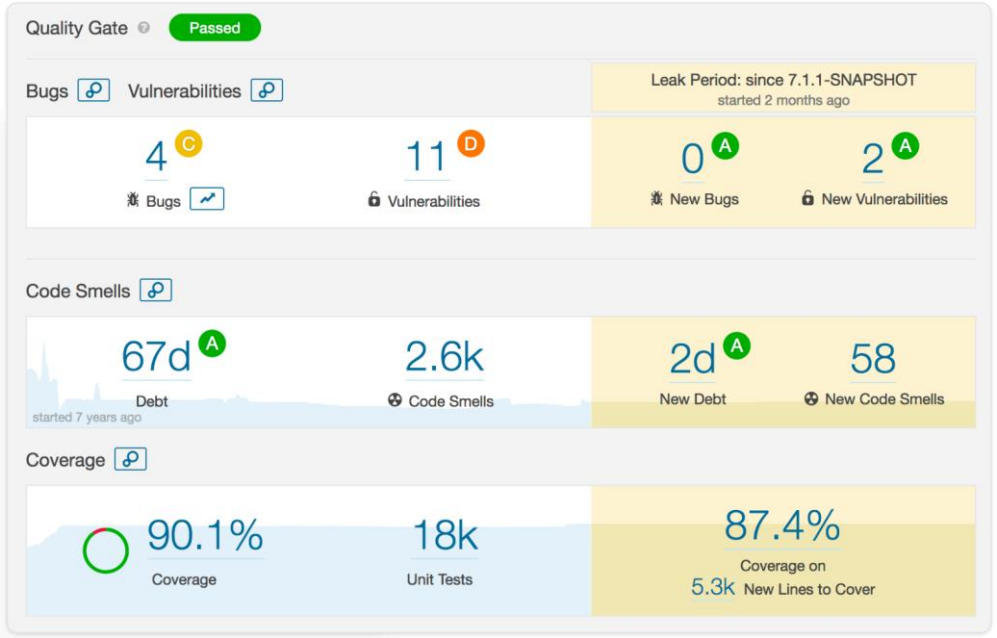
https://towardsdatascience.com/5-key-principles-of-software-architecture-e5379cb10fd5

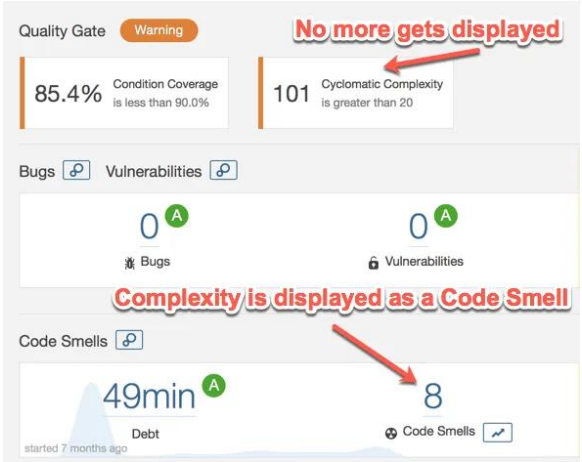
Arquitetura Não Mensurável



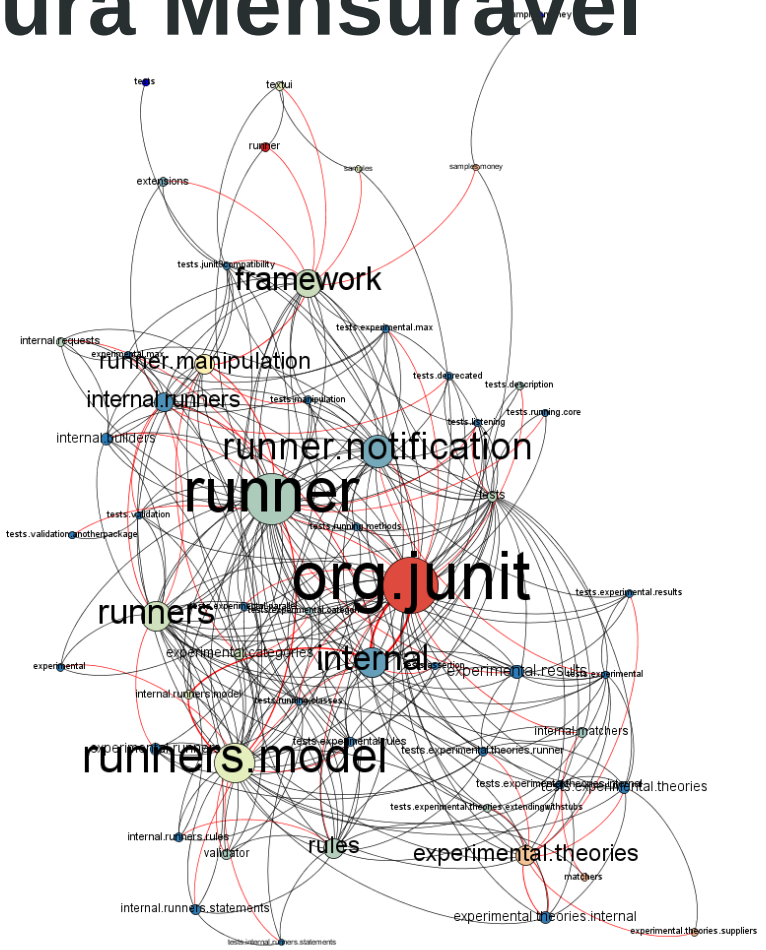


Arquitetura Mensurável





Arquitetura Mensurável



https://dzone.com/articles/visualizing-and-analysing-java

Arquitetura Mensurável

```
@Test
public void verificarViolacaoCamadas() {

ArchRule rule = layeredArchitecture()
.layer("Service").definedBy("..service..")
.layer("Persistence").definedBy("..persistence..")

.whereLayer("Persistence").mayOnlyBeAccessedByLayers("Service");

rule.check(importedClasses);

}
```

Resumindo...

Arquiteto da Torre de Marfim vs Arquitetura Colaborativa.

Abordagem antecipada vs Iterativa e Incremental.

Arquitetura não mensurável vs mensurável

Fundamentos

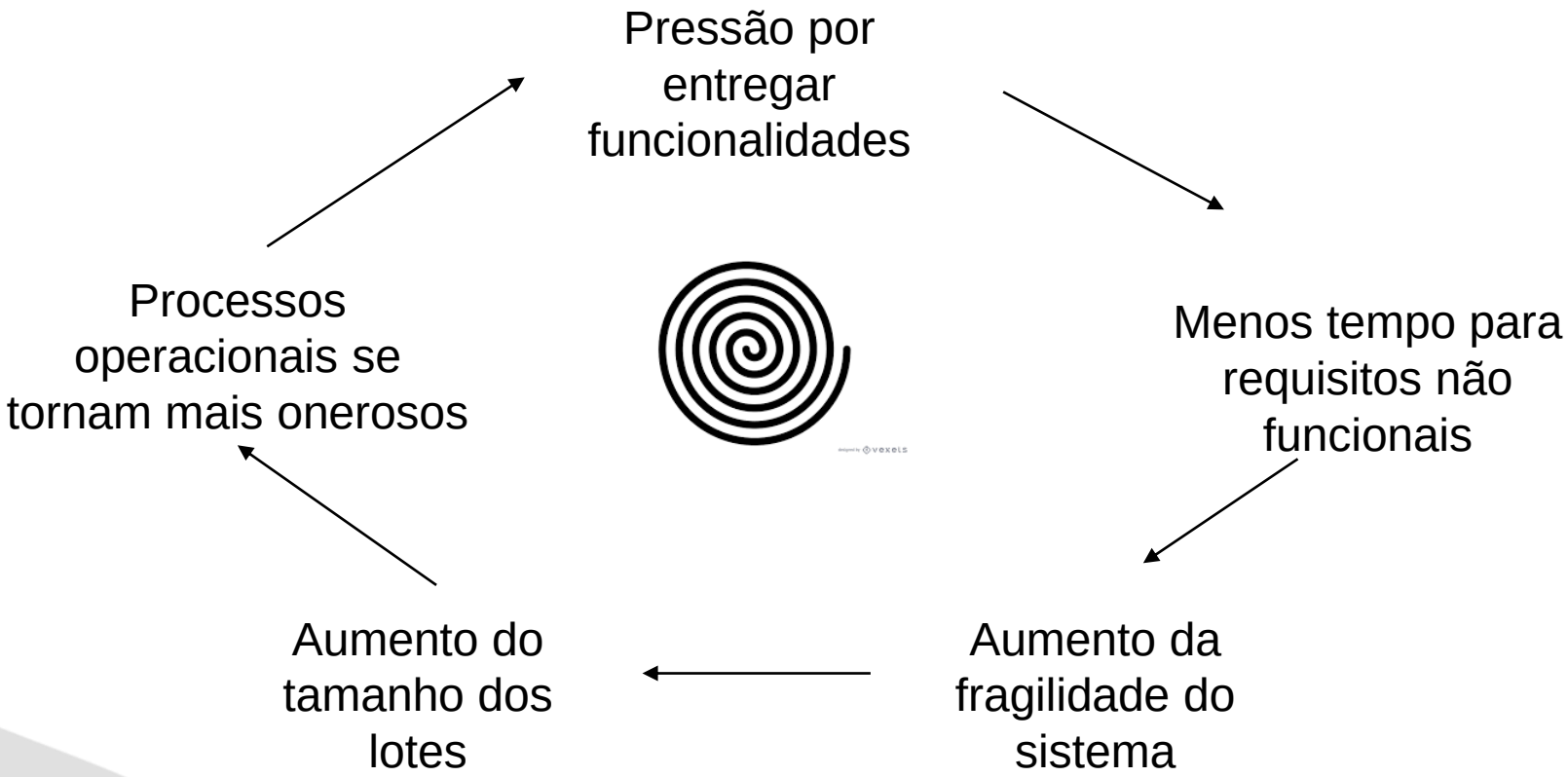
Aula Interativa 2 – parte 2: DevOps

Prof: Junilson Pereira Souza

Nesta aula

- Espiral da Morte e débito técnico.
- Continuous Integration, Delivery e Deployment.
- Arquitetura e código legado.

Espiral da Morte



[Kim, Humble, Willis, Debois]

Too Busy for Improvements?

http://rhizome.coop/taking-decisions-defining-consensus/challenges-facing-transformation-too-busy-wheel/

Causas e Efeitos do Débito Técnico

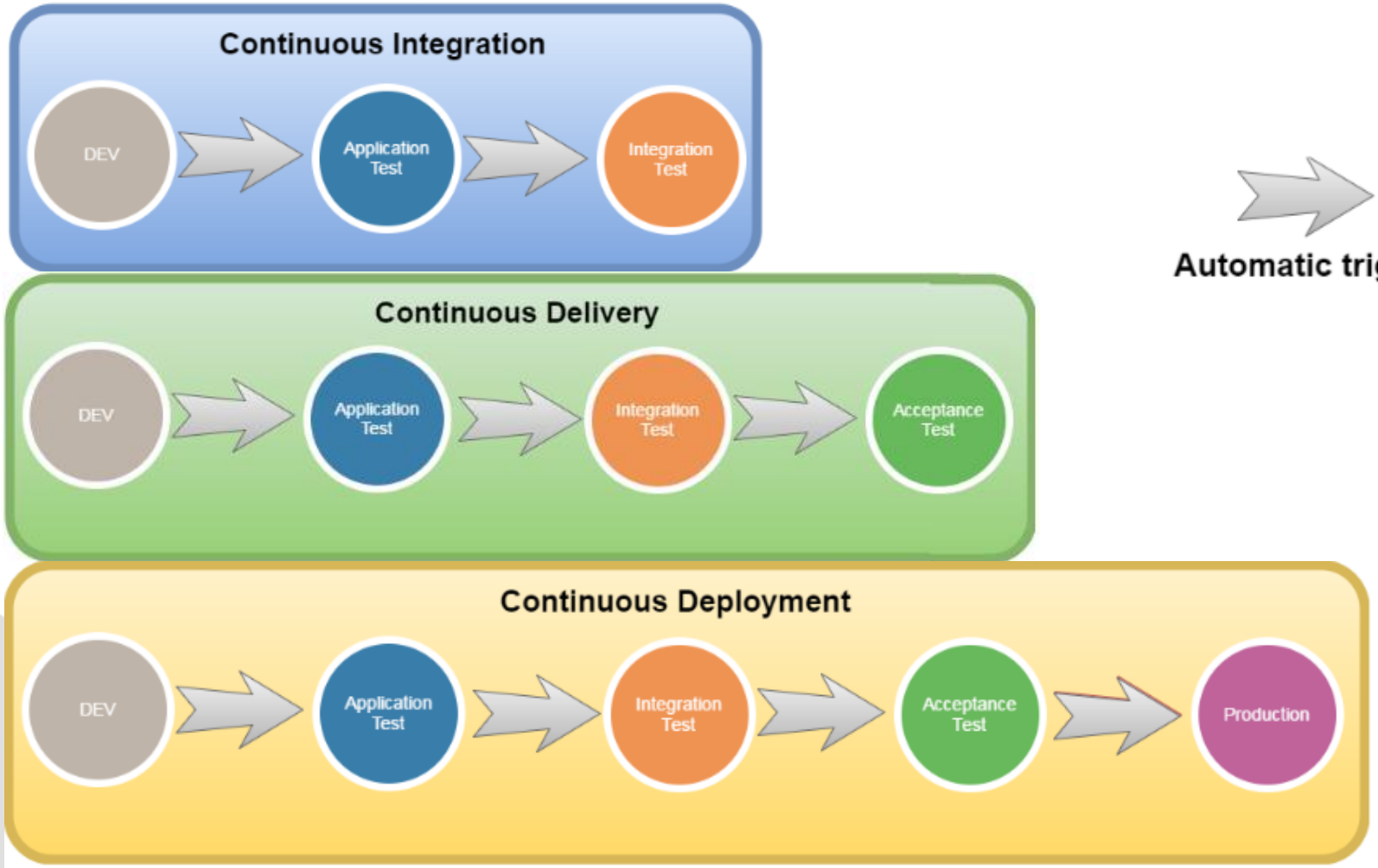
What are the causes and effects of technical debt?	

Intentional causes of technical debt	• Lack of time given for development
	• Pressure to the development team
	• Complexity of the source code
	• Business decisions
	  - Lack of technical knowledge
	  - Communication challenges

Unintentional causes of technical debt	• Lack of coding standards and guides • Junior coders • Lack of knowledge about future changes • Lack of documentation
Short-term effects of technical debt	• Time-to-market benefit • Increased customer satisfaction
Long-term effects of technical debt	• Extra working hours • Errors and bugs • Customer unsatisfaction • Complexity of the source code
What management and reduction strategies/practices are being used for technical debt?	
Practices for reducing and preventing technical debt	• Refactoring • Bug fixing days • Code reviews • Coding standards and guides • Communication structure between business management and development team

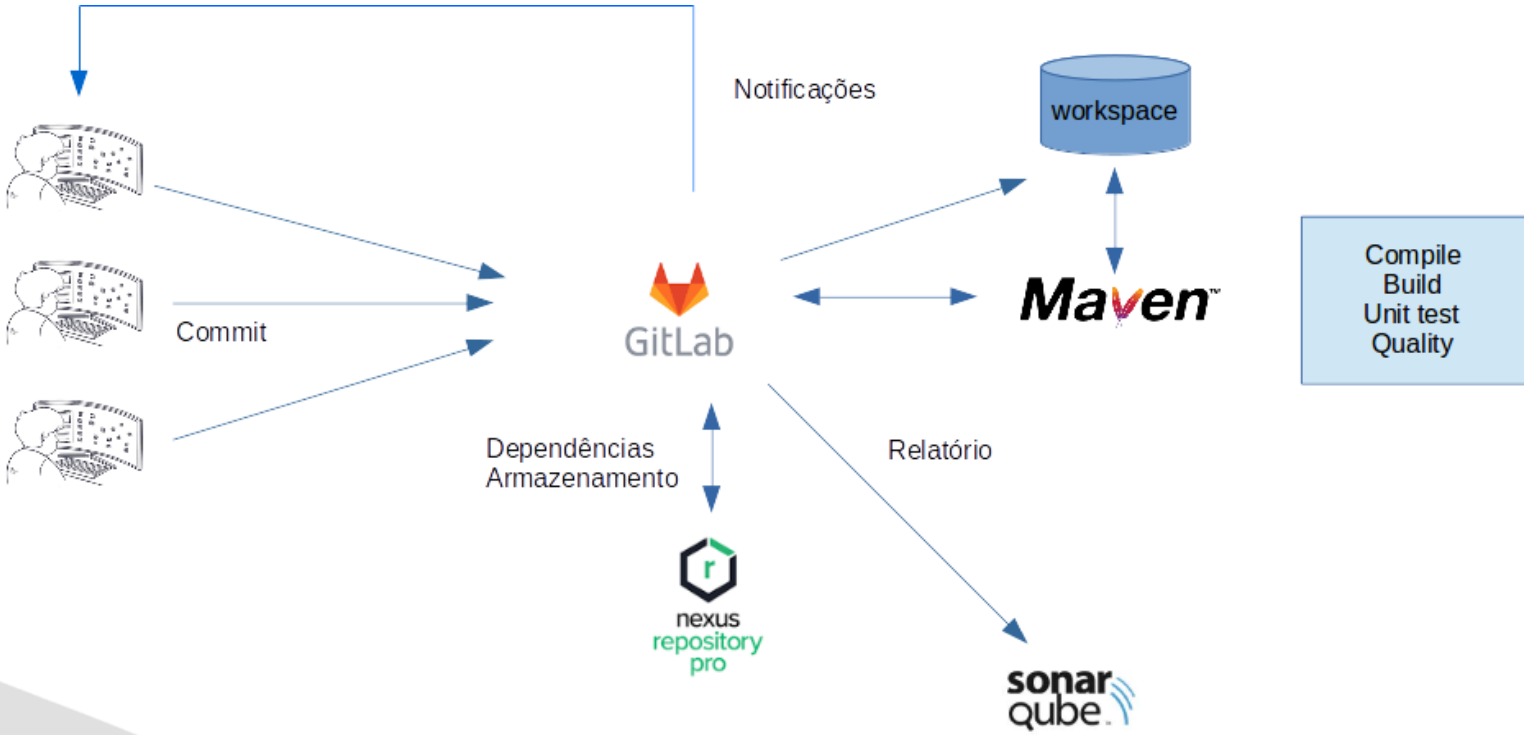
https://ispma.org/the-sources-and-approaches-to-management-of-technical-debt-yli-huumo-maglyas-smolander/

Continuous Integration/Delivery/Deployment



https://stackoverflow.com/questions/28608015/continuous-integration-vs-continuous-delivery-vs-continuous-deployment

Continuous Integration



Delivery e Deployment

Continuous Delivery

Práticas que visam reduzir os riscos de realizar deploy e release em produção. Inclui criar as fundações do pipeline de deployment automatizado.

Continuous Deployment

Quando estão sendo implantadas em produção em intervalos regulares (ao menos uma vez ao dia por desenvolvedor, ou até automaticamente quando alguém faz um commit).

Delivery e Deployment

Deployment

Instalação de uma versão específica do software em um dado ambiente.

Release é a disponibilização de funcionalidades para todos usuários ou apenas um segmento.

O deployment pode ou não estar associado com a liberação de funcionalidades para o cliente.

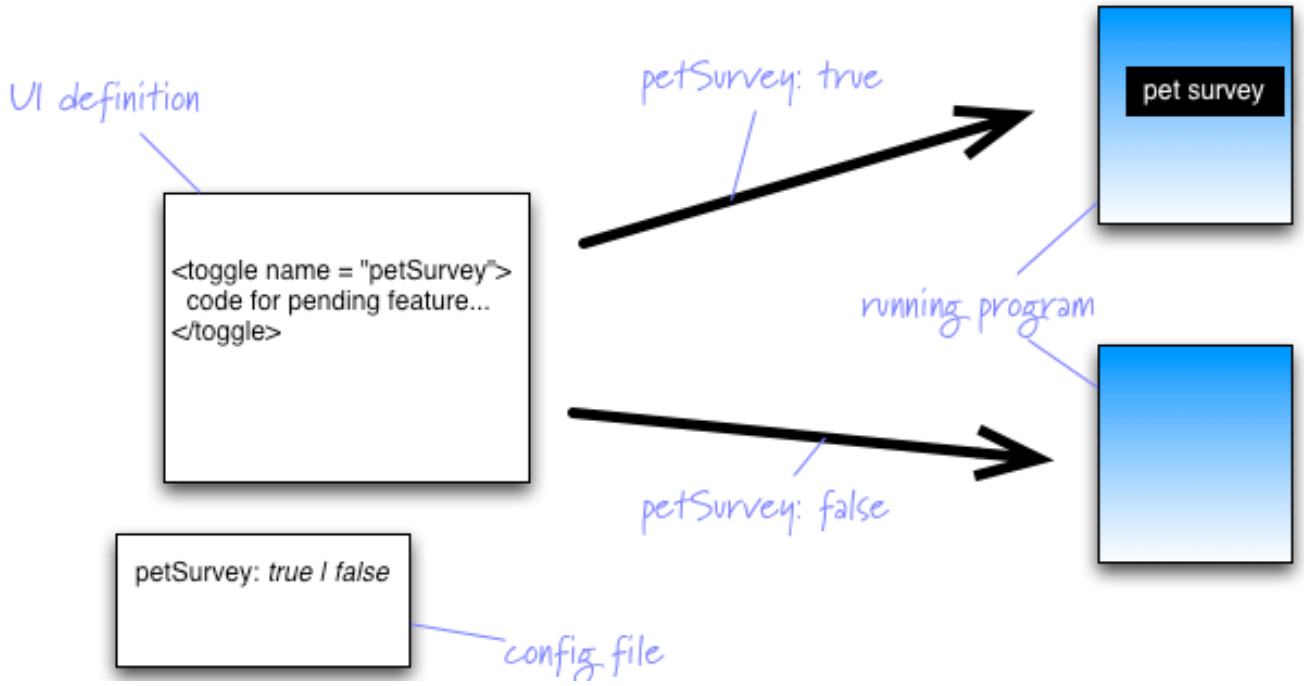
Padrões para Liberação

Feature Toggle.

Blue-green deployment.

Canary releases.

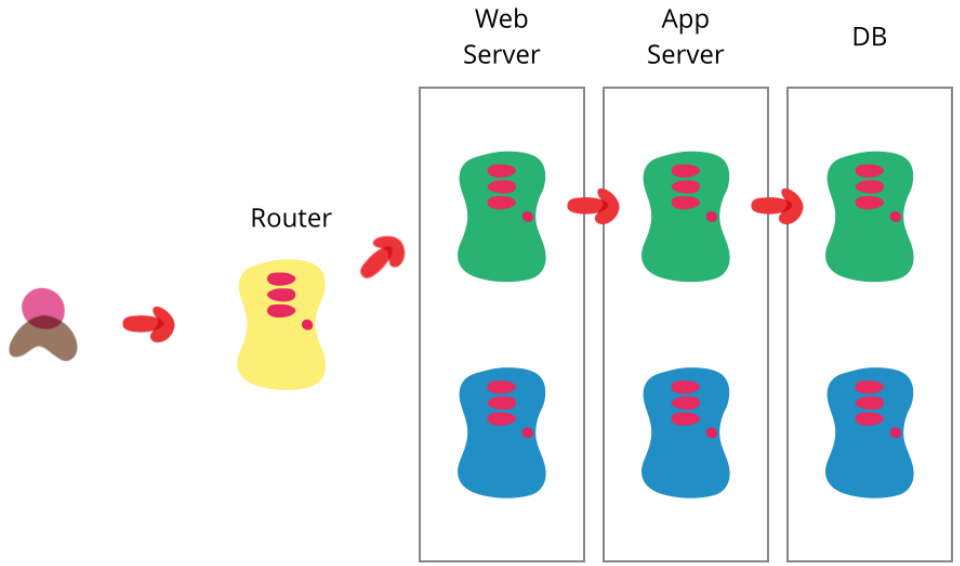
Feature Toggle



https://martinfowler.com/bliki/FeatureToggle.html

Blue-green deployment

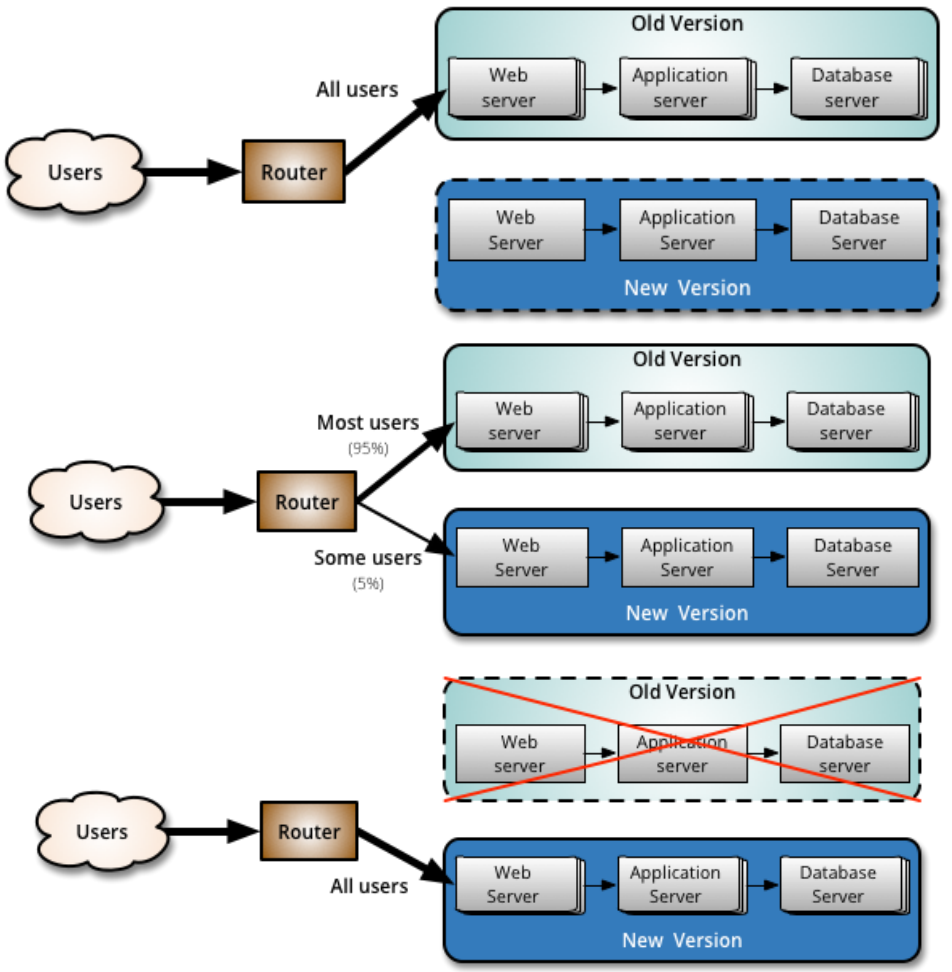
Existência de dois ambientes de produção (blue e green), sendo que apenas um serve ao tráfego de usuários.



https://martinfowler.com/bliki/BlueGreenDeployment.html

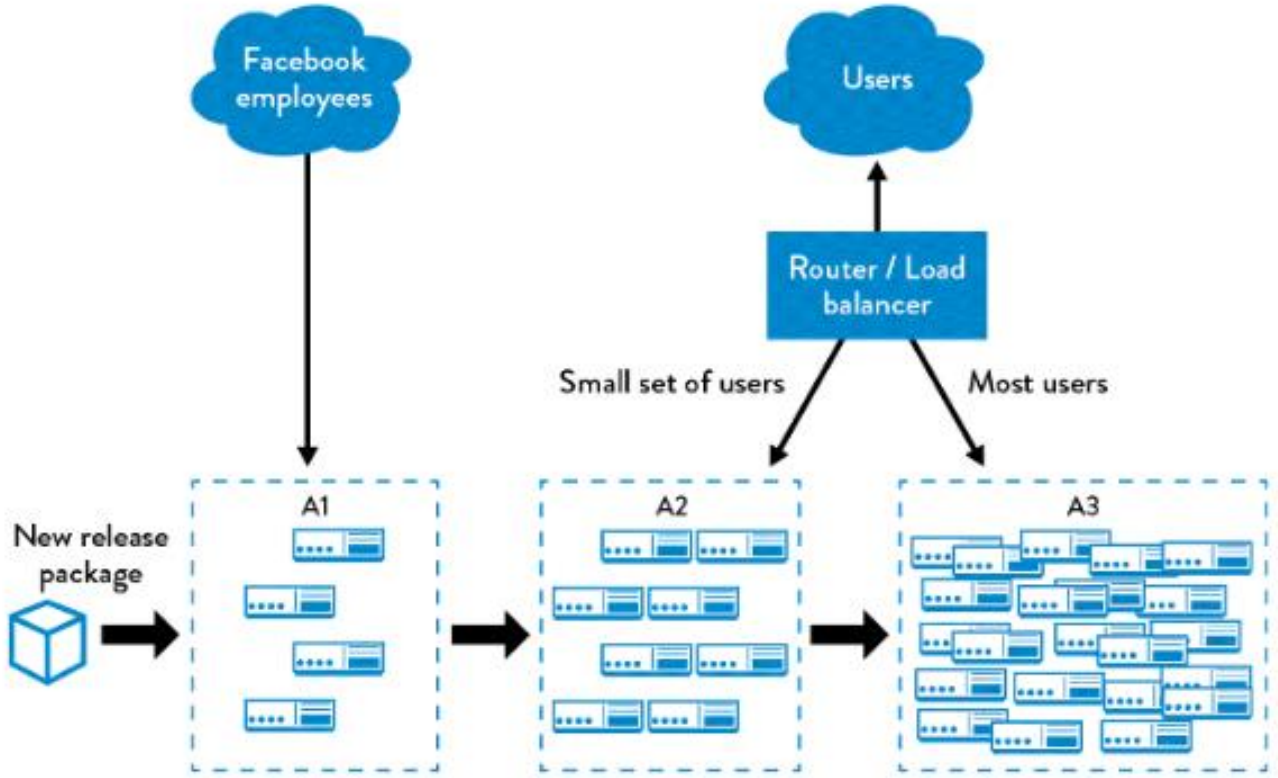
Canary releases

Automatizar o processo de release de forma a promover sucessivamente para audiências e ambientes mais críticos, à medida que vai se confirmando o funcionamento adequado.



https://martinfowler.com/bliki/CanaryRelease.html

Canary releases e cluster immune

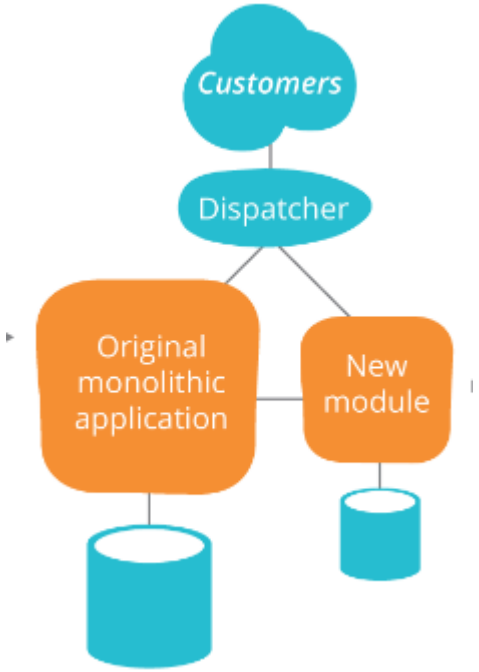


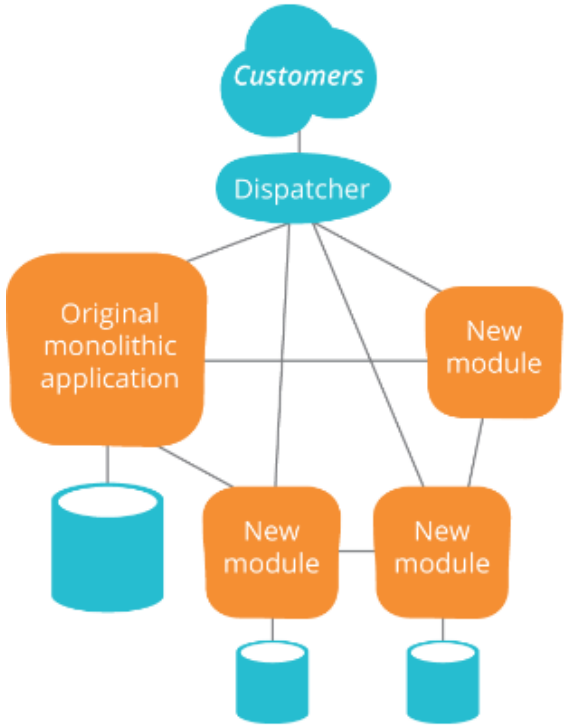
The DevOps Handbook

Arquitetura e código legado

Strangler Pattern.

Strangler Pattern





https://continuousdelivery.com/implementing/architecture/

Lidando Efetivamente com Legado

1. Identificar pontos de mudança.
2. Quebrar dependências.
3. Escrever os testes.
4. Fazer as mudanças.
5. Refatorar.

https://continuousdelivery.com/implementing/architecture/

Resumindo...

Espiral da Morte e débito técnico.

Continuous Integration, Delivery e Deployment.

Arquitetura e código legado.