



Este trabalho está licenciado  
com uma Licença Creative  
Commons - Atribuição 4.0  
Internacional

# Engenharia de Software Contínua: Visão Geral, Desafios e Perspectivas

Monalessa Perini Barcellos

monalessa@inf.ufes.br

<http://www.inf.ufes.br/~monalessa>

Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO)  
Departamento de Informática  
Centro Tecnológico  
Universidade Federal do Espírito Santo



1

## Um pouco sobre mim



Doutorado em Engenharia de Sistemas e Computação

Mestrado em Engenharia de Sistemas e Computação

Graduação em Ciência da Computação

Professora do Departamento de Informática/ PPGI/UFES

**COPPE**  
UFRJ



Membro Sênior do Núcleo de Estudos em Modelagem Conceitual e Ontologias

**nemo**

(Co)Coordenadora do Laboratório de Práticas em Engenharia de Software “Ricardo de Almeida Falbo”

<http://www.inf.ufes.br/~monalessa>

<http://lattes.cnpq.br/8826584877205264>

## NEMO (<https://nemo.inf.ufes.br>)

**Resultados**

ENGENHARIA DE SOFTWARE CONTÍNUA: VISÃO GERAL, DESAFIOS E PERSPECTIVAS

## Engenharia de Software Contínua

**Resultados**

APLICAÇÕES

- Semantic Documentation (HO-ON)
- Human-Computer Interaction Strategy Network (INTEGRA DOCE)
- MeSE - Measurement in Software Engineering (INTERSEPE)
- Ontology-based Source Code Interoperability (ROSE)
- OSDN
- ROSE
- SEON

ERES 2022

3

## LabES (<https://labes.inf.ufes.br>)

Monalessa P. Barcellos    Vítor E. Silva Souza

Camila Z. de Aguiar    Patrícia Dockhorn Costa

ENGENHARIA DE SOFTWARE CONTÍNUA: VISÃO GERAL, DESAFIOS E PERSPECTIVAS

Como mostra a figura, o processo sugere duas atividades para estabelecer um contexto inicial (PM Canvas e User Stories (Histórias)), a partir do qual o processo entra em um ciclo iterativo, no qual é estabelecido um Escopo (MVP), criados os critérios de aceitação, definido o backlog, realizadas as sprints (Sprint Sprints) e realizadas as demonstrações (Protótipo). O processo é finalizado com a apresentação das classes.

ERES 2022

4

2

## Escopo desta palestra

Motivação

Visão Geral

Questões de Pesquisa

Propostas de Solução

Desafios

## Introdução

# Introdução

A sociedade contemporânea e digital transformaram o cenário de desenvolvimento de software.

Mudanças na forma de desenvolver e entregar software.

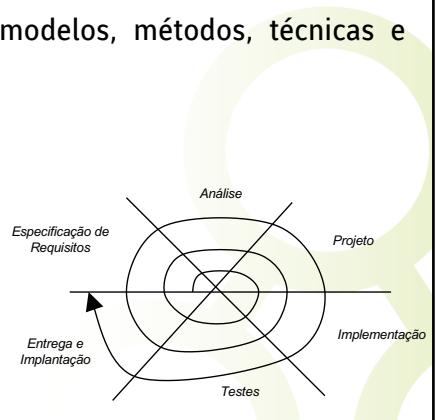
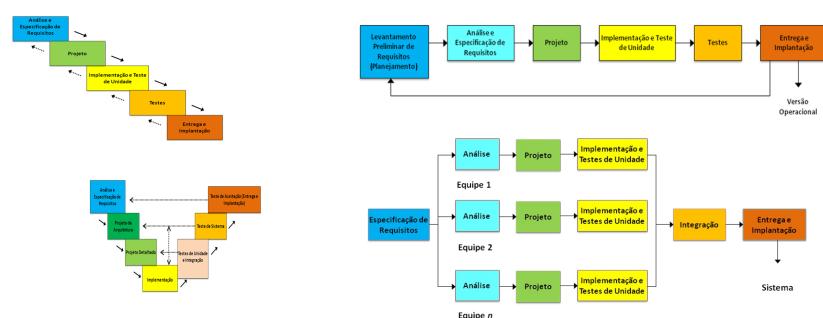
Novos desafios para o desenvolvimento de software:

- ✓ Entregas mais rápidas e frequentes
- ✓ Mudanças frequentes em requisitos (requisitos x hipóteses)
- ✓ Menor tolerância a falhas
- ✓ Adaptação a novos modelos de negócio (SaaS, decisões baseadas em feedback de usuário,...)
- ✓ Softwares mais complexos (capacidade de interoperar, evolução por tempo indeterminado, UX, ...)
- ✓ Software como valor percebido
- ✓ ...

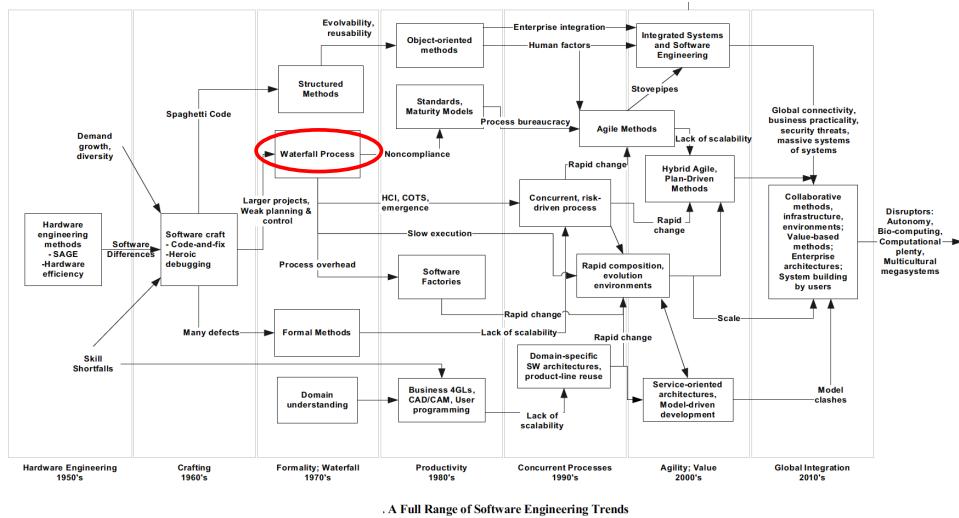
# Introdução

Essas mudanças não aconteceram “da noite para o dia”...

Ao longo dos anos, a Engenharia de Software vem propondo modelos, métodos, técnicas e ferramentas para atender as mudanças nas demandas.



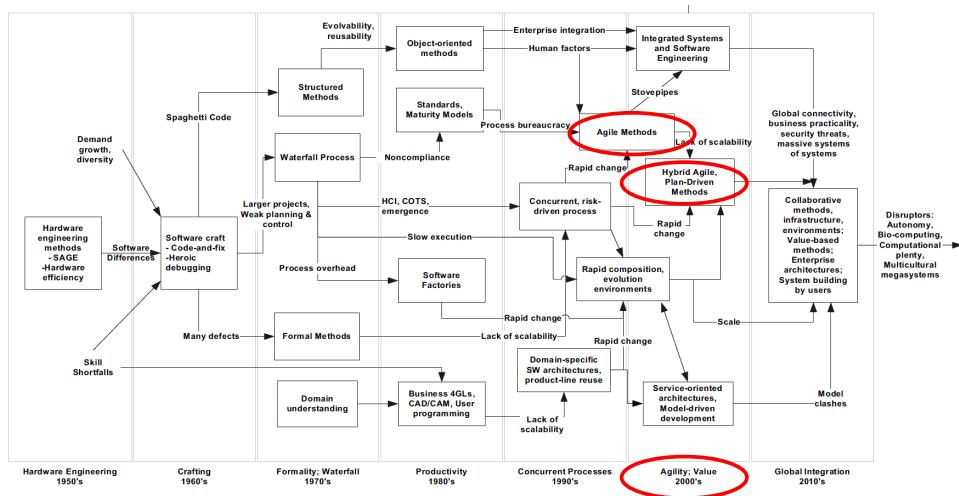
# Introdução



. A Full Range of Software Engineering Trends

Barry Boehm. 2006. A view of 20th and 21st century software engineering. In Proceedings of the 28th international conference on Software engineering (ICSE '06). ACM, New York, NY, USA, 12–29.

# Introdução

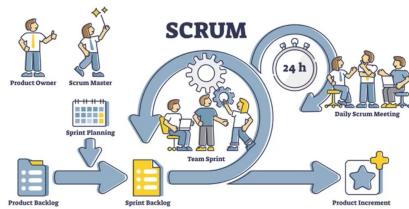
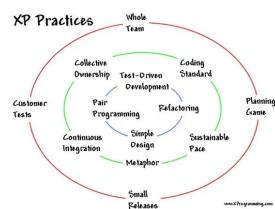


. A Full Range of Software Engineering Trends

Barry Boehm. 2006. A view of 20th and 21st century software engineering. In Proceedings of the 28th international conference on Software engineering (ICSE '06). ACM, New York, NY, USA, 12–29.

# Introdução

Grande mudança no desenvolvimento de software a partir dos **Métodos Ágeis**



## Manifesto Ágil



- ✓ Ciclos curtos de desenvolvimento
- ✓ Entregas frequentes
- ✓ Equipes empoderadas
- ✓ Melhor absorção de mudanças em requisitos
- ✓ Aumento da colaboração com o cliente
- ✓ Entrega de software de valor

# Introdução

A adoção de práticas ágeis evidenciou que:

- ✓ **Aumentar frequência** de algumas atividades críticas no desenvolvimento contribui para reduzir alguns problemas
- ✓ Práticas como "*release early, release often*" são benéficas em termos de qualidade e consistência
- ✓ Desenvolvimento de software requer **flexibilidade e adaptação**
- ✓ **Fluidez** entre as atividades do processo ajuda a acelerar o desenvolvimento (e.g., integração contínua)
- ✓ **Descontinuidade** entre desenvolvimento e entrega gera problemas (necessidade de DevOps)
- ✓ **Dificuldade no uso de dados** para guiar o desenvolvimento de software e a tomada de decisão leva a decisões baseadas em intuição e conhecimento tácito

# Introdução

É preciso ir além das práticas ágeis para atender as demandas atuais:

- ✓ **Diminuir descontinuidades** entre negócio, desenvolvimento e operação
- ✓ **Visão holística** do processo de desenvolvimento
- ✓ **Fluxo contínuo** entre atividades do ciclo de vida do software
- ✓ **Alinhamento contínuo** ao negócio
- ✓ Realizar ações **continuamente** (e.g., planejamento, construção, operação, implantação e avaliação)
- ✓ Desenvolvimento de software **orientado a dados** (decisões diárias, melhorias, oportunidades)

# Introdução

É preciso ir além das práticas ágeis para atender as demandas atuais:

- ✓ **Diminuir descontinuidades** entre negócio, desenvolvimento e operação
- ✓ **Visão holística** do processo de desenvolvimento
- ✓ **Fluxo contínuo** entre atividades do ciclo de vida do software
- ✓ **Alinhamento contínuo** ao negócio
- ✓ Realizar ações **continuamente** (e.g., planejamento, construção, operação, implantação e avaliação)
- ✓ Desenvolvimento de software **orientado a dados** (decisões diárias, melhorias, oportunidades)

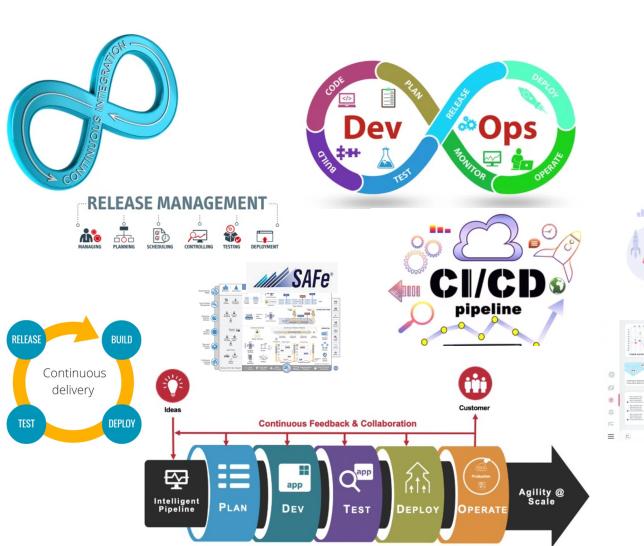
## Engenharia de Software Contínua

# O que é Engenharia de Software Contínua?

## Uma Visão Geral

15

## Engenharia de Software Contínua



Experiment-driven Development



Continuous Evolution



... e mais

16

# Engenharia de Software Contínua

Engenharia de Software Contínua (ESC) consiste em um conjunto de práticas e ferramentas que apoiam uma **visão holística** do desenvolvimento de software com o objetivo de torná-lo **mais rápido, iterativo, integrado, contínuo e alinhado ao negócio**.

Seu objetivo é estabelecer um **fluxo contínuo** entre as atividades relacionadas ao software, levando em consideração **todo o ciclo de vida do software**.

Busca transformar práticas de desenvolvimento discretas em alternativas mais **iterativas, flexíveis e contínuas**, preservando o objetivo de construir e entregar software de qualidade, e que atenda requisitos de tempo e custos.

Brian Fitzgerald and Klaas-Jan Stol. 2017. Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software* 123: 176–189.

ENGENHARIA DE SOFTWARE CONTÍNUA: VISÃO GERAL, DESAFIOS E PERSPECTIVAS



ERES 2022

17

# Engenharia de Software Contínua

Em Engenharia de Software Contínua:

- ✓ **Clientes** são **proativos** e os **usuários** e outros interessados estão **envolvidos** no processo de desenvolvimento, aprendendo com **dados de uso e feedback**;
- ✓ **Planejamento** é **contínuo**, assim como a **engenharia de requisitos**, que se concentra em funcionalidades, arquitetura e projeto modularizados e rápida realização de mudanças;
- ✓ São empregadas **práticas ágeis**, incluindo ciclos curtos de desenvolvimento, **integração contínua** e **entrega contínua** e automatizada de releases;

Jan Ole Johansson, Anja Kleebaum, Barbara Paech, and Bernd Bruegge. 2019. Continuous software engineering and its support by usage and decision knowledge: An interview study with practitioners. *Journal of Software: Evolution and Process* 31, 5.

ENGENHARIA DE SOFTWARE CONTÍNUA: VISÃO GERAL, DESAFIOS E PERSPECTIVAS



ERES 2022

18

# Engenharia de Software Contínua

Em Engenharia de Software Contínua:

- ✓ Há **controle de versão** de código, estratégias de *branching*, *commits* rápidos de código, cobertura de código e revisões;
- ✓ A **garantia de qualidade** envolve testes automatizados, *builds* regulares, *pull requests*, auditorias e adaptação em tempo de execução;
- ✓ O **conhecimento** é compartilhado e o **aprendizado contínuo** acontece, capturando decisões e fundamentos.

Jan Ole Johannsen, Anja Kleebaum, Barbara Paech, and Bernd Bruegge. 2019. Continuous software engineering and its support by usage and decision knowledge: An interview study with practitioners. *Journal of Software: Evolution and Process* 31, 5.

# Engenharia de Software Contínua

Algumas abordagens/práticas relacionadas a ESC:

## Lean Thinking



Visa diminuir o tempo entre a solicitação do cliente e a entrega do produto/funcionalidade, removendo atividades que não acrescentam valor.

**Movimento contínuo**, fluxo entre atividades/pessoas, automatização, melhoria contínua, alinhamento ao negócio.

Ágil x Lean: desenvolvimento de software ágil foca na função de desenvolvimento, enquanto Lean provê visão holística do processo mais amplo (processo ponta a ponta: do cliente à entrega), que envolve outras funções da organização.

(Fitzgerald and Stol, 2017)

# Engenharia de Software Contínua



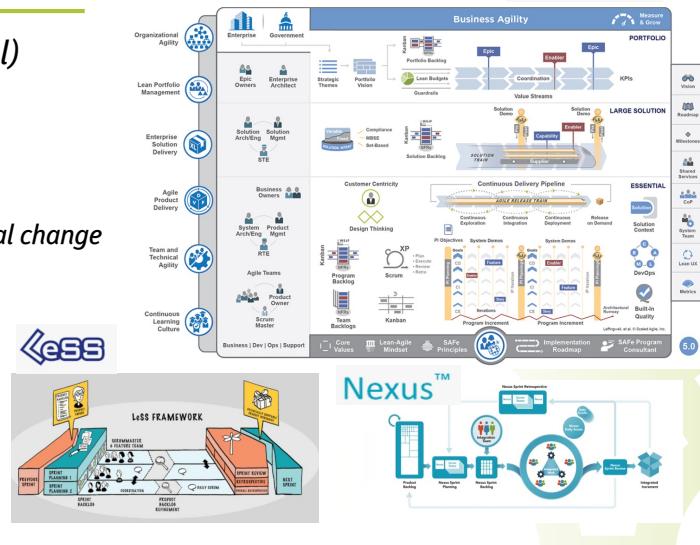
## Ágil Escalado (Agilidade Organizacional)

Agilidade além da função de desenvolvimento

*"The ability of organizations to sense environmental change and respond appropriately"* (Overby et al., 2005)

Duas capacidades:

- ✓ *Sense* (análise de dados de feedback de usuários)
- ✓ *Response* (rápida adaptação e entrega de novas funcionalidades)



Overby, E. , Bharadwaj, A. , Sambamurthy, V. , 2005. A framework for enterprise agility and the enabling role of digital options, business agility and information technology diffusion. *Business Agility and Information Technology Diffusion*, IFIP, vol. 180. Springer, pp. 295–312.

# Engenharia de Software Contínua

## DevOps

Provê continuidade entre as atividades do desenvolvimento de software a sua implantação em ambiente de produção.

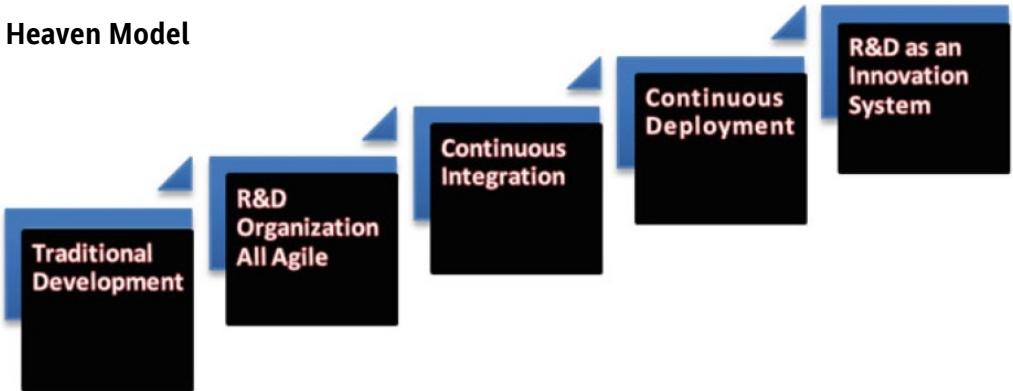


Humble, J. , Molesky, J. , 2011. Why enterprises must adopt devops to enable continuous delivery. *Cutter IT J.* 24 (8), 6–12

# Engenharia de Software Contínua

Algumas propostas que proveem uma visão geral da ESC:

## Stairway to Heaven Model



Helena H. Olsson, Hiva Alahyari, and Jan Bosch. 2012. Climbing the "Stairway to Heaven" - A Multiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software. In 2012 38th Euromicro Conference on Software Engineering and Advanced Applications, 392–399.

# Engenharia de Software Contínua

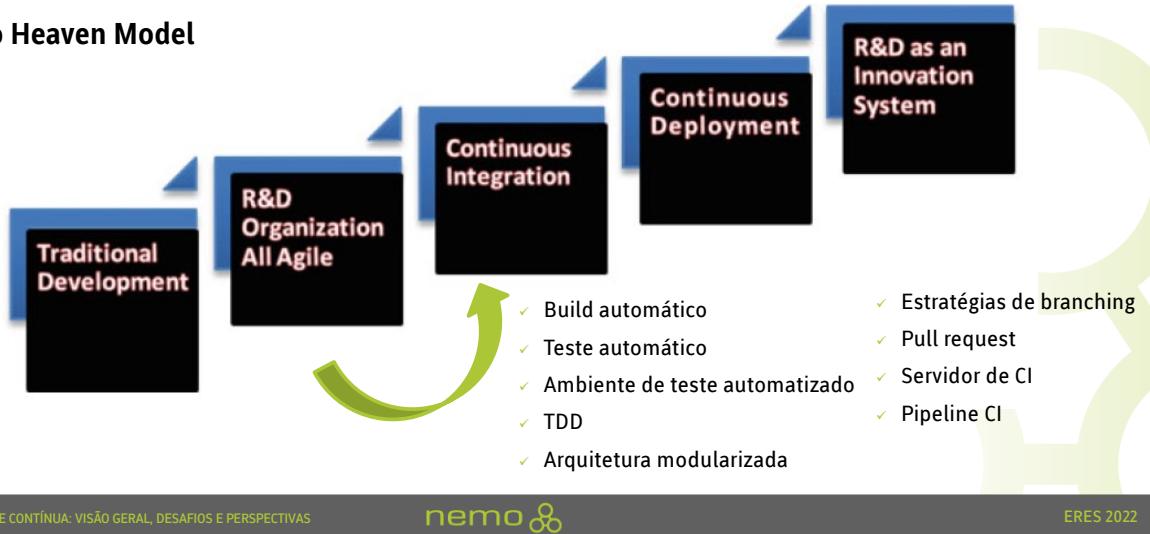
## Stairway to Heaven Model



- ✓ Equipes pequenas e empoderadas
- ✓ Ciclos curtos com entregas frequentes
- ✓ Backlog
- ✓ Eventos/cerimônias para gerenciamento
- ✓ Foco em funcionalidades ao invés de componentes

# Engenharia de Software Contínua

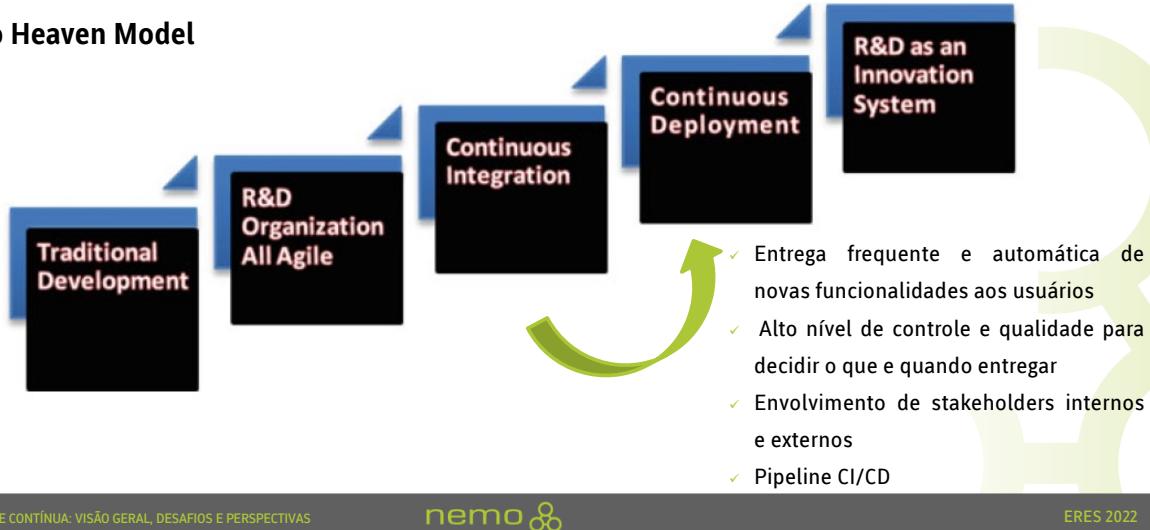
## Stairway to Heaven Model



25

# Engenharia de Software Contínua

## Stairway to Heaven Model



26

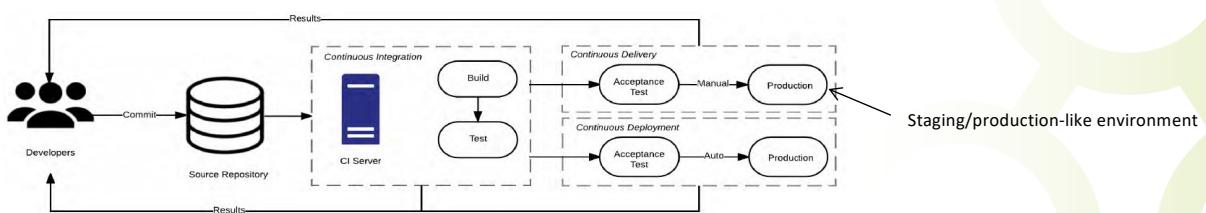
13

# Engenharia de Software Contínua

Há discussão sobre esses conceitos  
(não há consenso).

## Continuous Delivery (CDE) x Continuous Deployment (CD)

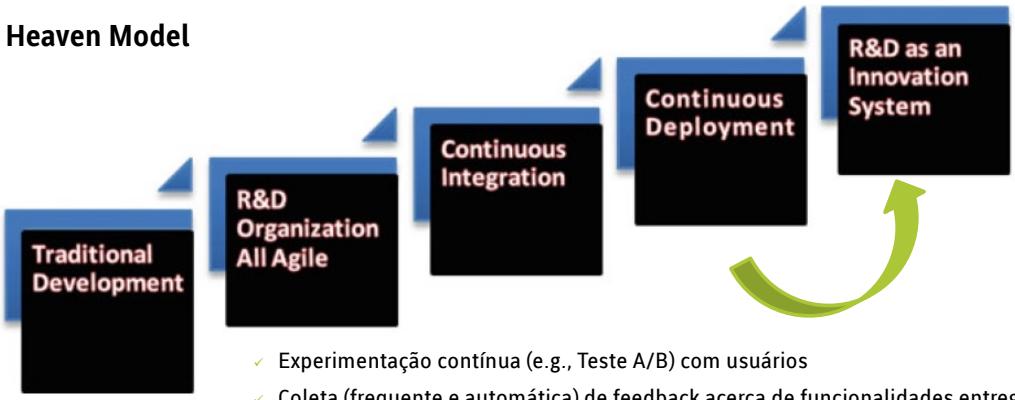
- ✓ Diferem na forma – *push-based* x *pull-based* (CDE envolve alguma intervenção ‘manual’ - e.g., para aprovar o que será entregue; CD disponibiliza as alterações automaticamente)
- ✓ CDE faz sentido em todas as aplicações, CD não (CI → CDE → CD)
- ✓ Diferem no ambiente onde a entrega é feita (CD entrega em ambiente de produção, enquanto CDE pode entregar em um ambiente intermediário)



M. Shahin, M. Ali Babar and L. Zhu, Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices, in *IEEE Access*, vol. 5, pp. 3909-3943, 2017.

# Engenharia de Software Contínua

## Stairway to Heaven Model

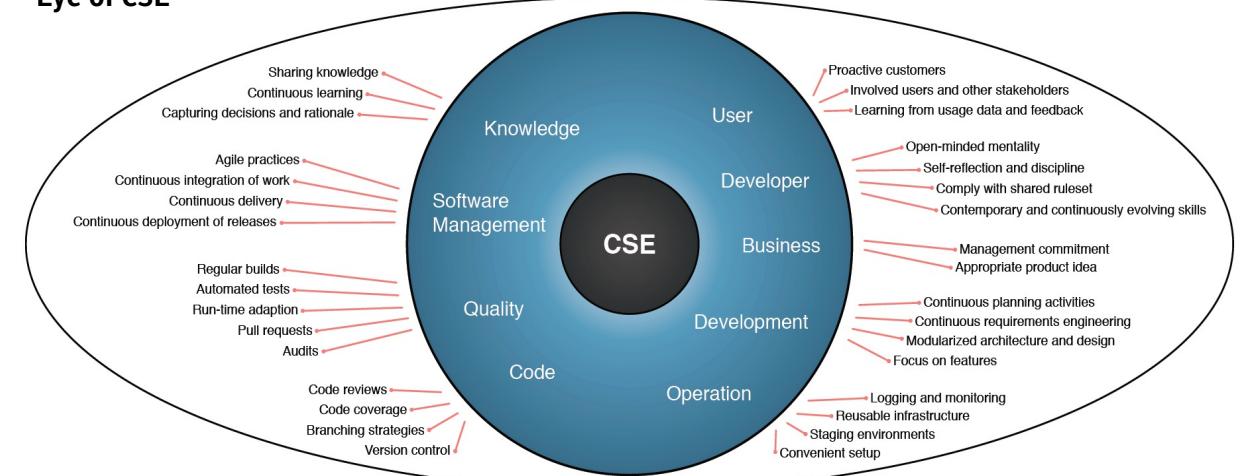


- ✓ Experimentação contínua (e.g., Teste A/B) com usuários
- ✓ Coleta (frequente e automática) de feedback acerca de funcionalidades entregues
- ✓ Uso de dados para identificar melhorias, novas features e oportunidades de negócio
- ✓ Alinhamento com o ecossistema do negócio

# Engenharia de Software Contínua

## Eye of CSE

33 práticas agrupadas em 9 categorias



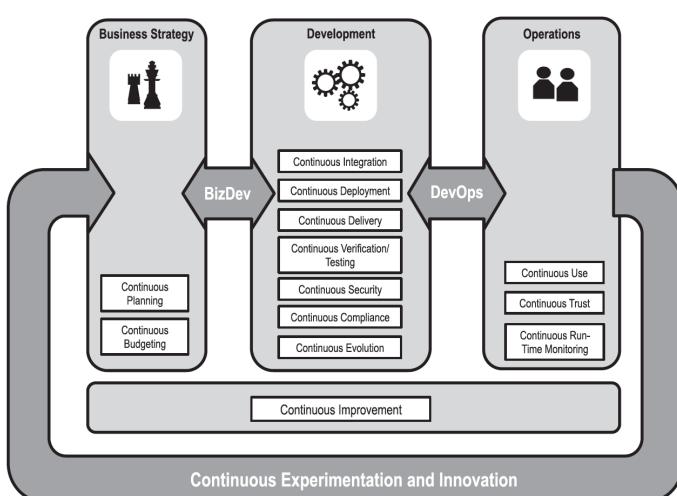
Jan O. Johanssen, Anja Kleebaum, Barbara Paech, and Bernd Bruegge. 2018. Practitioners' Eye on Continuous Software Engineering: An Interview Study. In *Proceedings of the International Conference on Software and System Process*, 41–50.

# Engenharia de Software Contínua

## Continuous\*

16 atividades agrupadas em 4 categorias:

Business Strategy and Planning  
Development  
Operations  
Improvement and Innovation

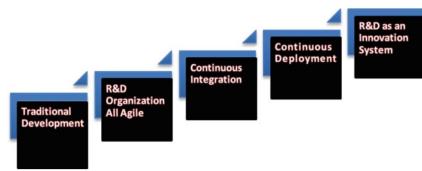


Brian Fitzgerald and Klaas-Jan Stol. 2017. Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software* 123: 176–189.

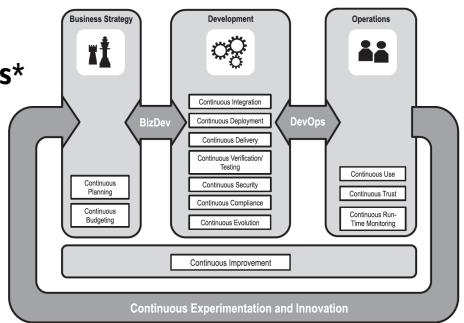
# Engenharia de Software Contínua

## Stairway to Heaven Model

Que práticas estão contidas em cada estágio? E outros processos (KM, QA)?



## Continuous\*



Como as atividades/processos relacionam-se uns com os outros?

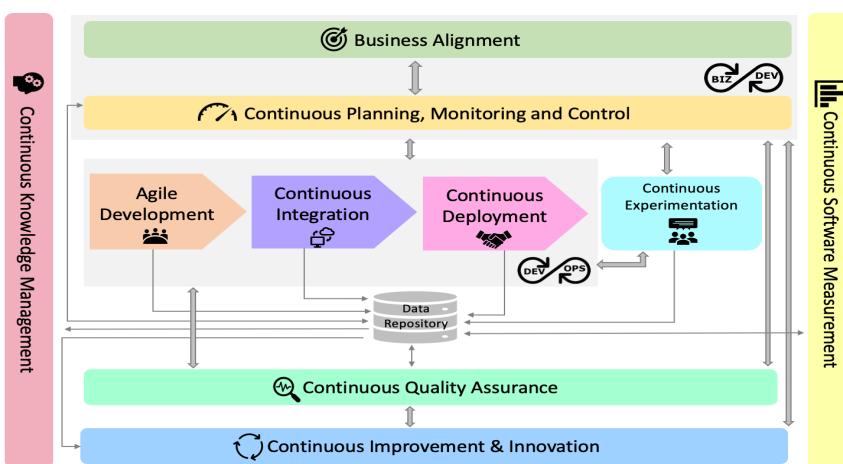
## Eye of CSE

A quais processos as práticas se relacionam?



# Engenharia de Software Contínua

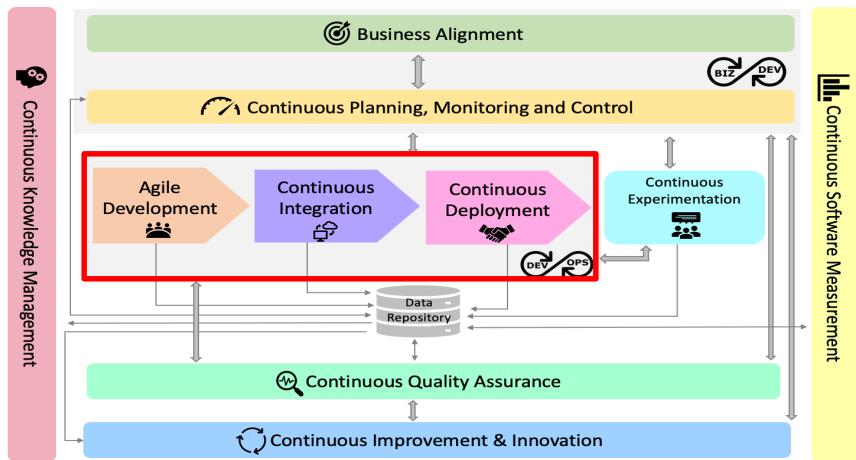
## Continuous Software Engineering Framework



10 processos da ESC e suas relações (fluxos de dados e informações)

# Engenharia de Software Contínua

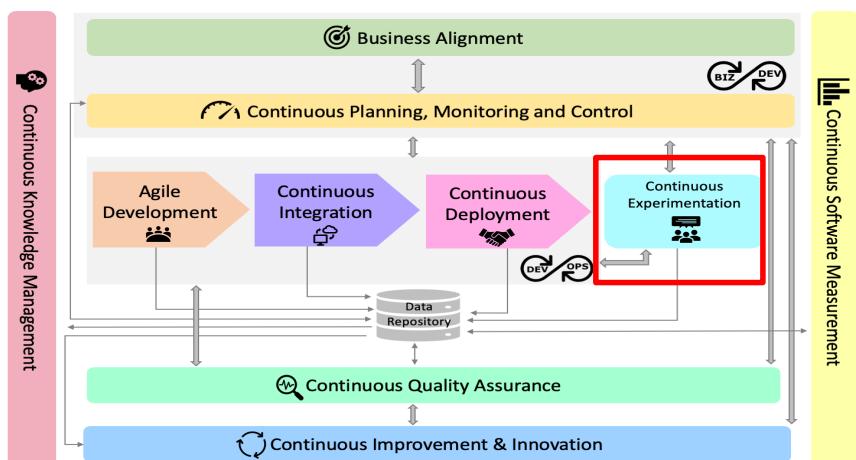
## Continuous Software Engineering Framework



M. P. Barcellos, Towards a Framework for Continuous Software Engineering, in 34th Brazilian Symposium on Software Engineering (SBES 2020), 2020, p. 626–631.

# Engenharia de Software Contínua

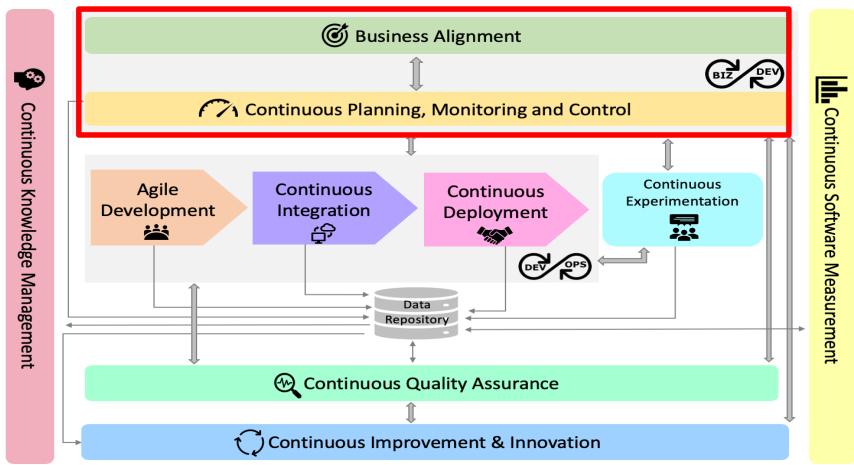
## Continuous Software Engineering Framework



M. P. Barcellos, Towards a Framework for Continuous Software Engineering, in 34th Brazilian Symposium on Software Engineering (SBES 2020), 2020, p. 626–631.

# Engenharia de Software Contínua

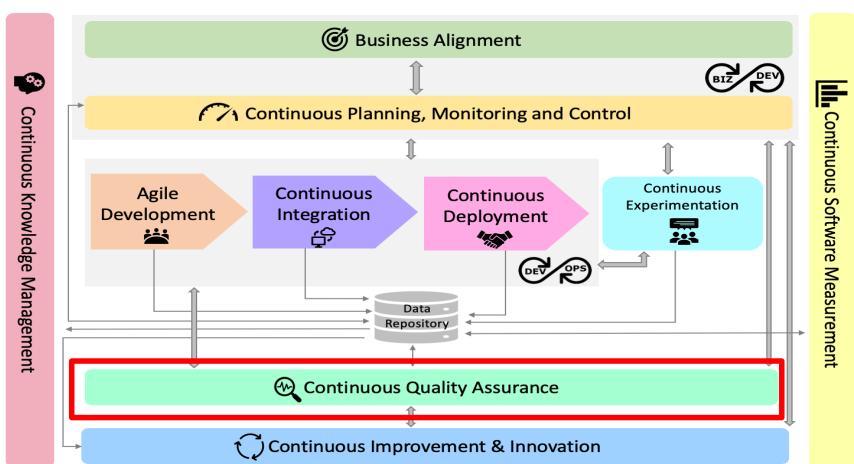
## Continuous Software Engineering Framework



M. P. Barcellos, Towards a Framework for Continuous Software Engineering, in 34th Brazilian Symposium on Software Engineering (SBES 2020), 2020, p. 626–631.

# Engenharia de Software Contínua

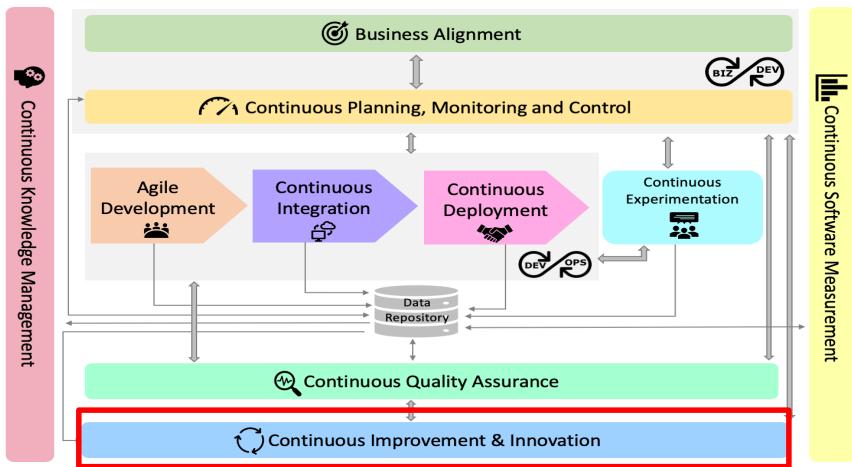
## Continuous Software Engineering Framework



M. P. Barcellos, Towards a Framework for Continuous Software Engineering, in 34th Brazilian Symposium on Software Engineering (SBES 2020), 2020, p. 626–631.

# Engenharia de Software Contínua

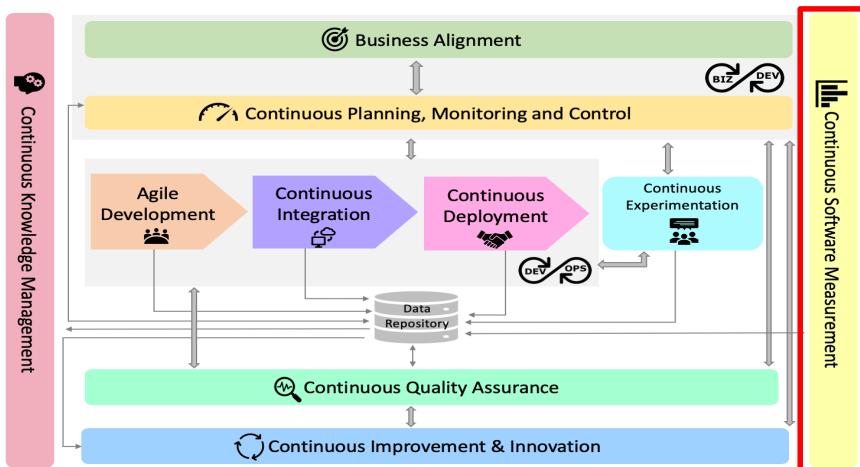
## Continuous Software Engineering Framework



M. P. Barcellos, Towards a Framework for Continuous Software Engineering, in 34th Brazilian Symposium on Software Engineering (SBES 2020), 2020, p. 626–631.

# Engenharia de Software Contínua

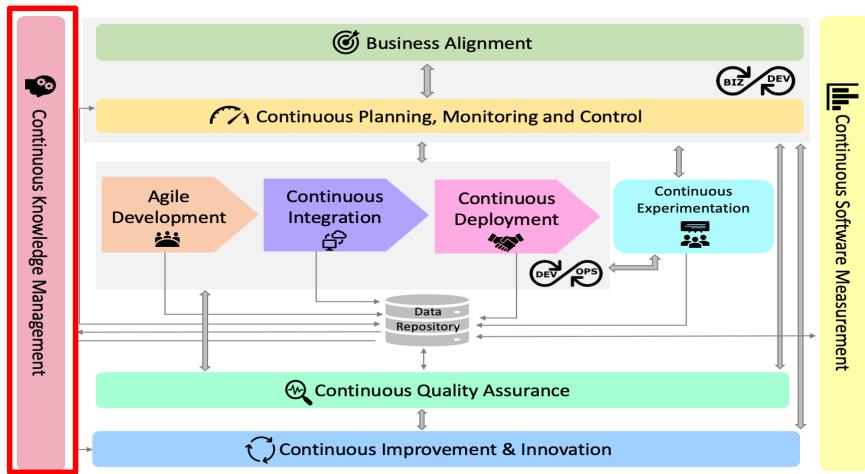
## Continuous Software Engineering Framework



M. P. Barcellos, Towards a Framework for Continuous Software Engineering, in 34th Brazilian Symposium on Software Engineering (SBES 2020), 2020, p. 626–631.

# Engenharia de Software Contínua

## Continuous Software Engineering Framework



M. P. Barcellos, Towards a Framework for Continuous Software Engineering, in 34th Brazilian Symposium on Software Engineering (SBES 2020), 2020, p. 626–631.

ENGENHARIA DE SOFTWARE CONTÍNUA: VISÃO GERAL, DESAFIOS E PERSPECTIVAS

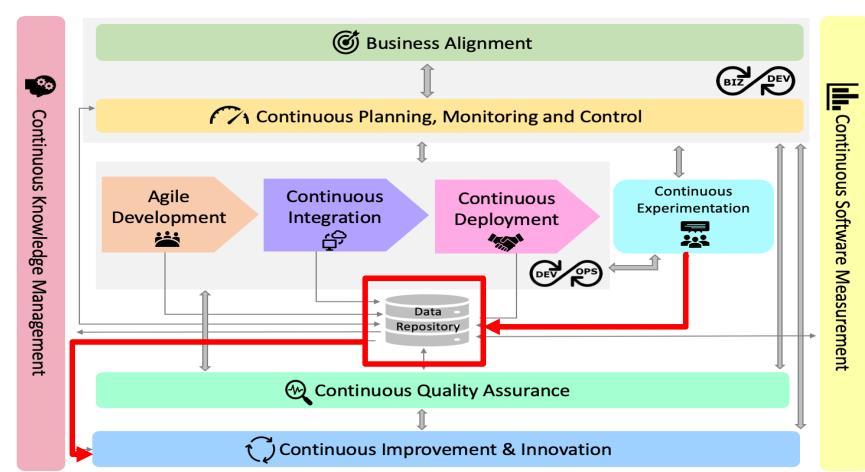


ERES 2022

39

# Engenharia de Software Contínua

## Continuous Software Engineering Framework



M. P. Barcellos, Towards a Framework for Continuous Software Engineering, in 34th Brazilian Symposium on Software Engineering (SBES 2020), 2020, p. 626–631.

ENGENHARIA DE SOFTWARE CONTÍNUA: VISÃO GERAL, DESAFIOS E PERSPECTIVAS



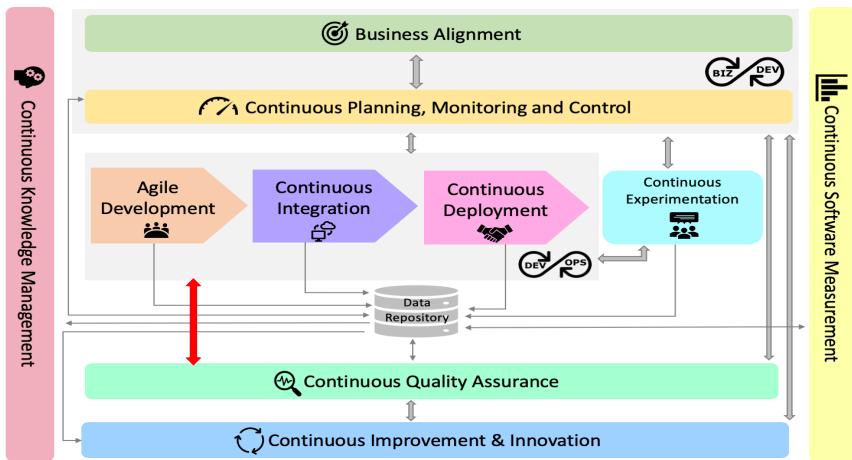
ERES 2022

40

20

# Engenharia de Software Contínua

## Continuous Software Engineering Framework



M. P. Barcellos, Towards a Framework for Continuous Software Engineering, in 34th Brazilian Symposium on Software Engineering (SBES 2020), 2020, p. 626–631.

ENGENHARIA DE SOFTWARE CONTÍNUA: VISÃO GERAL, DESAFIOS E PERSPECTIVAS



ERES 2022

41

## Questões de Pesquisa

## Propostas de Solução

42

21

# Questões de Pesquisa e Algumas Propostas de Solução

**QP1. Quais os processos/atividades, recursos, artefatos e stakeholders envolvidos em Engenharia de Software Contínua? Por exemplo, o que é Continuous Delivery? Como processos contínuos que possuem equivalentes "tradicionais" (e.g., Garantia da Qualidade) diferem destes?**



É preciso aumentar o corpo de conhecimento sobre o que é ESC, suas práticas, processos, etc.



Terminologias



Guias / Padrões

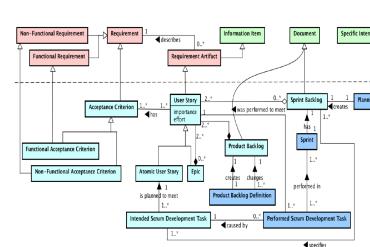


Modelos

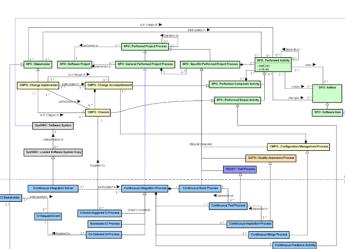
# Questões de Pesquisa e Algumas Propostas de Solução

**Ontologias** podem ser usadas para representar e estruturar conhecimento acerca dos processos.

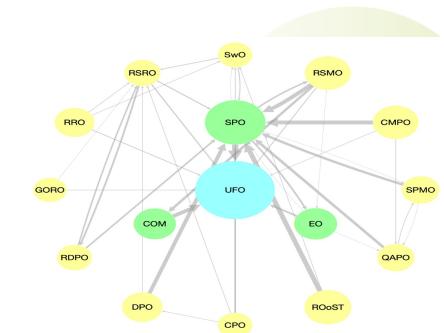
## Ontologias de Tarefa - Ontologias de Domínio - Redes de Ontologias



SRO



CIRO



SEON  
<https://nemo.inf.ufes.br/projetos/seon/>

# Questões de Pesquisa e Algumas Propostas de Solução

**QP2. Como identificar as práticas de ESC a ser implementadas e como evoluir/avançar na implementação das práticas?**

É necessário considerar os diferentes contextos organizacionais.

Entender as necessidades de cada organização.

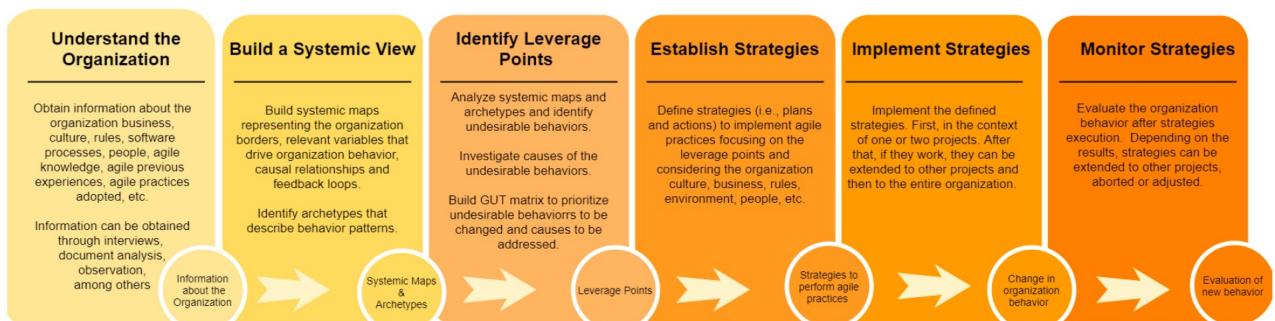
*One size does not fit all!*

Identificar o que ela já faz, o que funciona, o que não funciona, o que falta fazer, o que pode melhorar.



# Questões de Pesquisa e Algumas Propostas de Solução

## California



P. S. dos Santos Jr, M. P. Barcellos, R. F. Calhau, Am I Going to Heaven? First Step Climbing the Stairway to Heaven Model – Results from a Case Study in Industry, in 34th Brazilian Symposium on Software Engineering (SBES 2020), 2020, p. 309–318.

# Questões de Pesquisa e Algumas Propostas de Solução

## Zeppelin

Instrumento diagnóstico para apoiar organizações a obterem uma visão panorâmica das práticas de ESC que adotam e ajudá-las a identificar ações de melhoria para implementar ESC.

The screenshot shows a survey titled "Continuous Integration" with 76 statements. The columns include "Statement", "Adoption Level", and "Comments". The adoption levels are categorized as Not Adopted (green), Partially Adopted (yellow), and Fully Adopted (blue). Many statements mention code coverage, automated testing, and continuous integration environments.

Questionário (76 práticas organizadas em 4 estágios do StH)



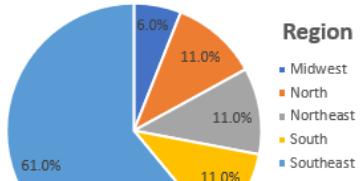
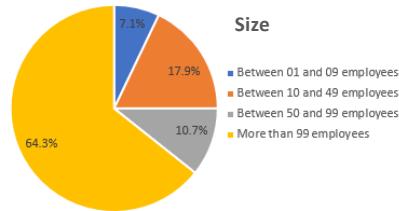
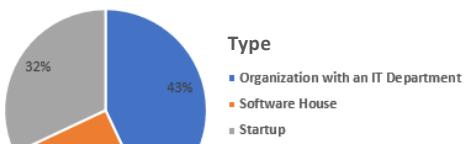
Relatório Analítico

P. S. SantosJr, M. P. Barcellos, F. Ruy, Tell me: Am I going to Heaven? A Diagnosis Instrument of Continuous Software Engineering Practices Adoption, in 25th International Conference on Evaluation and Assessment in Software Engineering (EASE 2021), 2021, pp. 30-39.

# Questões de Pesquisa e Algumas Propostas de Solução

## ESC no Brasil

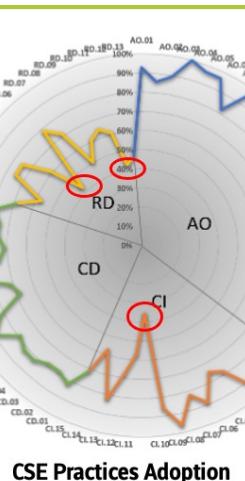
Estudo com 28 organizações – abril/março 2022



P. S. S. Júnior, M. P. Barcellos, F. B. Ruy, M. S. Omêna, Flying over Brazilian Organizations with Zeppelin: A Preliminary Panoramic Picture of Continuous Software Engineering, in Proceedings of the 36th Brazilian Symposium on Software Engineering (SBES 2022), 2022, p. 279-288.



# Questões de Pesquisa e Algumas Propostas de Solução



## Some Practices:

- CI.11 (~40%): Data is collected for metrics (e.g., number of bugs) that allow evaluating the continuous integration process;
- RD.05 (~40%): Experiments (e.g., A/B tests) are performed with customers/consumers to improve products.
- RD.13 (~40%): Knowledge related to continuous experimentation (e.g., internal lectures, tutorials, and knowledge repositories) is shared.

P. S. S. Júnior, M. P. Barcellos, F. B. Ruy, M. S. Omêna, Flying over Brazilian Organizations with Zeppelin: A Preliminary Panoramic Picture of Continuous Software Engineering, in Proceedings of the 36th Brazilian Symposium on Software Engineering (SBES 2022), 2022, p. 279–288.

# Questões de Pesquisa e Algumas Propostas de Solução

(RQ2) How have different type of organizations adopted CSE?

- Organization with IT Department:
  - Agile Organization: 48.8%
  - Continuous Integration: 44.6 %
- Software House:
  - Agile Organization: 65.1%
  - Continuous Deployment: 55.1%
- Startup:
  - Agile Organization: 50.2%
  - Continuous Deployment: 50.1%

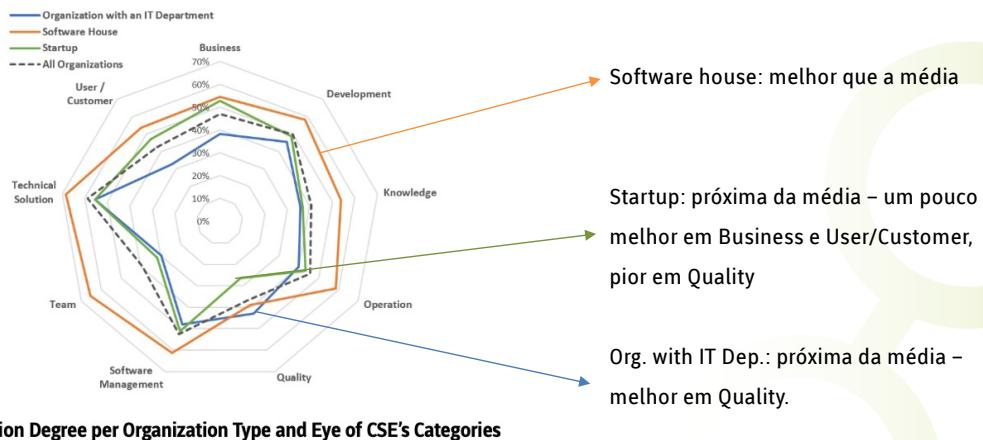
{ Desenvolvimento de software para a própria organização - pode ser mais importante integrar automaticamente o software do implantá-lo automaticamente.

{ Entrega de software para clientes externos (novas versões disponibilizadas na medida que novas funcionalidades ficam prontas)

P. S. S. Júnior, M. P. Barcellos, F. B. Ruy, M. S. Omêna, Flying over Brazilian Organizations with Zeppelin: A Preliminary Panoramic Picture of Continuous Software Engineering, in Proceedings of the 36th Brazilian Symposium on Software Engineering (SBES 2022), 2022, p. 279–288.

# Questões de Pesquisa e Algumas Propostas de Solução

(RQ2) How have different type of organizations adopted CSE?

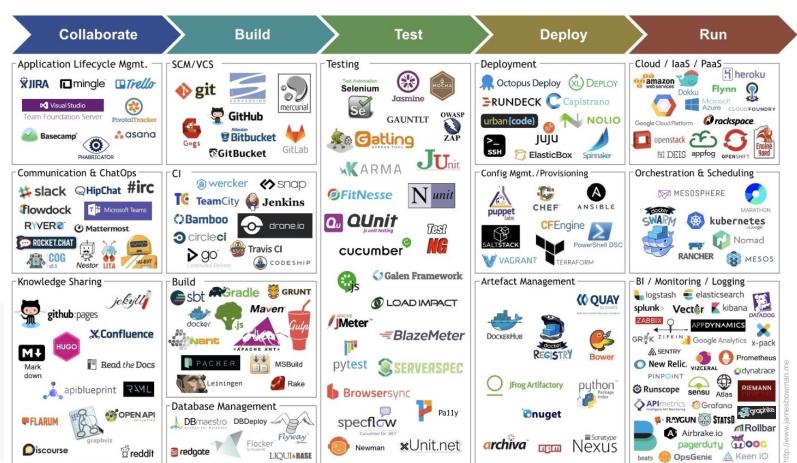


P. S. S. Júnior, M. P. Barcellos, F. B. Ruy, M. S. Omêna, Flying over Brazilian Organizations with Zeppelin: A Preliminary Panoramic Picture of Continuous Software Engineering, in Proceedings of the 36th Brazilian Symposium on Software Engineering (SBES 2022), 2022, p. 279–288.

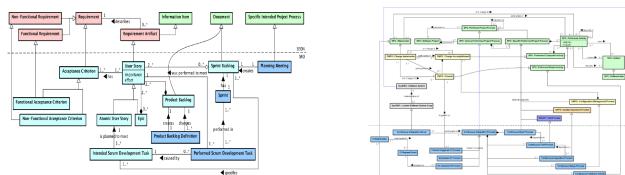
# Questões de Pesquisa e Algumas Propostas de Solução

QP3. Que ferramentas usar para apoiar as práticas de ESC?

ESC é fortemente baseada em ferramentas (depende delas!)



# Questões de Pesquisa e Algumas Propostas de Solução



Ontologias (informam processos, atividades e artefatos de ESC)



Ferramentas



Critérios

(organizacionais, tecnológicos, etc.)

Continuous Software Engineering Tools Catalogue



Guidelines to set your CSE Suite

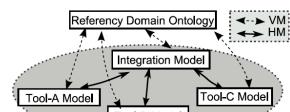


# Questões de Pesquisa e Algumas Propostas de Solução

## QP4. Como integrar dados de diversas ferramentas visando ao desenvolvimento de software e à tomada de decisão orientados a dados?

Mineração de repositórios de software

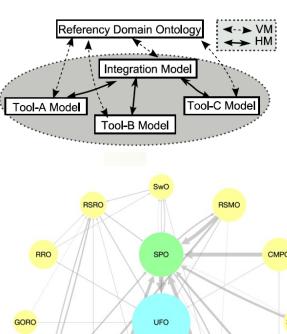
Soluções de interoperabilidade semântica



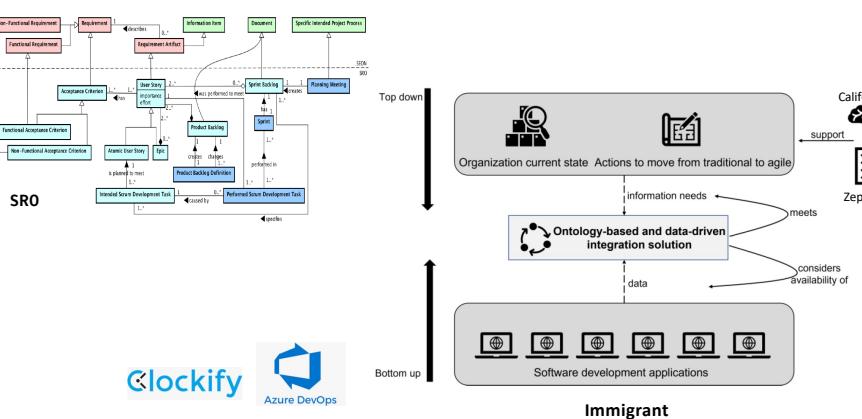
## QP5. Como definir as métricas apropriadas?

Necessidades de informação

Dados disponíveis



# Questões de Pesquisa e Algumas Propostas de Solução



P. S. SantosJr, M. P. Barcellos, R. A. de Falbo, J. P. A. Almeida, From a Scrum Reference Ontology to the Integration of Applications for Data-Driven Software Development, Information and Software Technology, vol. 136, 2021.

P. S. SantosJr, M. P. Barcellos, J. P. A. Almeida, An Ontology-based Approach to enable Data-Driven Decision-Making in Agile Software Organizations, in 5th Doctoral and Masters Consortium on Ontologies – 14th Seminar on Ontology Research in Brazil (ONTOBRAZ), 2021.

# Questões de Pesquisa e Algumas Propostas de Solução

**QP6. Como gerenciar o conhecimento e promover sua captura e compartilhamento sem criar gargalos no processo de desenvolvimento de software ?**

ESC fortemente baseada em conhecimento x agilidade tende a neglienciar captura e armazenamento de conhecimento

Identificar conhecimentos úteis e propor técnicas/métodos para capturá-los, armazená-los, recuperá-los e utilizá-los.

Representar o conhecimento de tal forma que encoraje as equipes a pesquisá-lo e recuperá-lo.

→ Uso de ontologias para atribuir semântica a dados de repositórios, a fim de identificar “quem sabe o quê” e criar uma rede de conhecimento na organização.



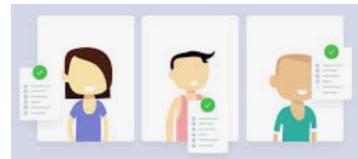
## Questões de Pesquisa e Algumas Propostas de Solução

**QP7. Como coletar feedback continuamente dos usuários? Como usar o feedback dos usuários para apoiar a melhoria de processos e produtos e identificar novas oportunidades de negócios?**

Identificar métodos e técnicas para apoiar a obtenção de feedback dos usuários e estimular o usuário a explicitar suas **impressões implícitas**.

Uso de dados de **experimentação contínua**.

→ Uso de **ontologias** para integrar dados de diferentes fontes.



## Desafios

## Desafios (apenas alguns deles...)

**Win the war, not the battles:** foco na visão holística ponta a ponta do processo ao invés de em cada técnica/prática – algumas podem não ser necessárias/adequadas para a organização.

**Contexto e Cultura:** devem ser levados em consideração para definir que práticas de ESC serão adotadas. Além disso, ESC requer mudança de cultura, o que pode não ser trivial.

**Focar não só em velocidade, mas em continuidade :** "*Speed is meaningless without continuity*" (Fitzgerald and Stol, 2017). Alcançar fluxo e continuidade é mais importante, em primeira instância, do que a velocidade.

## Desafios (apenas alguns deles...)

**Necessidade de descontinuidade em engenharia de software :** "*Creativity and innovation require discontinuous thinking*" (Fitzgerald and Stol, 2017). Em alguns casos mudanças abruptas e descontínuas (i.e., disruptivas) são necessárias.

**BizDev:** desalinhamento entre as expectativas e estratégias de negócio e vendas e a capacidade de a equipe de desenvolvimento entregar funcionalidades em curtos períodos e com qualidade.

**Medição:** definição de métricas apropriadas para promover desenvolvimento de software orientado a dados e experimentação contínua.



## Considerações Finais

63

## Considerações Finais

Termos como *Integração Contínua*, *Entrega Contínua* e *DevOps*, entre outros, fazem parte do dia a dia de muitas organizações.

O fenômeno "*contínuo*" indica claramente uma tendência comum de crescente necessidade de se estabelecer um fluxo de ponta a ponta entre a demanda do cliente e a entrega rápida do produto ou serviço.

Vai além das práticas ágeis e faz emergir um conjunto mais holístico de atividades contínuas.

## Considerações Finais

Não há um único caminho correto para adotar ESC.

Organizações têm seguido o caminho mais adequado para elas, adotando práticas gradualmente, cobrindo diferentes estágios/processos e categorias, e evoluindo de acordo com suas necessidades.

Muitos resultados têm sido alcançados, mas há, ainda, muitos desafios a ser superados.

Desafios são oportunidades para melhoria e inovação



# Engenharia de Software Contínua: Visão Geral, Desafios e Perspectivas

Monalessa Perini Barcellos

monalessa@inf.ufes.br

<http://www.inf.ufes.br/~monalessa>

Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO)  
Departamento de Informática  
Centro Tecnológico  
Universidade Federal do Espírito Santo

