

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



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



17 a 19 de outubro de 2024



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 DI/PPGI/UFES



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>

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



Camila



João Paulo



Monalessa



Ricardo (in memoriam)



Veruska



Vítor



Giancarlo



Renata

MEMBROS SÊNIOR

MEMBROS EXTERNOS

ENGENHARIA DE SOFTWARE CONTÍNUA: VISÃO GERAL, DESAFIOS E PERSPECTIVAS



- ✓ Criado em 2006 (18 anos de existência)
- ✓ + 600 publicações nacionais e internacionais
- ✓ + 70 alunos formados (graduação e pós)
- ✓ + 20 orientações em andamento




3

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

MEMBROS SÊNIOR



Camila



Monalessa



Patrícia



Vítor

ENGENHARIA DE SOFTWARE CONTÍNUA: VISÃO GERAL, DESAFIOS E PERSPECTIVAS



**LABORATÓRIO DE PRÁTICAS EM
ENGENHARIA DE SOFTWARE
“RICARDO DE ALMEIDA FALBO”**

labes.inf.ufes.br [@labes.ufes](https://twitter.com/labes.ufes)



Antes de iniciar o projeto, o processo pode ser aplicado exatamente como proposto ou ajustes ocorrem à medida que o projeto avança e o gerente, junto com a equipe, aprendem o que está dando certo e o que precisa de ajustes.

A figura abaixo mostra uma visão geral do processo:

```

    graph TD
        PM[1 PM Canvas] --> UserStories[2 User Stories (Método)]
        UserStories --> Esboço[3 Esboço (MVPS)]
        Esboço --> Critério[4 Critério de Adequação]
        Critério --> Protótipo[5 Protótipo]
        Protótipo --> ComoDemonstrar[6 Como demonstrar]
        ComoDemonstrar --> Modelo[7 Modelo de Classes]
        Modelo --> Arquitetura[8 Arquitetura]
        Arquitetura --> Backlog[9 Backlog]
        Backlog --> Labes[10 Labes]
        Labes --> VisaoGeral[Visão Geral]
    
```

Como mostra a figura, o processo segue etapas循序 para estabelecer um contexto inicial (PM Canvas & User Stories / Backlog), a partir do qual o processo entra em um ciclo iterativo, no qual é estabelecido um critério de adequação para o projeto, que é usado para guiar a criação de um protótipo, que é usado para demonstrar como o projeto deve ser implementado, que é usado para definir um modelo de classes, que é usado para definir a arquitetura do sistema, que é usado para gerenciar o backlog de tarefas, que é usado para gerenciar o projeto em si.

4

2

Escopo desta palestra

Motivação

Visão Geral

Questões
de Pesquisa

Propostas
de Solução

Perspectivas

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:

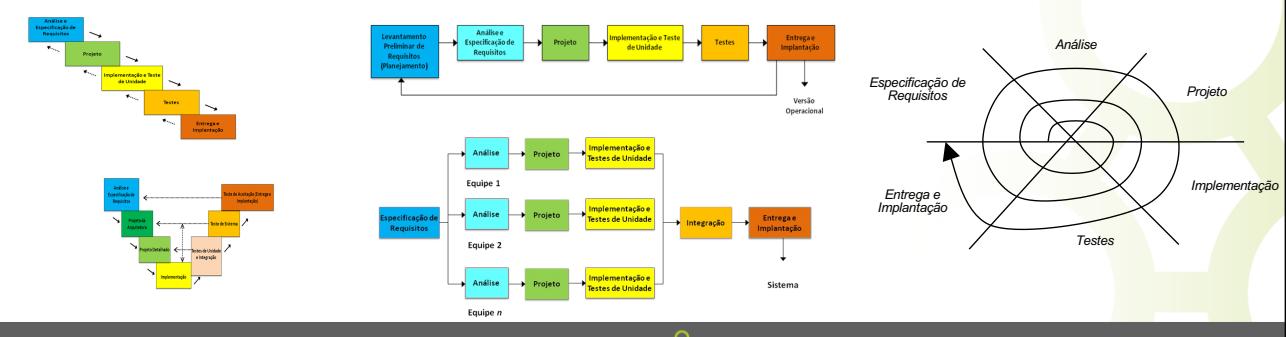
- ✓ Entregas mais rápidas e frequentes
- ✓ Mudanças frequentes em requisitos (requisitos x hipóteses)
- ✓ Menor tolerância a falhas
- ✓ Adaptação a novos modelos de negócio (SaaS, decisões baseadas em feedback de usuário,...)
- ✓ Softwares mais complexos (capacidade de interoperar, evolução por tempo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



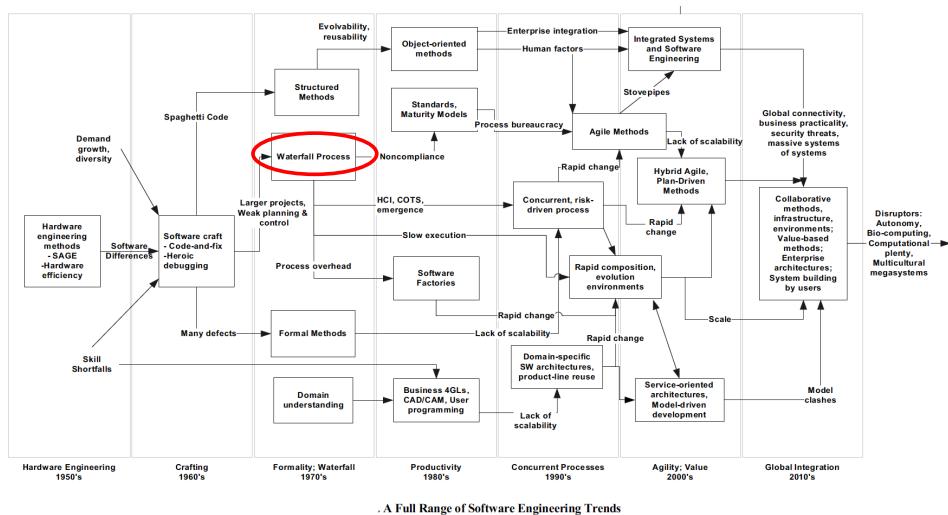
Introdução

Mas... Essa mudança não aconteceu “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.

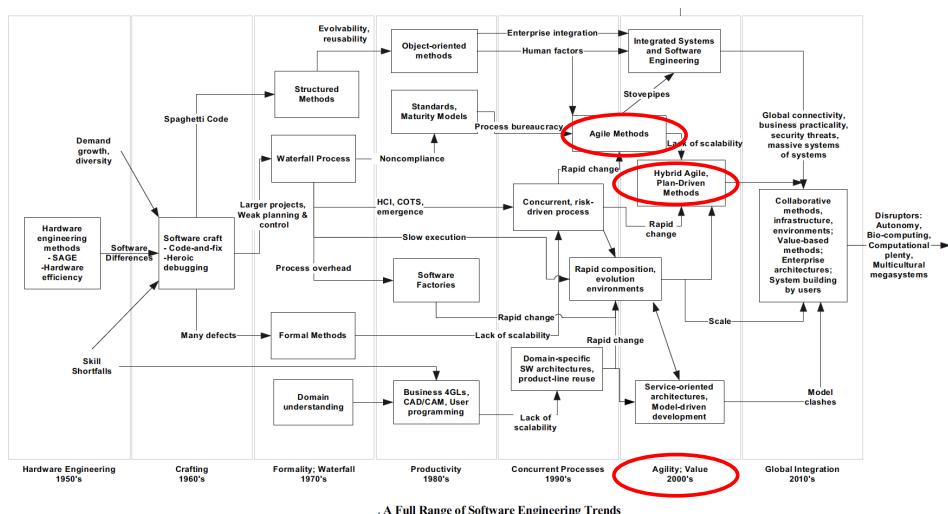


Introdução



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.

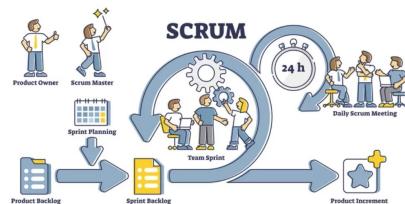
Introdução



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

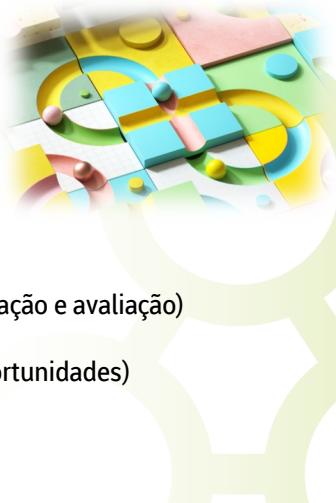
A adoção de práticas ágeis evidenciou que:

- ✓ **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 para operação gera problemas
- ✓ **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:

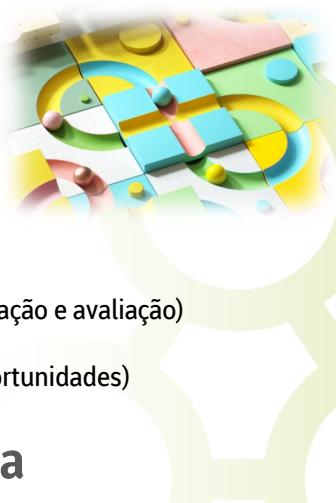
- ✓ **Diminuir descontinuidades** entre negócio, desenvolvimento e operação
- ✓ **Visão holística** do processo de desenvolvimento
- ✓ **Fluxo contínuo** entre atividades do ciclo de vida do software
- ✓ **Alinhamento contínuo** ao negócio
- ✓ Realizar ações **continuamente** (e.g., planejamento, construção, 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)



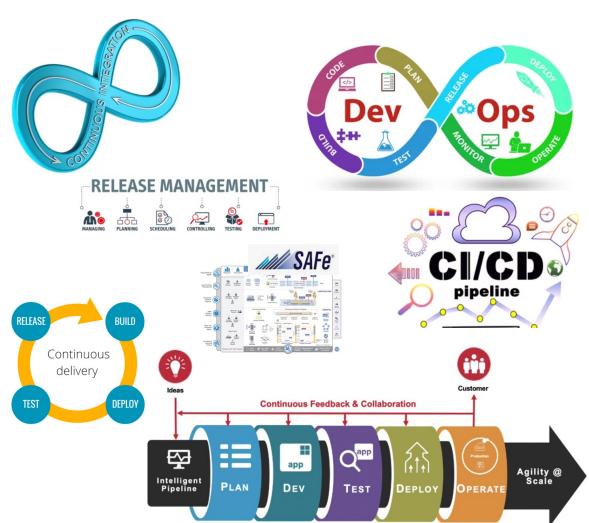
Engenharia de Software Contínua

O que é Engenharia de Software Contínua?

Uma Visão Geral

15

Engenharia de Software Contínua



Experiment-driven Development
Continuous Experimentation



... e mais

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



Monalessa P. Barcellos

ERI ES 2024

17

Engenharia de Software Contínua

Algumas bases da 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 funcão 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.

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



Monalessa P. Barcellos

ERI ES 2024

18

Engenharia de Software Contínua



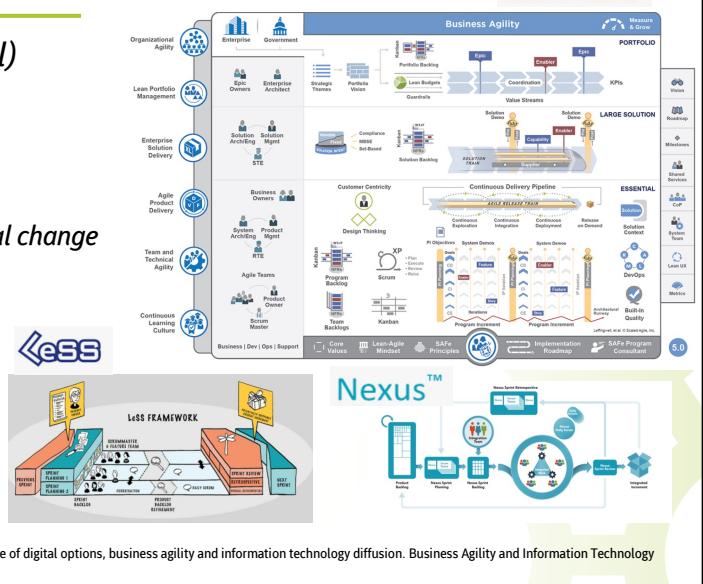
Ágil Escalado (Agilidade Organizacional)

Agilidade além da função de desenvolvimento

*"The ability of organizations to sense environmental change and respond appropriately" **

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.

Princípios

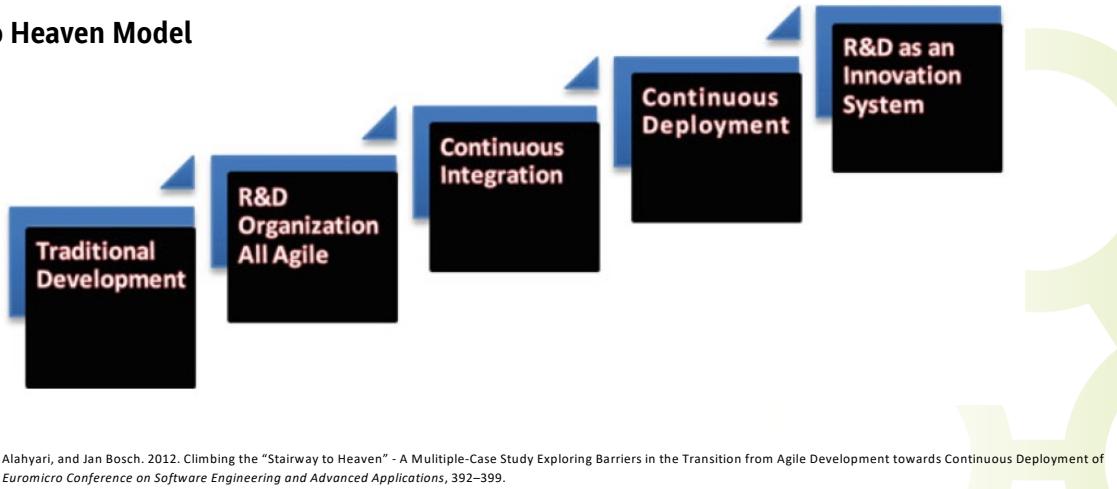


Humble, J. , Molesky, J. , 2011. Why enterprises must adopt devops to enable continuous delivery. *Cutter IT J.* 24 (8), 6–12

Engenharia de Software Contínua

Algumas propostas que proveem uma visão geral da ESC:

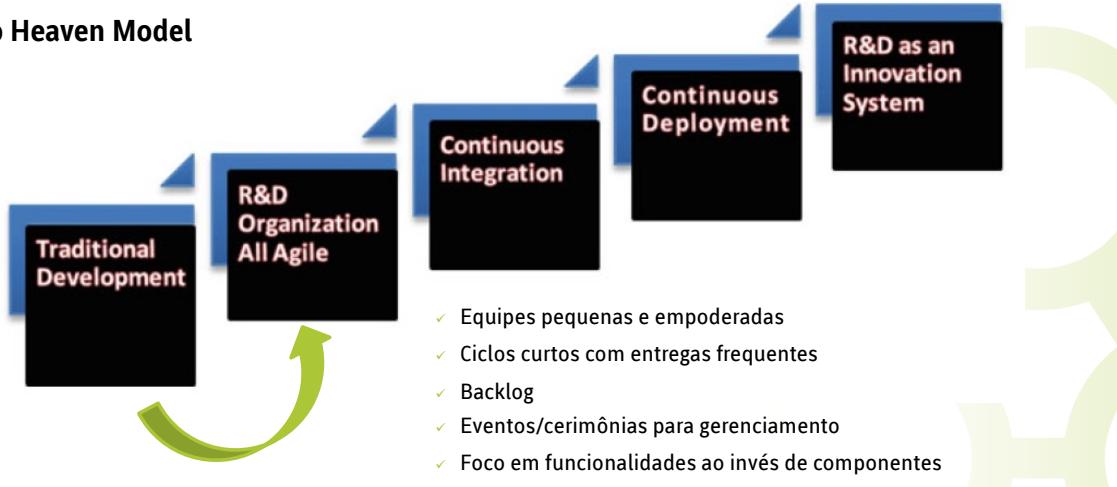
Stairway to Heaven Model



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.

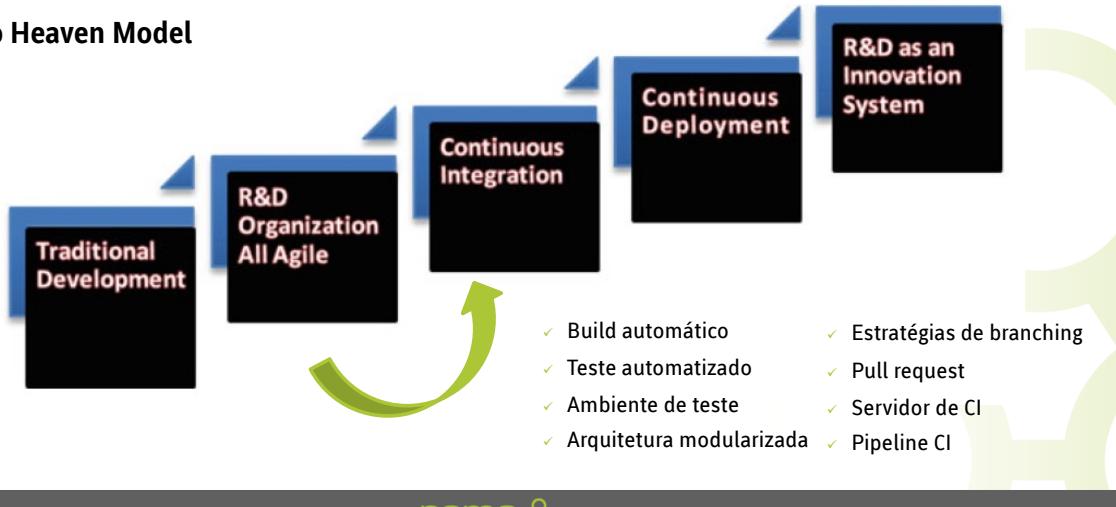
Engenharia de Software Contínua

Stairway to Heaven Model



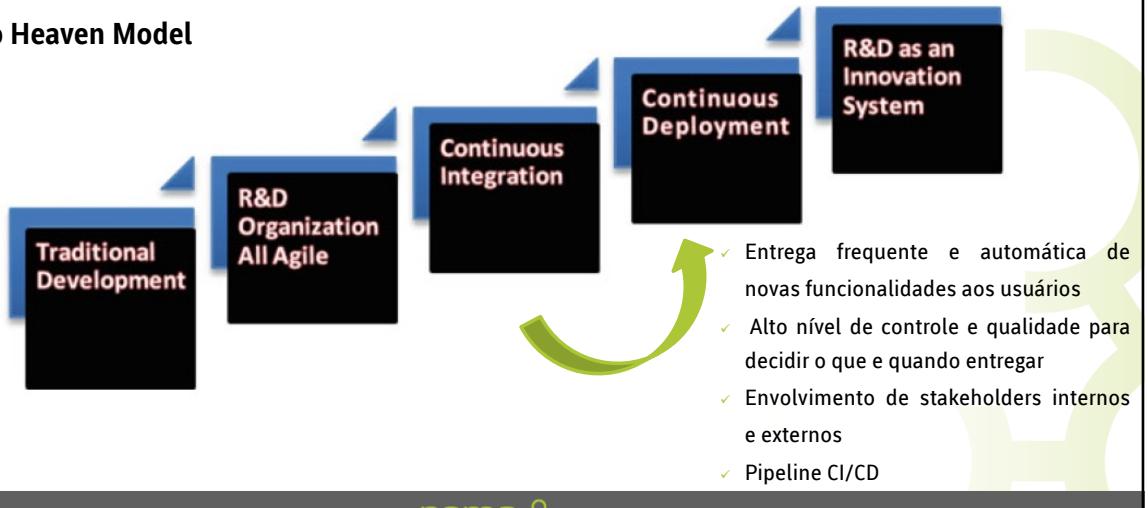
Engenharia de Software Contínua

Stairway to Heaven Model



Engenharia de Software Contínua

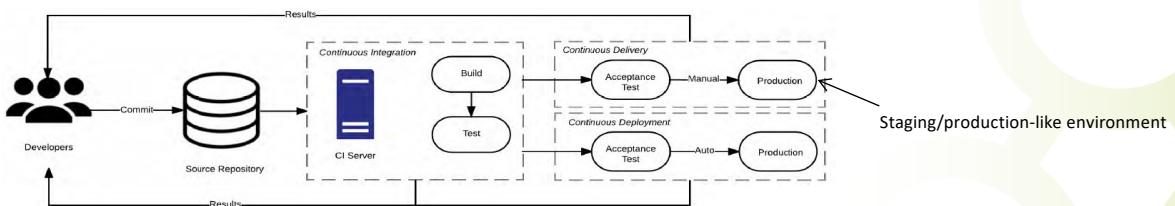
Stairway to Heaven Model



Engenharia de Software Contínua

Continuous Delivery x Continuous Deployment

- ✓ Diferem na forma (*Continuous Delivery* envolve alguma intervenção 'manual' - e.g., para aprovar o que será entregue)
- ✓ Diferem no ambiente onde a entrega é feita (*Continuous Deployment* entrega em ambiente de produção, enquanto *Continuous Delivery* pode entregar em um ambiente intermediário)

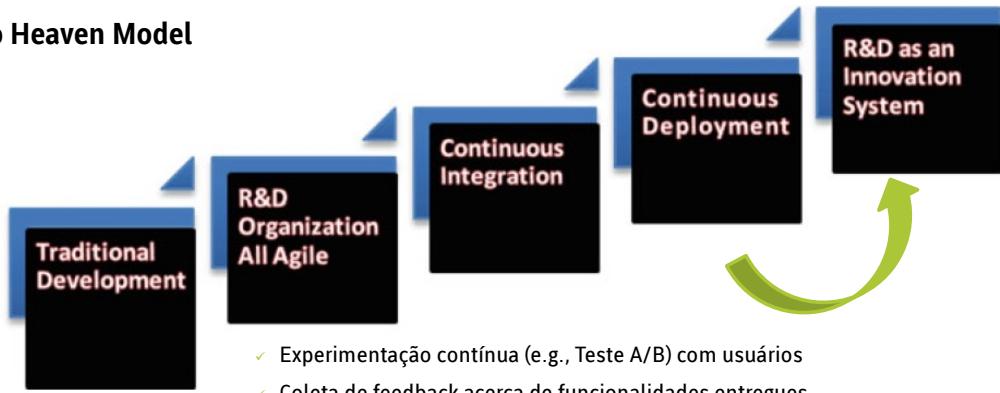


Há várias visões sobre esses conceitos (não há consenso).

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

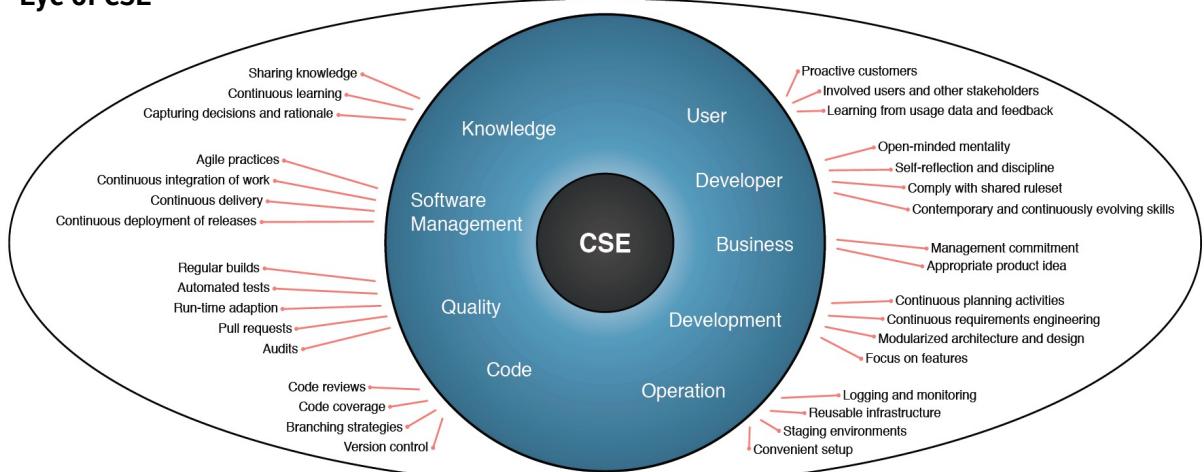


- ✓ Experimentação contínua (e.g., Teste A/B) com usuários
- ✓ Coleta de feedback acerca de funcionalidades entregues
- ✓ Uso de dados para identificar melhorias, novas features e oportunidades de negócio
- ✓ Alinhamento da estratégia da organização com o ecossistema do negócio

Engenharia de Software Contínua

Eye of CSE

33 práticas agrupadas em 9 categorias



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

ENGENHARIA DE SOFTWARE CONTÍNUA: VISÃO GERAL, DESAFIOS E PERSPECTIVAS



Monalessa P. Barcellos

ERI ES 2024

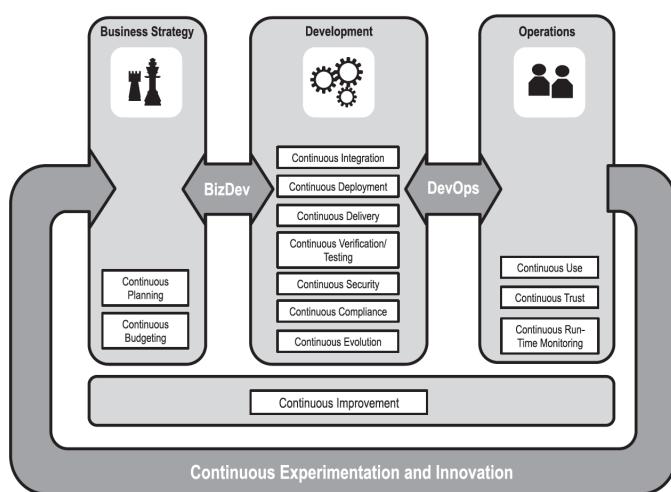
27

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: VISÃO GERAL, DESAFIOS E PERSPECTIVAS



Monalessa P. Barcellos

ERI ES 2024

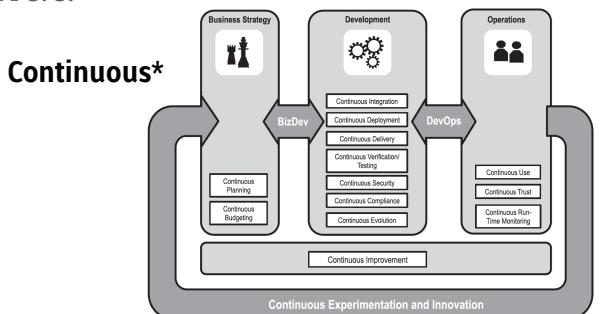
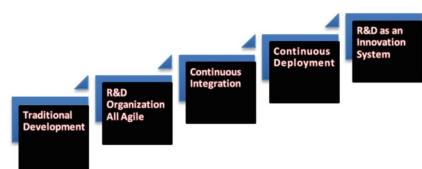
28

14

Engenharia de Software Contínua

Stairway to Heaven Model

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



Eye of CSE

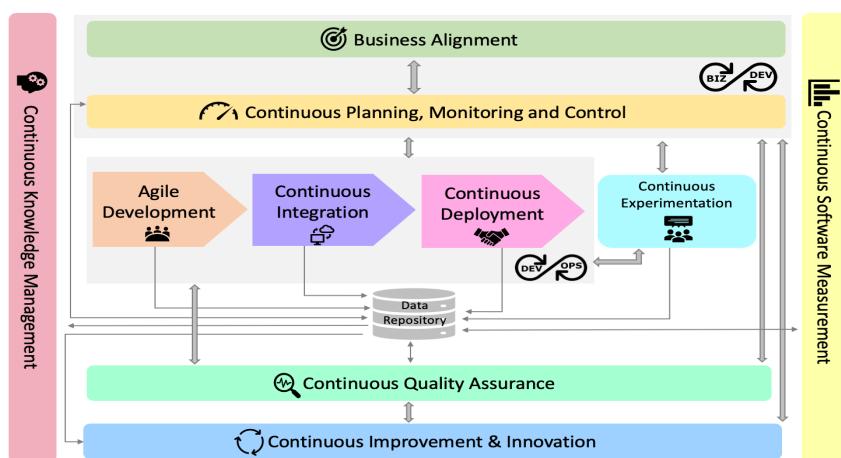
A quais processos as
práticas se
relacionam?



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

Engenharia de Software Contínua

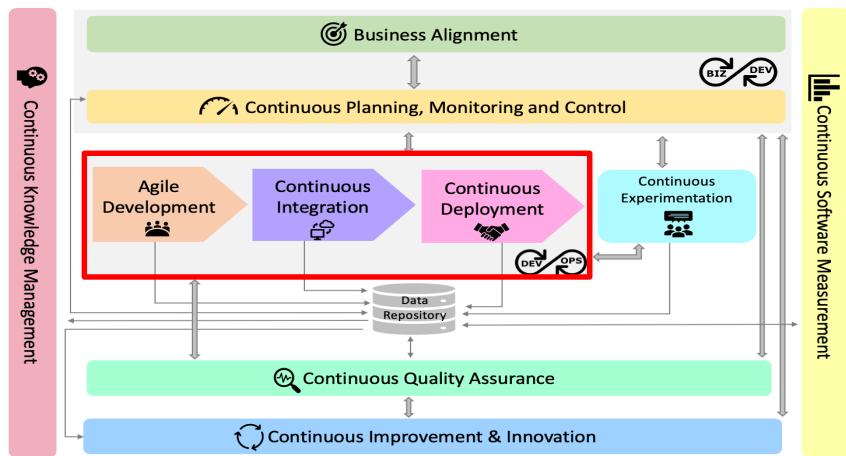
Continuous Software Engineering Framework



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

Engenharia de Software Contínua

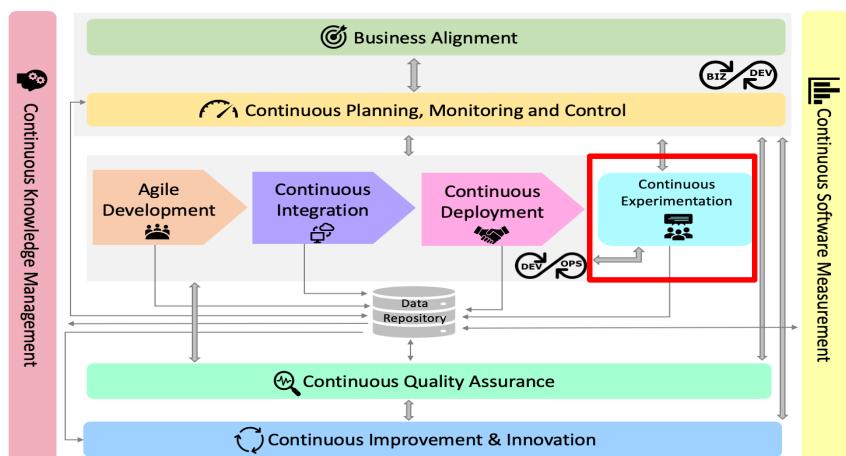
Continuous Software Engineering Framework



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

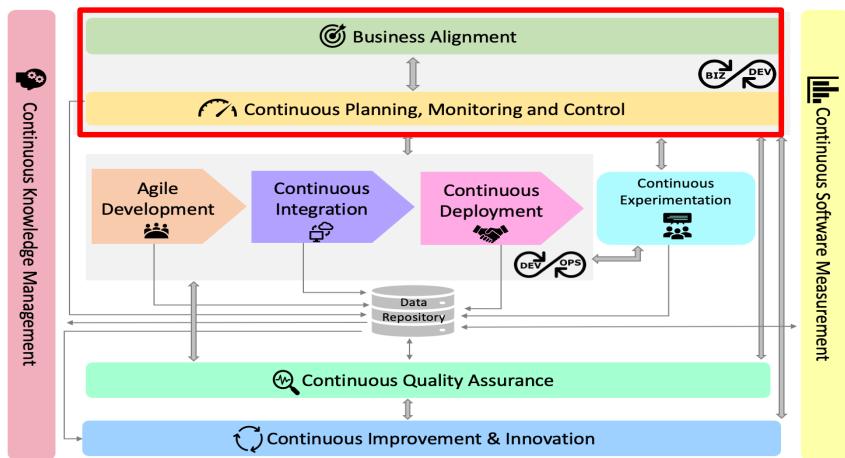
Continuous Software Engineering Framework



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: VISÃO GERAL, DESAFIOS E PERSPECTIVAS



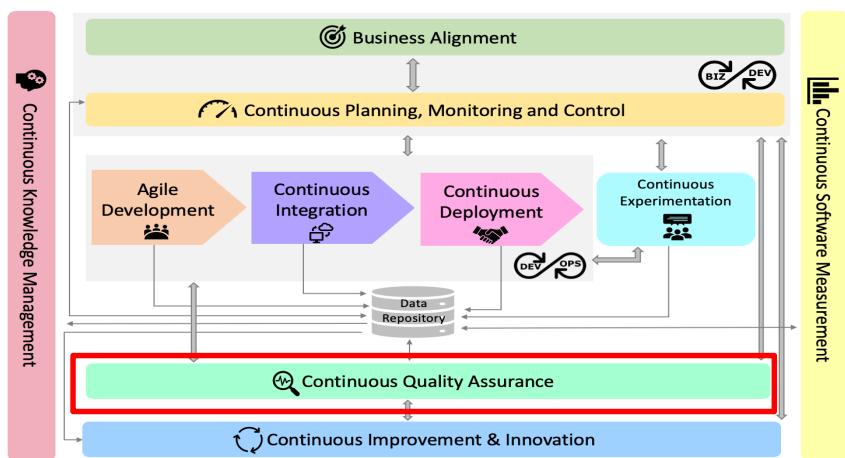
Monalessa P. Barcellos

ERI ES 2024

33

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



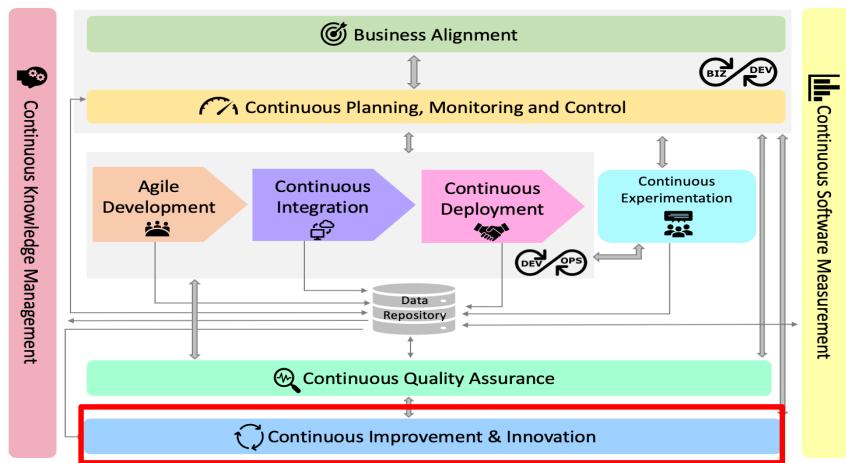
Monalessa P. Barcellos

ERI ES 2024

34

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



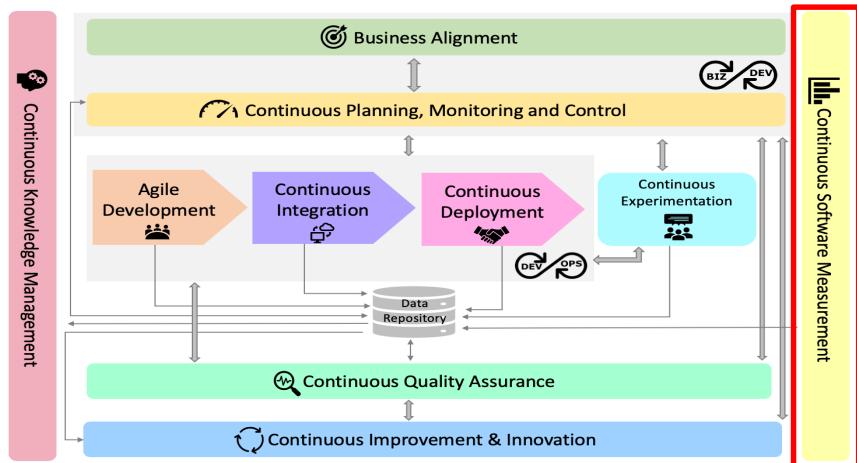
Monalessa P. Barcellos

ERI ES 2024

35

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



Monalessa P. Barcellos

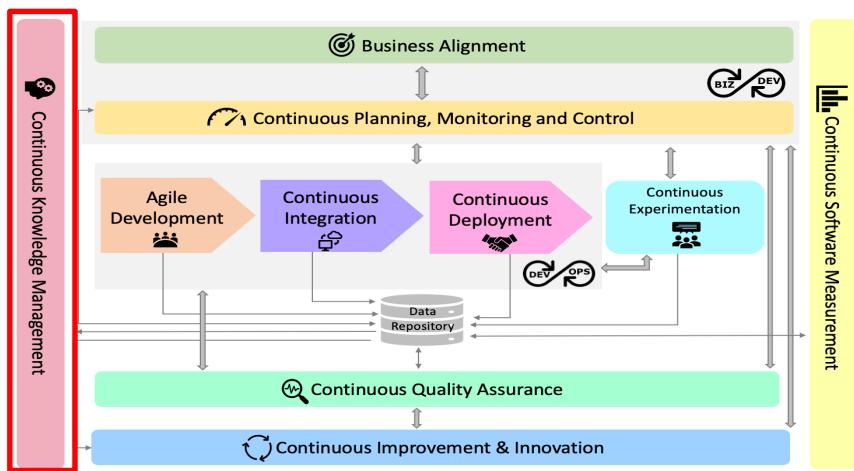
ERI ES 2024

36

18

Engenharia de Software Contínua

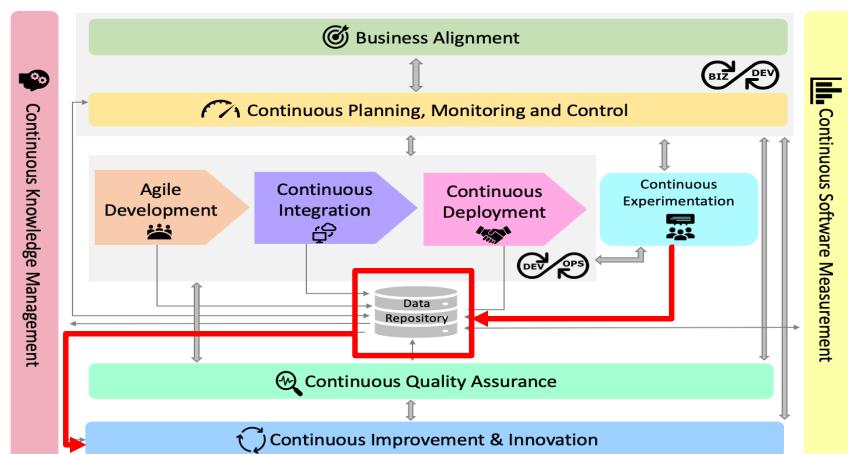
Continuous Software Engineering Framework



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

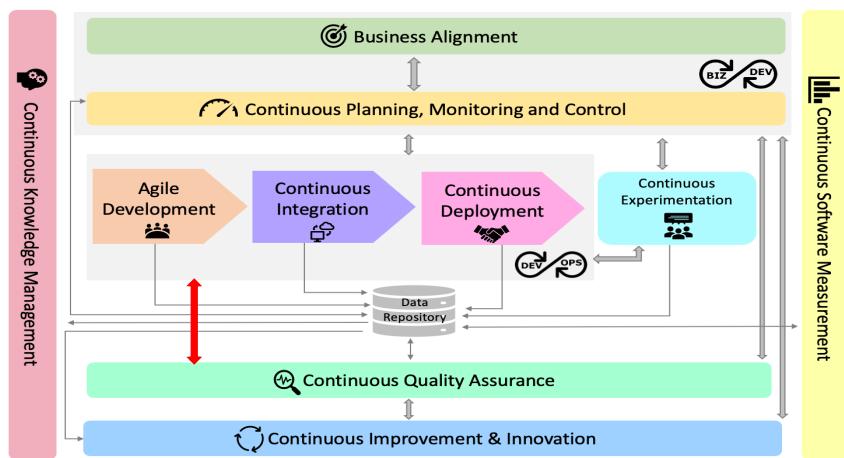
Continuous Software Engineering Framework



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



Monalessa P. Barcellos

ERI ES 2024

39

Questões de Pesquisa
Propostas de Solução

40

20

Questões de Pesquisa e Algumas Propostas de Solução

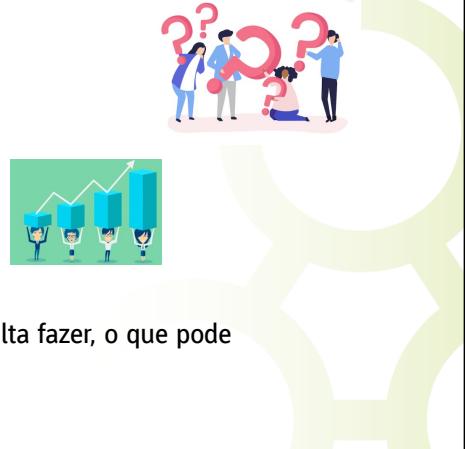
QP1. Como um organização pode identificar as práticas de ESC a ser implementadas e como evoluir/avançar na implementação das práticas?

É necessário considerar os diferentes contextos organizacionais.

Entender as necessidades de cada organização.

One size does not fit all!

Continuous Software Engineering



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

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.

Continuous Integration

Implementation of concepts and techniques related to Continuous integration and automated testing, such as CI/CD, automated build and testing, and test environments.

Statement	Adoption Level	Comments
C01 The software architecture is modular in order to allow automated testing.	Adopted	
C02 The software architecture is modular in order to allow automated builds.	Adopted	
C03 Tests run automatically, periodically, in a test environment.	Adopted	
C04 Clean code is integrated automatically in a test environment.	Adopted	
C05 Tests run automatically, periodically, in a test environment, to verify code coverage.	Adopted	
C06 Builds occur frequently and automatically.	Adopted	
C07 Builds are canceled if one or more tests fail.	Adopted	
C08 Code is integrated constantly and automatically by the (multidisciplinary) development team.	Adopted	
C09 Requirements verification is performed by the (multidisciplinary) development team.	Adopted	
C10 Code is integrated constantly and automatically.	Adopted	
C11 Continuous integration is used to do, e.g., test, scripts, etc.) performed in a repository.	Adopted	
C12 Check-in good practices are applied in the development trunk (e.g., use of tools such as Gitflow and Toggl Feature).	Adopted	
C13 The organization adopts practices that allow external organizations to act in the development of the project.	Inform the adopted practices	
C14 The organization adopts practices that allow external organizations to act in the development of the project process.	Inform the adopted practices	
C15 Data produced in continuous integration environments is stored in one (or more) data repository.	Inform the adopted practices	
C16 The continuous integration process is evaluated and improved continuously.	Adopted	
C17 Data produced in continuous integration environments is stored in one (or more) data repository.	Inform the adopted practices	
C18 The organization adopts practices for sharing knowledge related to continuous integration (e.g., internal lectures, tutorials, knowledge repositories, guild implementations).	Inform the adopted practices	

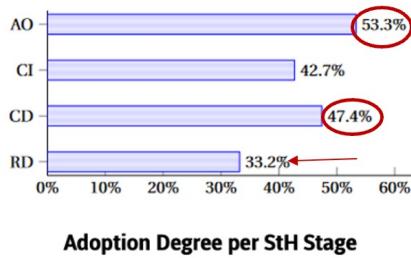
Questionário (76 práticas organizadas em 4 estágios do Sth)



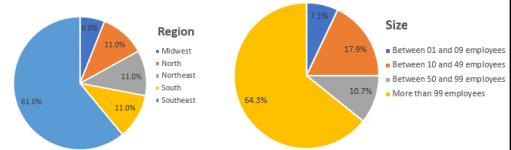
Questões de Pesquisa e Algumas Propostas de Solução

ESC no Brasil

Estudo com 28 organizações – abril/março 2022



- 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%



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

QP2. 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



Recursos de conhecimento (informam processos, atividades e artefatos de ESC)



Critérios

(organizacionais, tecnológicos, etc.)



Ferramentas

Continuous Software Engineering Tools Catalogue



Guidelines to set your CSE Suite

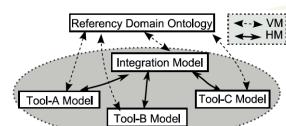


Questões de Pesquisa e Algumas Propostas de Solução

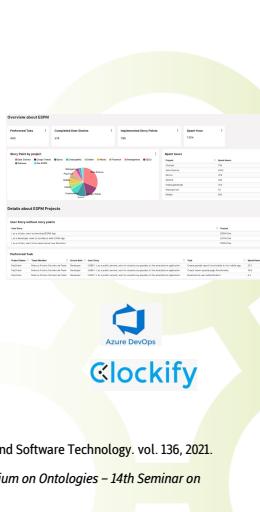
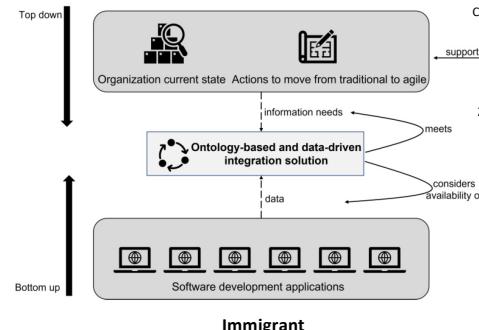
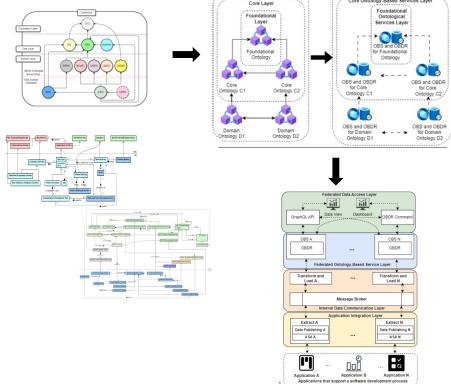
QP3. Como integrar dados de diversas ferramentas visando ao desenvolvimento de software e à tomada de decisão orientados a dados?

- Mineração de repositórios de software

- Soluções de interoperabilidade semântica



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.

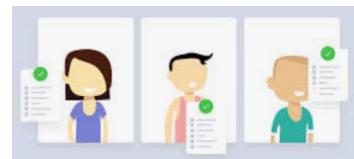
Questões de Pesquisa e Algumas Propostas de Solução

QP4. 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?

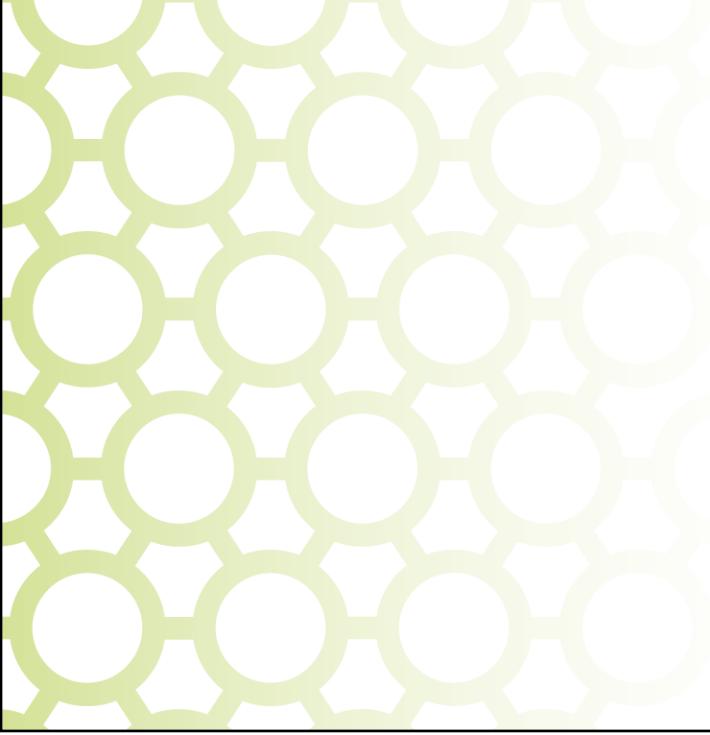
Aspectos humanos: identificar métodos e técnicas para apoiar a obtenção de feedback dos usuários e estimular o usuário a explicitar suas **impressões implícitas**.



Uso de dados de **experimentação contínua**.



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



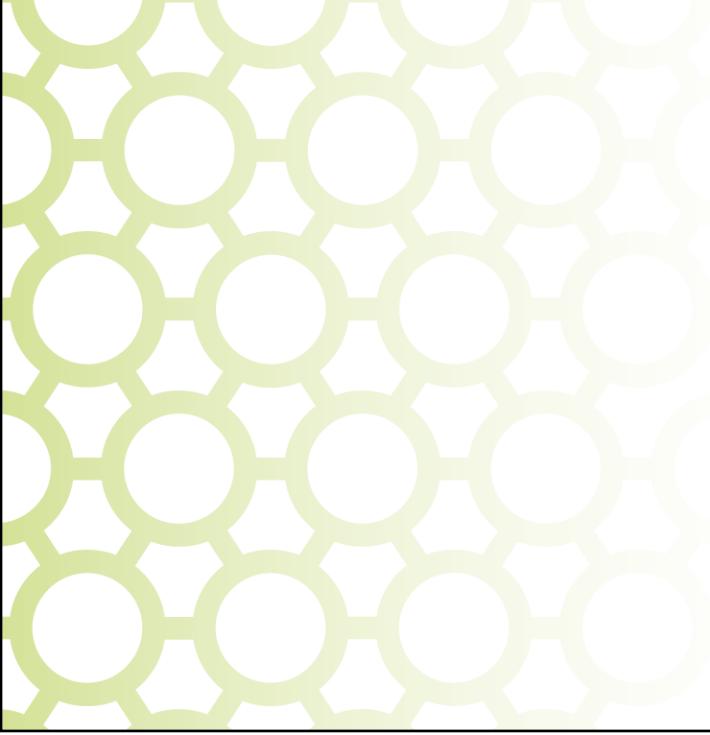
O caminho à frente

49

O que nos espera no caminho...

- Mais ferramentas, mais dados, mais heterogeneidade
- Necessidade de alinhamento entre o negócio e o desenvolvimento (BizDev)
- Necessidade de equilíbrio entre qualidade e velocidade
- Aumento de preocupação com questões de segurança e privacidade
- Avanços relacionados à operação
- Equipes mais heterogêneas, novos modelos de colaboração
- Soluções de apoio à ESC utilizando ferramentas de IA
- Introdução de problemas devido ao uso de ferramentas de IA
-





Considerações Finais

51

Considerações Finais

ESC: ferramentas + práticas + técnicas/métodos

Inclui práticas ágeis e vai além: fluxo contínuo, visão holística, alinhamento ao negócio.

Não há um único caminho correto para adotar ESC.

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

Muitos resultados têm sido alcançados, mas há, ainda há um caminho com desafios à frente.

Desafios são oportunidades para melhoria e inovação



Agradecimentos

O estudos relacionados ao tema ESC têm sido realizados em colaboração com



ENGENHARIA DE SOFTWARE CONTÍNUA: VISÃO GERAL, DESAFIOS E PERSPECTIVAS

nemo

Monalessa P. Barcellos

ERI ES 2024

53

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



17 a 19 de outubro de 2024



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

54

27