

# Aula 6: Gestão de Liberação

## Apresentação

---

Abordaremos um tópico chave para a Gerência de Configuração, no caso o do Gerenciamento de Liberação. Explicaremos os fundamentos, tipos de liberação, bem como boas práticas gerais específicas para o tópico de Gerenciamento de Liberação.

Daremos ainda uma visão geral sobre o uso voltado para Gerenciamento de Liberação de uma ferramenta consagrada, no caso Jenkins.

## Objetivo

---

- Examinar os fundamentos e as boas práticas gerais do Gerenciamento de Liberação;
- Identificar tipos de liberação;
- Identificar e conhecer a ferramenta Jenkins.

## O Gerenciamento de Liberação e seus Fundamentos

---

Vimos que itens de configuração são fundamentais como elementos habilitadores de projetos, operações e outros empreendimentos, daí a necessidade da gestão cuidadosa de suas configurações. Vimos também que itens de configuração novos ou alterados podem ser resultado direto ou produto desses empreendimentos.

### Comentário

Independentemente do papel que tenham, é normal que, ao longo de seus ciclos de vida, itens de configuração passem por muitas alterações, necessitando, portanto, do Controle de Mudanças, de forma que se mantenha níveis aceitáveis de rastreabilidade e supervisão sobre cada mudança solicitada.

O Controle de Mudança cuida do aceite (ou rejeite) e da implementação da mudança em si, mas é necessário algo a mais para

que essa mudança chegue ao ambiente operacional, ou seja, o ambiente disponibilizado aos usuários, sejam estes usuários internos ou externos à organização.

O Gerenciamento de Liberação é esse algo a mais, e tem por propósito assegurar a distribuição (ou liberação) de itens de configuração para fora das fronteiras das atividades de desenvolvimento.

**Estão inclusos no Gerenciamento de Liberação a identificação, empacotamento e entrega direta dos elementos componentes de um produto, como, por exemplo, os arquivos executáveis, documentação, manual de usuário, notas de liberação e dados de configuração.**

É importante mencionar que existe uma íntima ligação entre liberação e testes, sejam eles de verificação (que focam em defeitos - conduzidos por equipes de teste e / ou qualidade) ou de validação (que focam em conformidade com os requisitos - conduzidos por usuários ou clientes).

## Saiba mais

Procedimentos de teste podem estar diretamente acoplados aos de liberação, ou podem ser representados por procedimentos totalmente independentes. O nosso foco é na liberação em si, portanto não daremos enfoque aos testes.

Considerando que mudanças basicamente ocorrem a todo momento, fica óbvio que não é prático gerar uma liberação correspondente para toda e cada mudança. Dessa maneira, uma preocupação do Gerenciamento de Liberação é determinar quando são justificáveis a formação e emissão de uma liberação.

## Exemplo

Coisas que vão desde a severidade dos problemas sendo endereçados, até mensurações operacionais que apontem para uma alta densidade de defeitos originários de liberações anteriores podem ser elementos que afetam a decisão.

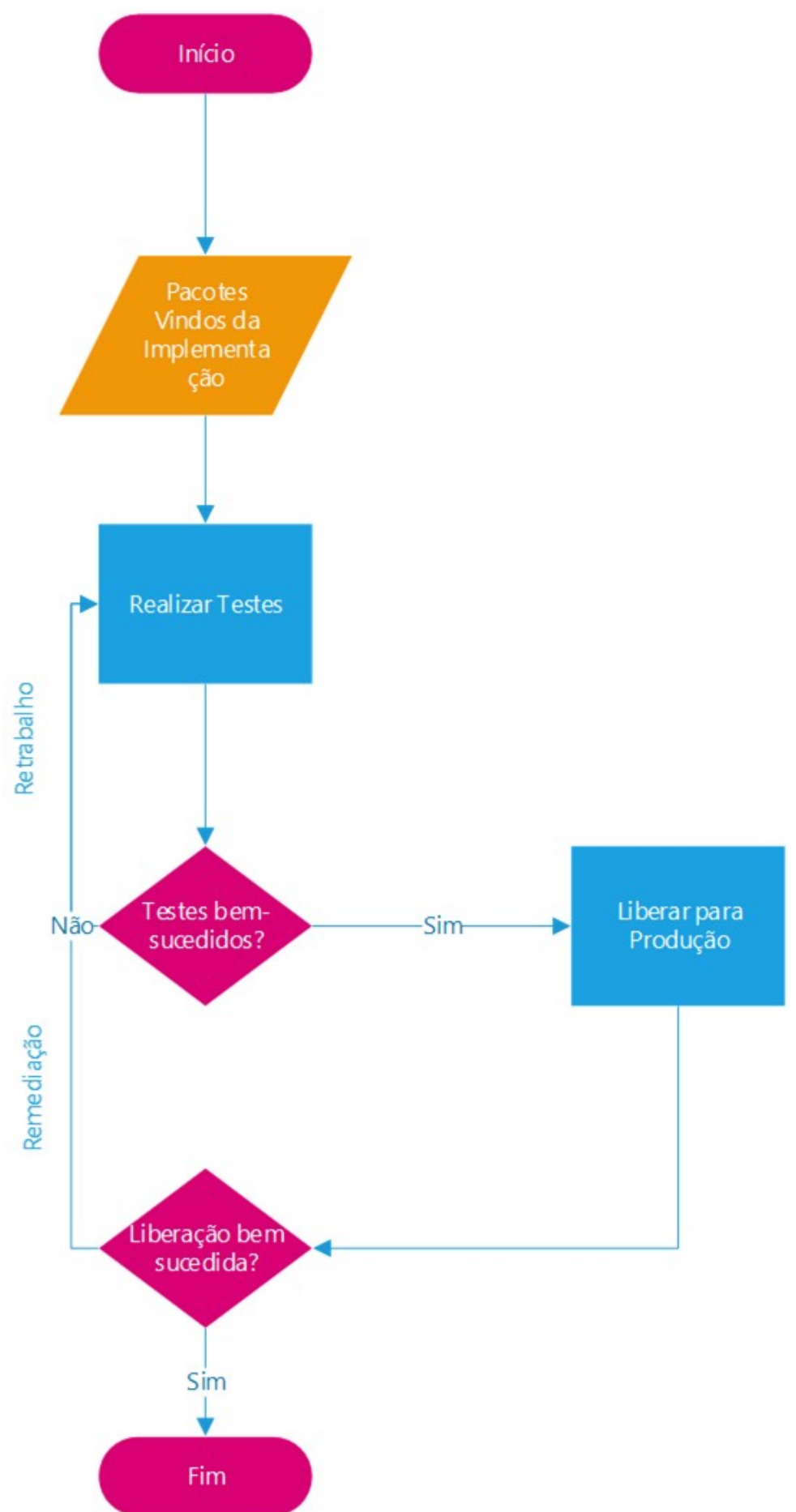
Uma vez que seja decidido que uma liberação é necessária, o empacotamento deve identificar claramente que itens de configuração componentes do produto sendo liberado devem ser entregues, e então deve selecionar as variantes corretas desses itens de configuração.

Por mais que os itens de configuração e a liberação em si sejam compostos por elementos intangíveis, pois se trata afinal de *software* e/ou documentação, pode-se dizer que dispõem de informação física, documentada em um **documento de descrição de versão**.

As notas de liberação normalmente irão descrever as novas capacidades, problemas conhecidos e os requisitos de plataforma necessários para a operação adequada do produto. O pacote a ser liberado também pode conter instruções para instalação e/ou atualização.

As informações de atualização podem ser mais complicadas de serem adquiridas e formalizadas, pois pode ser o caso de que alguns usuários podem estar usando versões que foram liberadas há muito tempo, estando, portanto, bem desatualizadas.

Esse é inclusive um motivo pelo qual é possível que seja delegada ao Gerenciamento da Liberação a responsabilidade por manter a rastreabilidade da distribuição do produto em uso pelos diferentes usuários. Outro ponto importante, e que também pode ser usado para esse propósito, é a implementação de algum mecanismo que assegure a integridade do item liberado, como por exemplo algum tipo de assinatura digital, podendo assim ter o propósito duplo de servir como incentivo para adoção das novas versões sendo liberadas.



📷 Possível Fluxos Integrados de Liberação e Teste. Fonte: Autor

## Os Diferentes Tipos de Liberação

Vimos que convém definir critérios claros e objetivos para que liberações sejam emitidas, já que existe uma forte ligação entre os dois conceitos. A principal razão das liberações existirem é o de transicionar mudanças para o ambiente operacional, e não existe uma relação 1 para 1 entre eles, além do fato de que existem tipos diferenciados de mudanças, bem como tipos diferenciados de liberação.

Novamente, como fizemos no caso do Controle de Mudanças, para o Gerenciamento de Liberação iremos recorrer à *Information Technology Infrastructure Library* (ITIL) para definir diferentes tipos de liberação.

### Liberações baseadas no nível de impacto

Listaremos três tipos, e é importante ressaltar que cada um deles pode implementar de uma a n mudanças por vez.

#### Liberações Maiores

São liberações que têm como objetivo primário o de entregar novas funcionalidades, em geral, de amplo escopo. São exemplos desse tipo de liberação grupos de telas ou relatórios, novos módulos, novas integrações etc. Como objetivo secundário, podem corrigir problemas, inclusive eliminando soluções temporárias que foram introduzidas via liberações menores ou mesmo emergenciais.

## Saiba mais

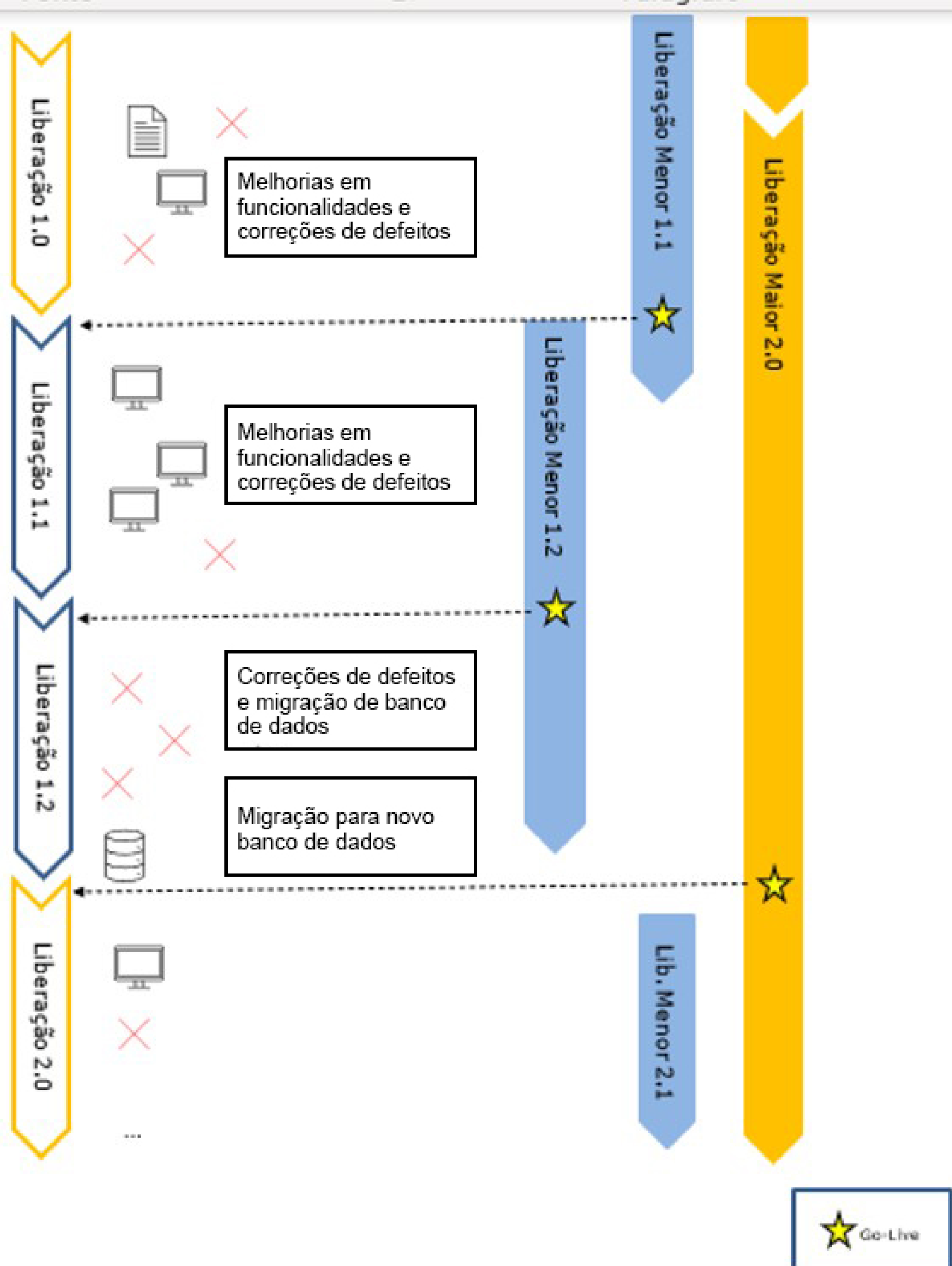
Se isso ocorrer, a liberação maior substitui e elimina as liberações menores e emergenciais que a precederam. É comum que toda liberação, ao ser executada, gere a atualização da numeração da versão de um item de configuração.

No caso das liberações maiores, pode-se definir uma forma específica de avanço de numeração versão, de forma a diferenciar a liberação maior das demais. Um exemplo possível pode ser avançar um incremento completo, como versão 2.40 para versão 3.0.

### **Liberações menores**

São liberações que têm como objetivo primário o de implementar correções para problemas existentes em uma linha de base ou estado acordado de um item de configuração. Liberações menores não tendem a adicionar novas funcionalidades, mas, se o fizerem, tal funcionalidade nunca será ampla, sendo restrita a pequenas melhorias, tais como uma nova tela, um novo relatório etc.

Essas liberações devem incrementar o número de versão de maneira diferente das liberações maiores. Um exemplo poderia ser 2.40 para 2.50.



📌 Possível Fluxo para Mudança Normal. Fonte: Adaptado de SAP.com.

### Liberações Emergenciais

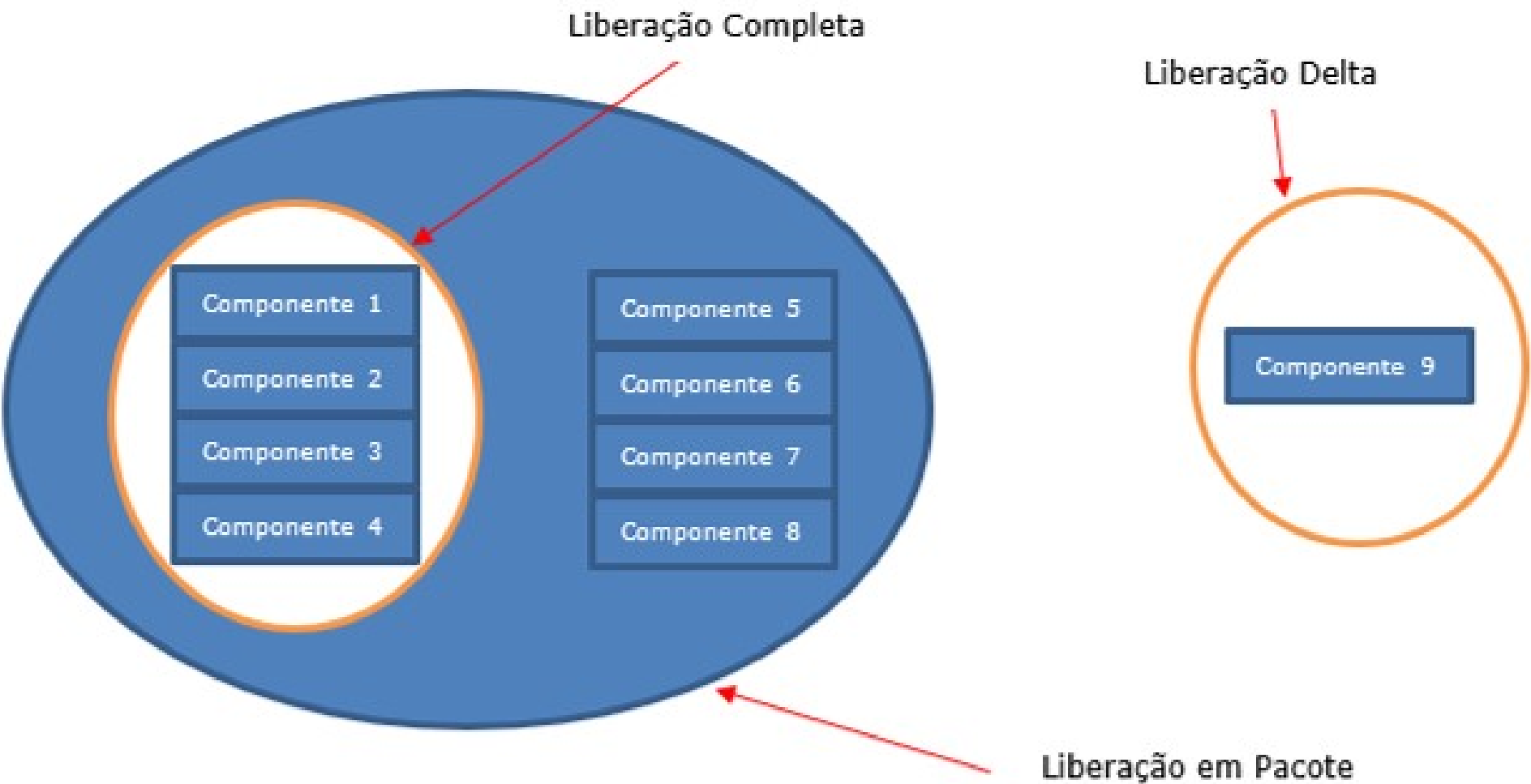
Liberações emergenciais normalmente se destinam a implementar mudanças emergenciais, ou seja, endereçam consertos rápidos para problemas inesperados ou soluções preventivas para problemas que estão na iminência de ocorrer.

Atenção

Liberações emergenciais não devem ser banalizadas, devendo ser usadas com parcimônia, apenas como último recurso. Mudanças emergenciais alteram o versionamento de sua própria maneira, como por exemplo de 2.40 para 2.4.1.

**Liberações baseadas no nível de agrupamento**

Listaremos mais três tipos, que novamente podem implementar de uma a n mudanças por vez. A categorização agora se dá com base na forma como agrupam os itens de configuração sendo liberados.



 Possível Fluxo para Mudança Emergencial. Fonte: Adaptado de Configuration Management Principles and Practice.

Essas liberações são:

## [Liberações Delta](#)



Liberações que incluem apenas os Itens de configuração que foram de fato alterados, mesmo que façam parte de um grupo de itens de configuração que trabalham em conjunto. Um exemplo seria a liberação de uma única tela que foi alterada em um módulo de um sistema.

## [Liberações Completas](#)



Liberações que incluem todos os elementos de um grupo de itens de configuração que trabalham em conjunto, mesmo aqueles que não foram diretamente alterados. Um exemplo seria a liberação de um módulo de um sistema completo, mesmo que somente uma tela tenha sido alterada.


## [Liberações em Pacote](#)



Liberações que incluem várias liberações Delta e/ou completas. São agrupadas na estrutura de pacote por algum motivo, tais como interdependências, fazer melhor uso de recursos, cronogramas, prioridades do negócio etc.

**Atenção!** Aqui existe uma videoaula, acesso pelo conteúdo online

 Boas Práticas no Gerenciamento de Liberação

 Clique no botão acima.

O Gerenciamento de Liberação coordena a distribuição de itens de configuração fundamentais para os projetos, operações e para as organizações, certificando-se que essas liberações estão alinhadas com as prioridades do negócio.

Ao fazer isso, ajuda na identificação e gestão de riscos, na coordenação de recursos, na aderência a regulações e recomendações vindas de auditorias e mantém o desenvolvimento de *software* em linha com as necessidades dos usuários.

Para que tudo isso seja atingido, uma série de boas práticas existe e pode ser adotada conforme faça sentido. Entre essas práticas podemos citar:

- **Instituir um Gerente de Liberações com papel bem definido:** Embora essa seja uma prática consolidada em grandes organizações, não é uma prática tão difundida em organizações menores, sendo que outros profissionais, tais como os gerentes de projeto ou gerentes de mudança, acabam por assumir a função de gerente de liberação. Isso ocorre, dentre outros motivos:
  - A organização não ter recursos suficientes para manter um gerente de liberação;
  - Por se acreditar que não há volume de trabalho para justificar a existência de um gerente de liberação em tempo integral;
  - Pelo fato de o papel de gerente de liberação não ter sua importância reconhecida pela organização.

Independentemente do caso, é interessante que ao menos exista o papel bem definido de gerente de liberação, com responsabilidades claras, autoridade formal e definida e escopo de atuação difundido em toda a organização, mesmo que esse papel seja acumulado com outros.

Uma forma de endereçar todas essas considerações é estabelecer o rodízio dos gerentes de projeto na função de gerente de liberação.

Durante o mandato como gerente de liberação, o gerente de projeto ocuparia os dois cargos de maneira part-time. Assim sendo, não só se garante a disseminação do conhecimento acerca da liberação, mas também se assegura um mínimo nível de apoio.

- **Estabelecer um ciclo regular para as liberações:** Esse é um começo importante, pois estabelece uma linha de tempo clara a todas as partes interessadas, desde a engenharia até os clientes e usuários. Isso estimula esses usuários a se comprometerem com a definição dos requisitos em tempo hábil, pois sabem que a obtenção de correções e novas funcionalidades estão atreladas a um cronograma pré-acordado.

Isso é valioso, em especial para organizações que ainda não são ágeis. Ao mesmo tempo, essa prática dá visibilidade ao time de engenharia sobre as expectativas que virão, otimizando o uso dos recursos, bem como gerando o comprometimento com as entregas, desde que atreladas ao ciclo acordado.

- **Considerar um ciclo ágil para as liberações:** Com a popularização dos *frameworks* ágeis, como, por exemplo, Scrum e XP, é natural que surjam esforços para tornar as liberações igualmente ágeis. Nessas abordagens, é comum que o comitê de mudanças seja descentralizado, permitindo a facilitação do aceite e implementação de mudanças a cada iteração ágil.

Se as mudanças são flexibilizadas, a liberação também precisa ser, e uma das formas de se atingir esse objetivo é por meio do estabelecimento de um cronograma previsível de liberações atreladas ao próprio cronograma das iterações ágeis.

Como exemplo podemos citar o planejamento de liberações para ocorrerem a cada 12 semanas de iterações de desenvolvimento, o que normalmente irá corresponder a 6 iterações de 2 semanas, ou 4 iterações de 3 semanas, ou 3 iterações de 4 semanas.



Informações sobre variantes ou ramos de um item de configuração também podem ser usadas para determinar se uma mudança tem impactos em vários locais.

Ainda é possível usar metadados para determinar se algum item de configuração possui solicitações de mudança em aberto, tais como mudanças em processo de implementação ou mesmo aguardando a decisão final sobre se serão ou não implementadas.

- **Incorporar requisitos claros com critérios de aceitação ainda mais claros:** Embora essa não seja uma prática diretamente ligada à Gestão de Liberação, é importante que os profissionais envolvidos nas liberações sejam campeões suportando essa prática da melhor maneira possível.

Isso porque o Gerenciamento da Liberação funciona a pleno vapor quando se tem visibilidade muito clara e não-ambígua sobre quando os itens de configuração estão realmente prontos para serem entregues.

Ao se operacionalizar essa prática, evita-se ainda possíveis conflitos entre a gestão da liberação e a gestão de projetos, gestão de testes e outros.

- **Estabelecer uma infraestrutura de liberação o mais rapidamente possível:** Qualquer coisa necessária para a emissão do *software* / documentação e para que os usuários possam usá-los faz parte da infraestrutura de liberação. Só construir o software não é suficiente para garantir satisfação, é necessário assegurar que o acesso e uso ao software também ocorrerá da forma o mais corriqueira possível.

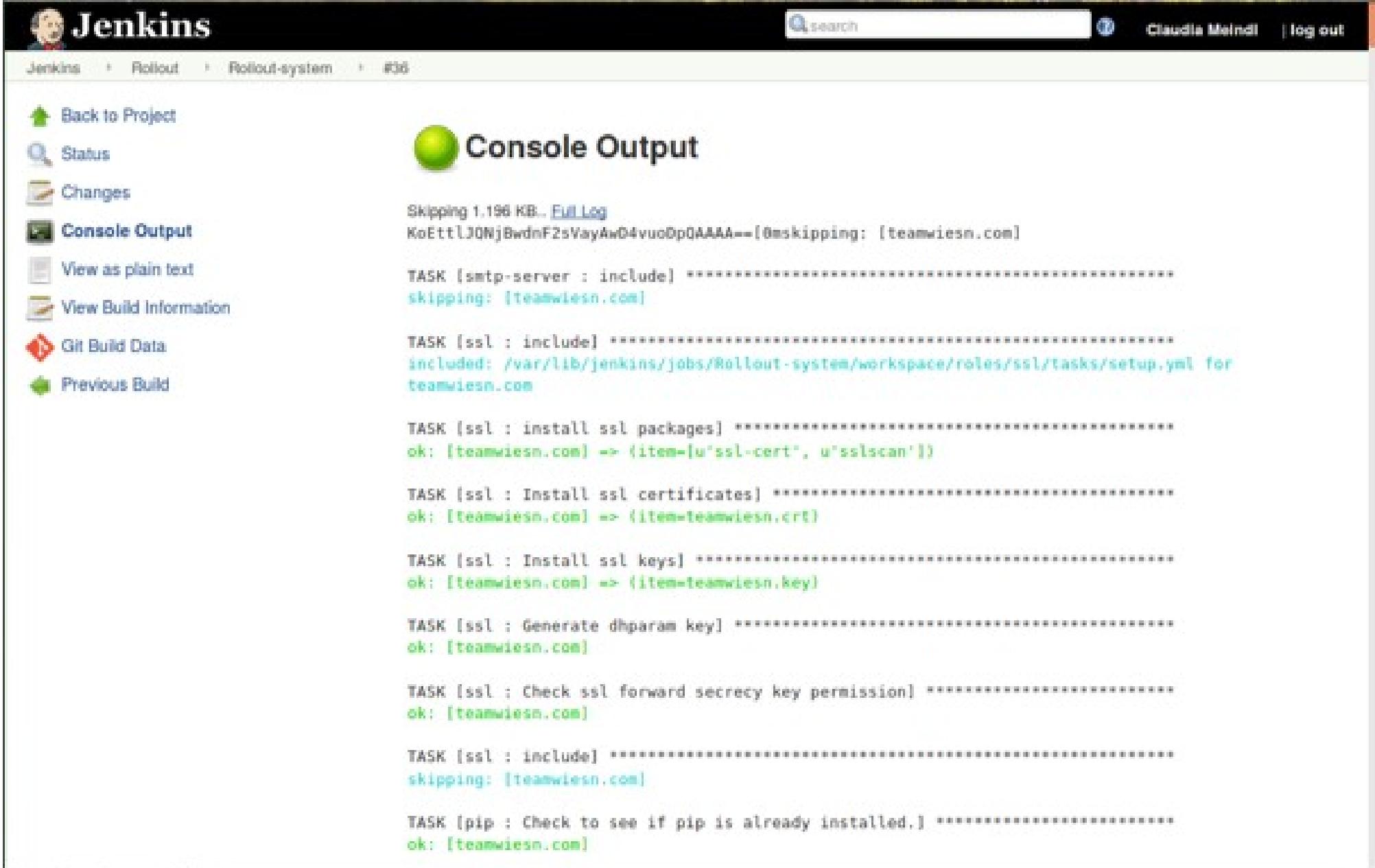
Para que isso ocorra, a determinação do que exatamente faz parte da infraestrutura de liberação precisa ser feita o mais cedo possível, antes mesmo que o time de engenharia finalize o produto. Podem estar inclusos elementos de hardware, software, redes, permissões de acesso, entre outros.

Considerar todos esses elementos ajuda a eliminar gargalos e surpresas desagradáveis que podem minar todo o esforço de desenvolvimento.

**Atenção!** Aqui existe uma videoaula, acesso pelo conteúdo online

## A Ferramenta Jenkins

A ferramenta Jenkins, originalmente conhecida como Hudson, é uma ferramenta *opensource* muito conhecida e usada no mercado por suas capacidades voltadas para a Integração Contínua e Gestão de Construção. Suporta esforços de desenvolvimento de software com base em diversas tecnologias, algumas mais legadas, tais como Java e PHP, e outras mais recentes, tais como Ruby on Rails e Groovy.



 Visão do console da ferramenta Jenkins. Fonte: Wikimedia Commons.

Apesar dos usos específicos previamente mencionados da ferramenta Jenkins, ela é flexível o bastante para ser usada como ferramenta de gestão de liberação. É ainda flexível para rodar como uma aplicação isolada (*standalone*) ou como um servidor de aplicação Java da mesma forma que é possível para o Apache Tomcat, por exemplo.

## Saiba mais

O Jenkins, para ser instalado, exige portanto a presença do Java na máquina hospedeira, no mínimo na versão 5, sendo recomendada a versão 5.0 do *Java Development Kit* (JDK). O Jenkins encontra-se disponível para Windows, Linux e Mac OS X e se integra bem com outras ferramentas, tais como Controle de Versão (Git e SVN, por exemplo).

Muito importantes nesse processo são as ferramentas de controle de ativos, tais como JFrog, e de gestão de construção, tais como o Maven, pois dão subsídio para que o Jenkins funcione como uma ferramenta de Gestão de Liberação, promovendo, de maneira transparente, versões intermediárias para versões definitivas, oficializando assim as suas liberações.



 Visão da página inicial do Jenkins. Fonte: Jenkins The Definitive Guide.

## Exemplo

É possível usar os metadados providos pelo JFrog, e então usar o Jenkins para controlar liberações em diferentes ambientes, tais como ambientes de produção, em de teste de qualidade, de teste de homologação etc.

Assim se mantém a rastreabilidade automatizada de quais versões de componentes são liberadas em conjunto para cada aplicação e para cada liberação, bem como quais componentes foram testados e receberam as aprovações devidas etc.

Infelizmente existem também desvantagens no uso do Jenkins. Veja a seguir.

- Por exemplo, a ferramenta é notoriamente pobre com relação a métricas, falhando em dar visibilidade a mudanças-chave na organização ou a desafios enfrentados por funcionários.
- Não é possível por exemplo identificar os funcionários mais produtivos, as mudanças mais recentes que causaram maior impacto positivo ou negativo na organização etc.
- Outro ponto negativo, de certa forma atrelado ao anterior, é a dificuldade que clientes do Jenkins têm com a rastreabilidade e contabilidade de mudanças feitas por diversos membros dos times de desenvolvimento. Isso porque a maioria das liberações e construções via Jenkins são feitas por um único usuário comum, tornando impossível o bom rastreamento das liberações. Toda a rastreabilidade precisa, portanto, ser transferida para outras ferramentas, tais como Git, e ser executada a nível de código-fonte.
- Mais um ponto de desvantagem é a dificuldade na colaboração, pois, por meio do Jenkins, não é possível para um desenvolvedor saber quais foram as mudanças realizadas por outros membros da equipe.
- A ferramenta também é um tanto inflexível quanto a customizações, não permitindo por exemplo a adaptação a coisas como fluxos de trabalho, reporte, gestão de mudanças voltadas para endereçamento de questões de segurança etc.

**Atenção!** Aqui existe uma videoaula, acesso pelo conteúdo online

# Atividade

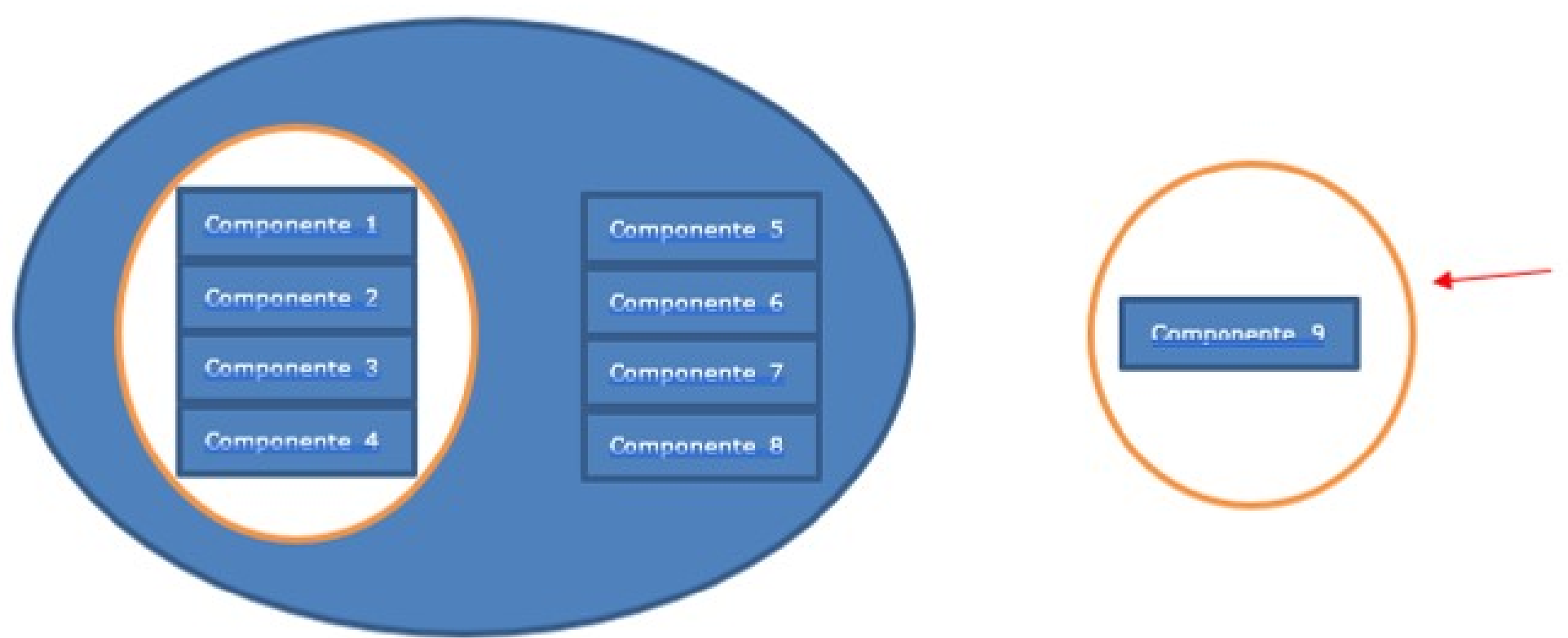
1. Uma organização provedora de software de contabilidade, e que começou apenas com dois funcionários, cresceu, recentemente atingindo 250 colaboradores que atendem a algumas dezenas de clientes diferentes. A alta administração deseja, portanto, abandonar o método de trabalho ad-hoc atual, passando a adotar boas práticas reconhecidas no mercado.

Para tal, contratou você como consultor externo, pedindo que foque seus esforços iniciais em melhorar as liberações, pois há uma tendência alarmante em se introduzir problemas em produção a cada emissão de novos produtos.

Como consultor, você foca inicialmente não necessariamente nas práticas de liberação em si, mas sim nas práticas de \_\_\_\_\_.

- a) Mudança.
- b) Configuração.
- c) Construção.
- d) Testes.
- e) Integração Contínua.

2. A liberação marcada pela seta é conhecida como:



- a) Liberação Delta.
- b) Liberação Menor.
- c) Liberação em pacote.
- d) Liberação Emergencial.
- e) Liberação Maior.

3. Endereçam consertos rápidos para problemas inesperados. Estamos falando de liberações do tipo:

- a) Padrão.
  - b) Normal.
  - c) Delta.
  - d) Completa.
  - e) Emergencial.
- 

4. Implementar um comitê de mudanças descentralizado:

- a) É uma prática não-recomendada, pois pode gerar muitos efeitos adversos no controle de mudanças e na gestão de liberação.
  - b) É uma prática não-recomendada, pois não existe algo como controle de mudanças ou gerenciamento de liberação ágil.
  - c) Não faz diferença, já que o comitê de mudanças é apenas um dos vários papéis envolvidos com mudanças e liberações.
  - d) É uma prática recomendada para toda e qualquer organização, pois é importante que se mantenha sintonia com tendências atuais.
  - e) É uma prática recomendada para os casos em que uma organização deseja dar passos rumo à agilidade e apresenta condições para isso, tal como apoio da alta administração.
- 

5. A ferramenta Jenkins é primariamente uma ferramenta de:

- a) Controle de Mudança.
  - b) Integração Contínua.
  - c) Desenvolvimento de *software*.
  - d) Gestão de Liberação.
  - e) Teste de *Software*.
- 

## Notas

### Mudança emergencial<sup>1</sup>

---

As mudanças emergenciais são basicamente o oposto das mudanças padrão: Geralmente, mais do que se caracterizarem por trazerem riscos médios a altos, exigem implementação o mais rapidamente possível, de forma que também possam ser operacionalizadas com grande rapidez. São, portanto, mudanças para “situações de crise”, e que requerem procedimentos à altura.

## Referências

---

AIELLO, B. Configuration Management Best Practices. 1.ed. Pearson, 2013.

BOURQUE, P.; FAIRLEY, R. Software Engineering Body of Knowledge (SWEBOK). 3.ed. IEEE Computer Society, 2017.

HAAS, J. Configuration Management Principles and Practice. 1.ed. Addison Wesley, 2003.

SMART, J. Jenkins The Definitive Guide Continuous Integration for the Masses. 1.ed. O'Reilly, 2011.

## Próxima aula

---

- Conceituação;
- Problemas e Soluções;
- Boas práticas da Integração Contínua;
- Software / Ferramentas de Integração Contínua.

## Explore mais

---

Pesquise a definição real do processo de Liberação e Implantação usada pelo Conselho de Justiça Federal.