

# Gerência de Configuração

## Aula 4: Gestão de Mudança

### Apresentação

---

Enfocaremos um tópico basilar da Gerência de Configuração, o Controle de Mudanças. Explicaremos seus fundamentos, bem como alguns tipos básicos de mudança e algumas boas práticas gerais do tópico, abordando Redmine e Gitlab, duas ferramentas consagradas no mercado e que, entre outras coisas, podem ser usadas para o Controle de Versão.

### Objetivo

---

- Examinar os fundamentos e as boas práticas gerais do Controle de Mudança;
- Identificar tipos de mudança;
- Identificar e conhecer as ferramentas Redmine e Gitlab.

### O Controle de Mudança e seus Fundamentos

---

A natureza das organizações atuais é a de lidar com mudanças frequentes. Mudanças de governo, no mercado, nos requisitos, em regulações, leis etc. Esse contexto de mudanças generalizadas e frequentes obviamente se reflete nos negócios das organizações, requerendo assim projetos, operações e empreendimentos flexíveis, adaptáveis, capazes de responder a essas mudanças. Obviamente, tais mudanças chegarão ao nível dos itens de configuração, requerendo assim níveis de controle adequados.

A fim de dispor desses controles, a Gerência de Configuração dispõe do Controle de Mudanças, cujo propósito é o de atuar como “porteiro”, ou seja, estar no comando de todos os pedidos de mudança para um produto, bem como de todas as mudanças implementadas. Para todo e qualquer item de configuração, é preciso que seja possível identificar mudanças em relação aos itens que vieram antes dele. Qualquer mudança precisa ser rastreável, sendo possível ligá-la ao item de configuração que recebeu tal mudança.

**O Controle de Mudanças, portanto, identifica, documenta, aprova ou rejeita e controla mudanças em itens de configuração disponíveis em projetos, operações etc. Em especial, controla as linhas de base. É iniciado por um evento, que nem sempre é formalizado logo em um primeiro momento, podendo ser inicialmente apenas por uma intenção ou desejo por mudança.**

É normal que esse evento seja algo negativo, inesperado, errado, durante o uso do item de configuração, mas pode ser também algo relacionado ao desejo por melhorias gerais. Esses eventos, portanto, podem ser coisas como:

**Um cálculo errado:** Algo percebido durante a revisão de um documento ou relatório.



**Erros em código-fonte:** Encontrados durante a realização de auditorias, revisão por pares, refatoração ou algum outro esforço semelhante.



**Pedidos por melhorias:** Originado do uso corriqueiro do software ou documentação, gerando ideias para melhorias de funcionalidades.



**Erros ou defeitos:** Encontrados durante testes de rotina, testes de integração ou uso comum.



**Desejos por expansão:** Gerados uma vez que um produto já finalizado esteja em operação, requerendo inclusão de novas funcionalidades inteiras.



**Interações com o service desk:** Resultante de problemas no uso do produto ou sistema, geram feedback que pode justificar mudanças.



**Mudanças no ambiente técnico:** Serviços, produtos ou sistemas normalmente não operam sozinhos, e interagem frequentemente com outros sistemas. Uma mudança em infraestrutura, middleware ou outros componentes relacionados pode desencadear a necessidade por mudanças no item de configuração sob controle de mudanças.



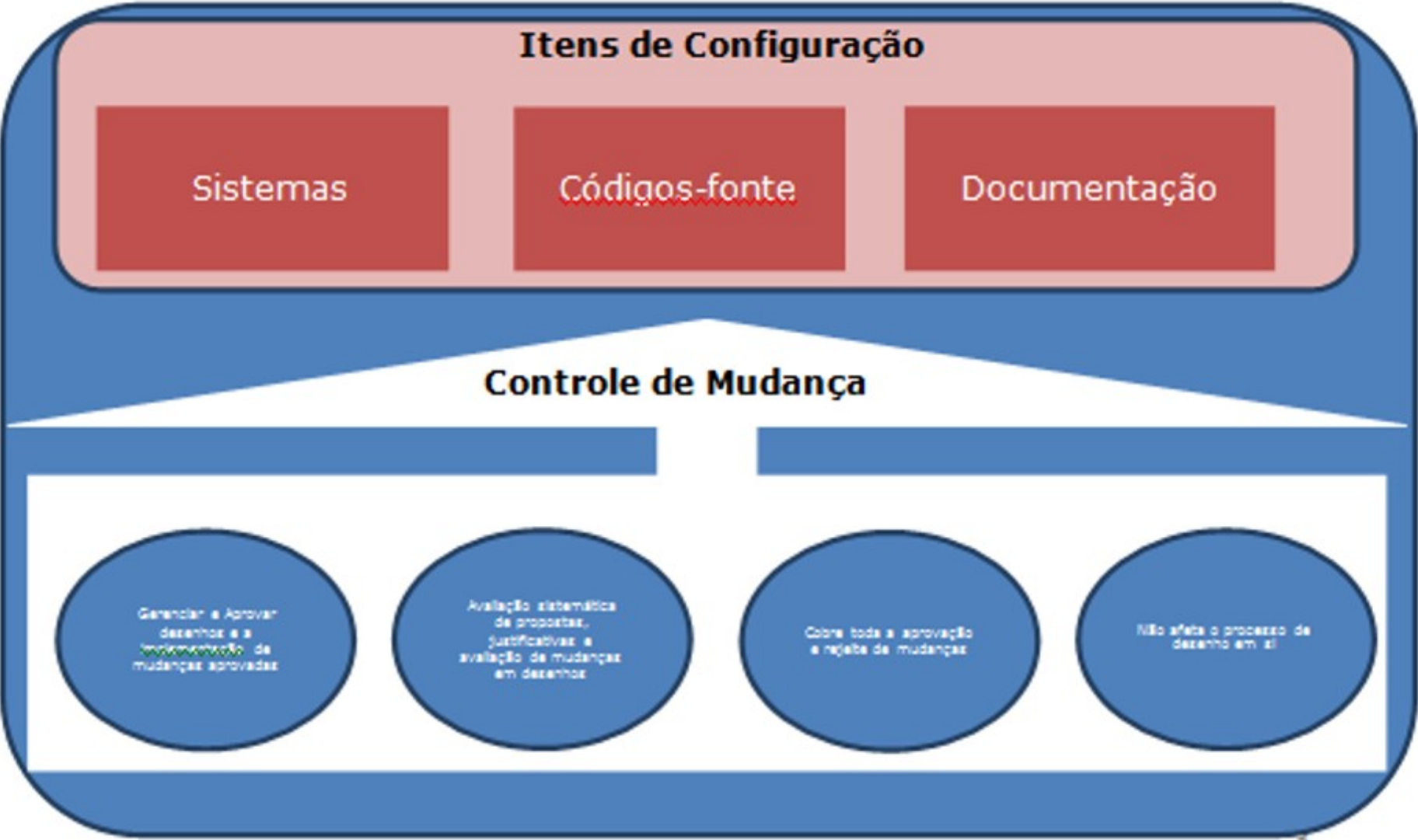


Figura 1: Responsabilidades do Controle de Configuração. Fonte : Autor.

Alguns conceitos fundamentais habilitadores do controle de mudanças são os seguintes:

## Mudança



A adição, modificação ou remoção de qualquer coisa nos itens de configuração. Seu escopo pode incluir arquiteturas, processos, ferramentas, métricas, documentações, alterações em serviços de TI e itens de configuração. Em tese, pode ser gerada em qualquer lugar da organização.

## Solicitação (ou Requisição) de Mudança



Mecanismo para formalização de uma proposta de mudança. É, portanto, uma documentação da mudança, e inclui detalhes sobre ela, podendo ser registrada em papel ou eletronicamente. É preciso ainda que seja seguramente armazenado em um banco de dados de algum tipo, de forma a assegurar a persistência da relação entre solicitações de mudança e os itens de configuração em questão.

## Linha de Base (*baseline*)



Versão considerada estável, que foi acordada formalmente para um item de configuração. É uma referência no tempo para um item de configuração, descrevendo seu estado naquele momento. Uma nova linha de base só pode ser estabelecida por meio do fluxo adotado para o Controle de Mudanças.

## Comitê de (Avaliação de) Mudanças



Ocasionalmente representado por uma única pessoa, é mais comum que seja representado por um grupo de pessoas. Suporta a autorização de mudanças e auxilia a gestão de mudanças na avaliação e priorização dessas mudanças. Algumas pessoas que tendem a fazer parte do comitê são gerentes de projeto, representantes do cliente, responsáveis por garantia da qualidade etc. Também conhecido como Comitê Consultivo de Mudança ou Comitê de Controle de Mudanças.

Fica clara aqui a conexão íntima existente entre o Controle de Mudança e o Controle de Versão, já que, por dedução, fica claro que, ao se ter mudanças aprovadas e implementadas, é inevitável que novas versões de itens de configuração vão também ser geradas.

Aliás, é comum que tanto o Controle de Mudança e o Controle de Versão (bem como outras atividades da Gerência de Configuração) sejam todos implementados e providos por meio de uma única ferramenta ou sistema integrado.

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

## Tipos de Mudança

Como já deve estar claro, as mudanças têm uma importância e peso muito relevantes durante o ciclo de vida dos itens de configuração, e, por consequência, para a Gestão da Configuração como um todo.

Também deve estar claro que se deve esperar um volume de mudanças considerado alto durante o ciclo de vida de um item de configuração, pois não só estará compreendido o momento de projeto, mas também o tempo de operação destes itens.

Tomando por base essas constatações, pode-se deduzir que é do interesse das organizações a execução de uma tratativa diferenciada de mudanças, em outras palavras, não adotar uma abordagem “*one size fits all*” (um-tamanho-se-adequa-a-todos).

Por mais que o Controle de Versão purista não preveja explicitamente tipos diferenciados de mudança, frameworks conhecidos no mercado o fazem. Um deles é a *Information Technology Infrastructure Library* (ITIL).

A ITIL, por meio do processo de Gestão de Mudanças, prevê três tipos de mudança, discriminadas a seguir.

## Mudança-padrão

São mudanças do tipo pré-autorizadas, pois normalmente são repetitivas, de baixo-risco e já são bem-testadas, tendo sido executadas inúmeras vezes na organização. Geralmente são empregadas para mudanças originadas de manutenções rotineiras ou corriqueiras em itens de configuração, sistemas, serviços etc, ou seja, situações que são consideradas “bem-entendidas”.

Fica claro que os riscos advindos de mudanças-padrão são baixos, não justificando, portanto, o investimento considerável de recursos financeiros e de tempo representados pelo envolvimento de vários profissionais de diferentes perfis para a avaliação, documentação e implementação dessas mudanças.

## Comentário

Tais mudanças normalmente são implementadas por meio de procedimentos conhecidos, repetíveis, muitas vezes roteirizados via algum tipo de checklist.

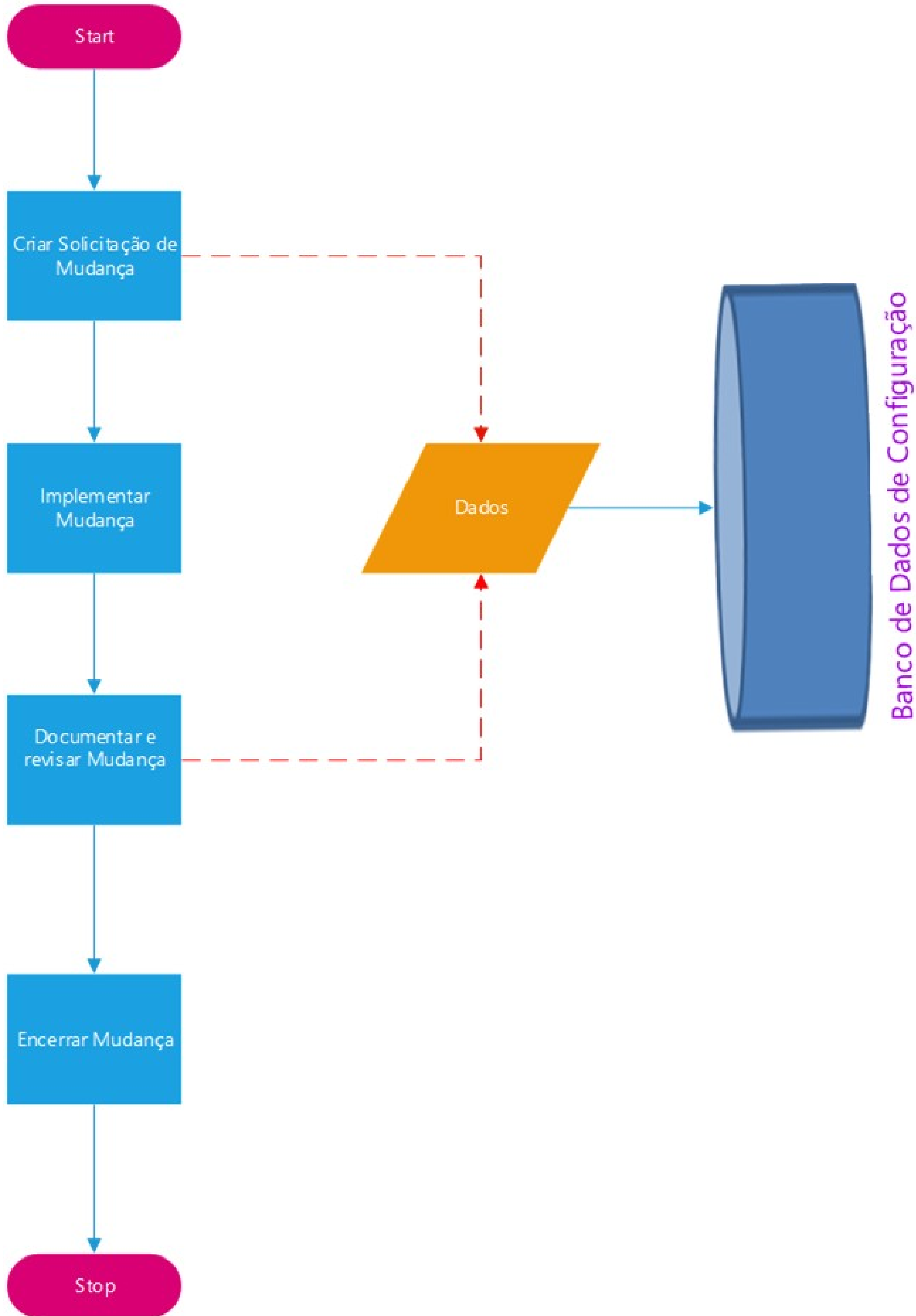


Figura 2: Possível Fluxo Simplificado para Mudanças Padrão. Fonte: Autor.

Mudança normal

Como o próprio nome diz, é uma mudança comum, que deve receber o tratamento e supervisão esperados para as mudanças. Essa é inclusive a variante de mudança que, caso o Controle de Mudanças esteja bem definido e maduro, deverá gerar o maior volume de solicitações de mudança, passando por todo o ciclo do Controle de Mudanças.

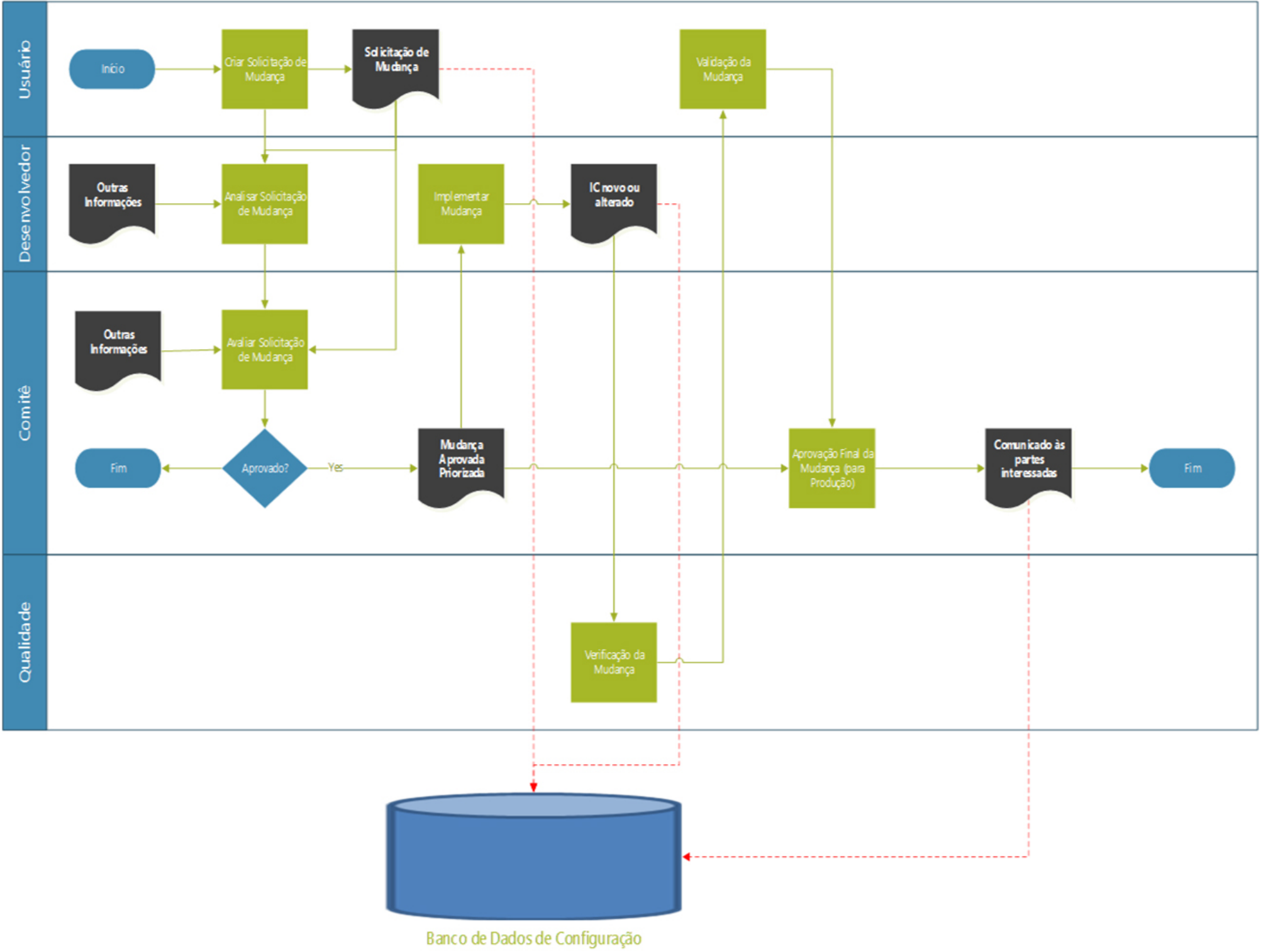


Figura 3: Possível Fluxo para Mudança Normal. Fonte: Adaptado de Configuration Management Principles and Practice

Isso implica dizer que serão analisadas e avaliadas de forma criteriosa, em busca de se entender o mérito da mudança, os vários riscos envolvidos, priorizando-a e implementando-a adequadamente, e possivelmente executando-se a verificação (testes de qualidade focados em defeitos) e a validação (normalmente testes dos usuários para assegurar conformidade com requisitos).

Os riscos destas mudanças normalmente são medianos, e o ciclo do Controle de Mudança ajuda a mitigá-los.

São exemplos de mudanças normais a migração de recursos críticos de informação, migração de sistemas de um ambiente não-operacional para o ambiente operacional, execução de scripts em bancos-de-dados, a liberação de uma nova versão de relatório, entre outros. Mudanças normais geralmente se encontram entre mudanças padrão e mudanças emergenciais em termos de complexidade e riscos.

### Mudança Emergencial

No ramo da Tecnologia da Informação, situações inesperadas que causam grande impacto ao negócio tendem a ocorrer com certa frequência.



São exemplos dessas situações falhas em um item de configuração, sistema ou serviço, que como consequência acabam por conduzir à indisponibilidade parcial ou mesmo total de uma ou mais faces dos negócios (operações, por exemplo), gerando prejuízos materiais, financeiros, de credibilidade e outros.

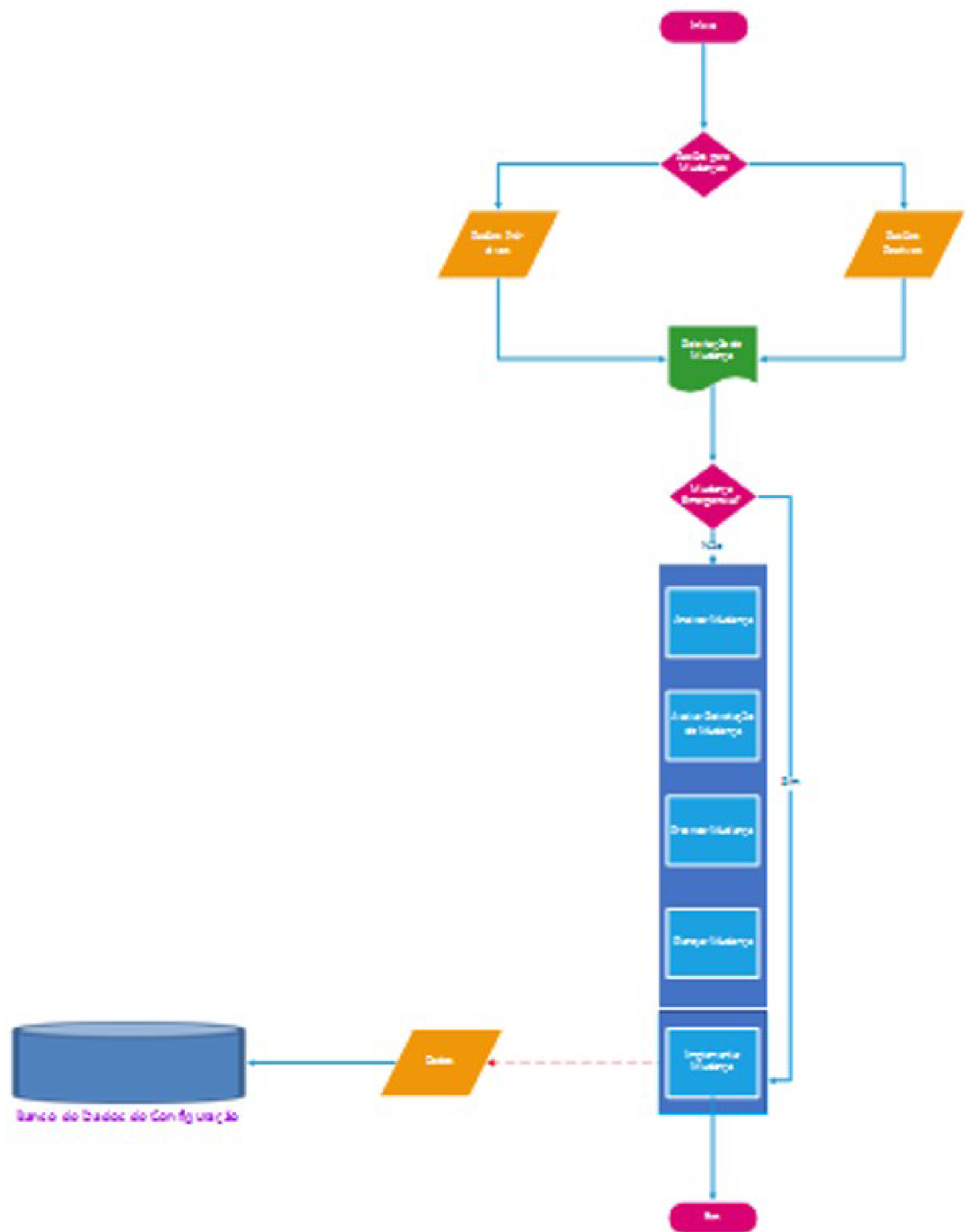
Também precisamos levar em consideração situações em que, para evitar potenciais disrupções como as previamente mencionadas, haja necessidade de uma ação preventiva, como por exemplo, aplicar um patch para uma vulnerabilidade grave conhecida, ou executar a reconfiguração de uma parte ou de toda a rede corporativa diante da ameaça de um iminente ataque malicioso etc. Quando isso ocorre, um tipo especial de mudança é requerido: A [mudança emergencial](#)<sup>1</sup> .

Normalmente, as mudanças emergenciais irão dispensar qualquer esforço de documentação inicial, indo diretamente para implementação. Dispensarão, muito provavelmente ainda, avaliação pelo Comitê de Mudanças, passando ao invés disso por um processo de aprovação rápido, que normalmente envolve perfis da alta administração, tais como gerentes sênior, diretores, vice-presidentes e outros, garantindo assim a agilidade necessária, ao mesmo tempo em que se mantém um mínimo de supervisão.

## Atenção

Essas aprovações são, portanto, incômodas de serem obtidas, pois acarretam perguntas acerca da mudança, constituindo ainda mecanismo que desencoraje a tratativa de qualquer mudança como sendo emergencial.

Por vezes, uma variação do Comitê de Mudanças, conhecido como Comitê de Mudanças Emergenciais, pode ser estabelecida para lidar com Mudanças Emergenciais. Como explanado, outra variação comum é para as categorias das mudanças, podendo ser classificadas de acordo com quem iniciou as mudanças, por exemplo: Mudanças (Vindas) de Usuário, Mudanças (Vindas) de Fornecedores, Mudanças de TI etc.



📷 Fonte: O autor.

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

# Boas Práticas no Controle de Mudança

Um bom Controle de Mudança é crítico para uma Gerência de Configuração eficaz, mas também eficiente, pois ajuda a evitar impactos negativos sobre os projetos e operações, tais como atrasos, erros, retrabalho e outros. Existe obviamente uma série de boas práticas que podem ser adotadas para que se obtenha esse controle de mudanças maduro, entre elas:


- **Usar ativamente os metadados disponíveis:** Quando se lida com mudanças, metadados podem e devem ser usados para propósitos analíticos. Isso pode se dar por meio de relatórios ou por busca direta nos bancos de dados que armazenam os metadados.

Informações disponíveis e que conduzem à rastreabilidade são usadas, por exemplo, para determinar em qual ou quais itens de configurações é necessário implementar as mudanças.

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.

- **Realizar análise de impacto:** Ao se analisar e avaliar requisições de mudança, é preciso considerar os impactos de tal mudança, como, por exemplo, os custos associados à implementação e à não-implementação. Embora isso pareça lógico, nem sempre é algo fácil de fazer. Um bom começo é adotar algo como um *checklist*.

 Análise de impactos

 Clique no botão acima.

- Identificar:
  - Todos os requisitos afetados ou em conflito com mudança proposta;
  - Consequências caso a mudança proposta não seja implementada;
  - Possíveis efeitos adversos e outros riscos ligados à mudança;
  - Quanto do valor já investido no produto ou item de configuração, alvos da mudança, será perdido se a mudança for implementada — ou se não for.
- Checar se a mudança proposta:
  - Tem algum efeito em requisitos não-funcionais, tais como resiliência, performance, disponibilidade etc.;
  - Pode ser introduzida com as tecnologias conhecidas e recursos que se encontram disponíveis;
  - Irá requerer níveis de recursos inaceitáveis para desenvolvimento e testes;
  - Irá gerar um preço melhor ou mais alto;
  - Irá afetar marketing, produção, serviços ou suporte.
- Checar se a mudança proposta acarreta adições, mudanças, ou remoções em:
  - Interfaces internas ou externas, relatórios, ou armazenamento de dados;
  - Desenhos, Código-fonte, *scripts*;
  - Planos e especificações de teste;
  - Textos de ajuda, manuais de usuários, material de treinamento;
  - Planos de Projeto e planos componentes (qualidade, cronograma, custos, outros);
  - Outros sistemas, aplicações, bibliotecas ou mesmo *hardware*.

## Ferramentas

---

### Redmine

A ferramenta Redmine é uma ferramenta opensource voltada para a Gestão de Projetos profissional.

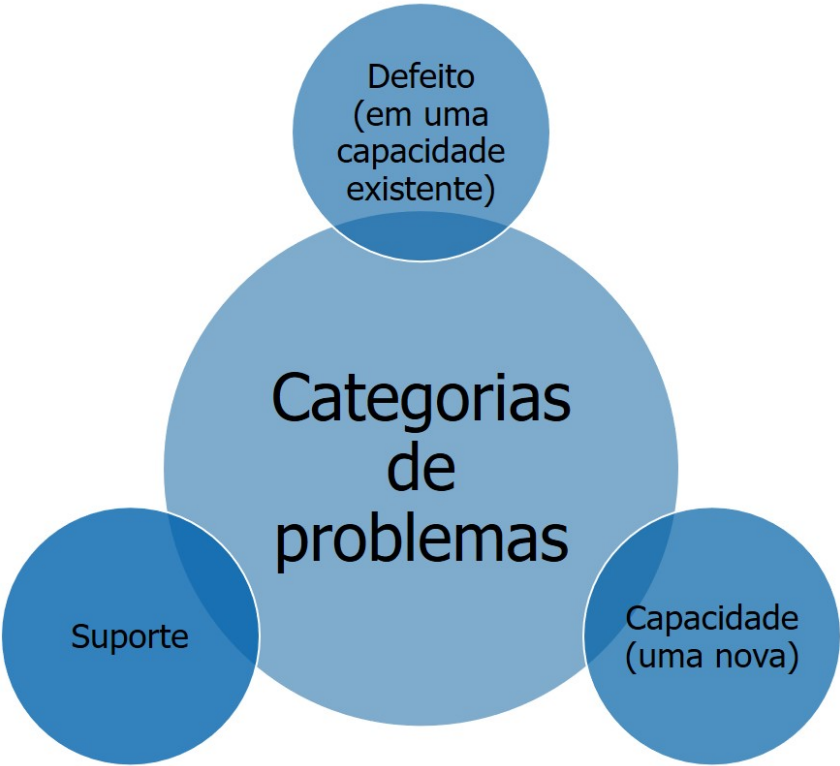
Tem em seu núcleo a proposta de rastreamento de aspectos relacionados aos projetos, permitindo a colaboração de times inteiros com relação à execução e progresso de tarefas, defeitos, capacidades e outros itens, além de proporcionar gráficos de Gantt, permitir a criação de wikis de projetos etc. É ainda agnóstica o bastante para se acoplar a vários sistemas de gerenciamento de banco de dados, tais como MySQL e SQL Server.

A Redmine é capaz de se integrar com vários sistemas de Controle de Versão por meio de um módulo especial, suportando várias ferramentas conhecidas, tais como o Subversion e o Git. No entanto, é na capacidade de rastreamento de problemas (*issues*) que a ferramenta se destaca, pois tal capacidade pode ser, até certo ponto, utilizada como ferramenta de Controle de Mudança.

### Comentário

Aliás, é comum que usuários se perguntem se a Redmine é mesmo uma ferramenta de Gestão de Projetos, ou se na verdade é uma ferramenta de rastreio de problemas. Como é de se esperar, vários campos que geram metadados existem, tais como prioridade, datas, tempo estimado, percentual completo etc.

Os problemas (*issues*) podem ser categorizados da seguinte forma:



Fonte: O autor.

É possível inclusive relacionar problemas a outros problemas existentes. Seja como for, o Redmine disponibiliza mecanismos para rastrear as mudanças<sup>2</sup> nos status destes problemas.

Novas capacidades estão disponíveis e também estão sendo constantemente desenvolvidas e adicionadas ao Redmine via plugins. Se tal capacidade de expansão é vantajosa, por outro lado também pode ser considerada como uma fraqueza do Redmine, pois a ferramenta não é excepcional em nada do que faz, sendo mediana se comparada a ferramentas mais especializadas em um ou alguns poucos aspectos.

Outra reclamação constante é a interface desatualizada, além de proporcionar uma experiência um tanto confusa aos usuários.

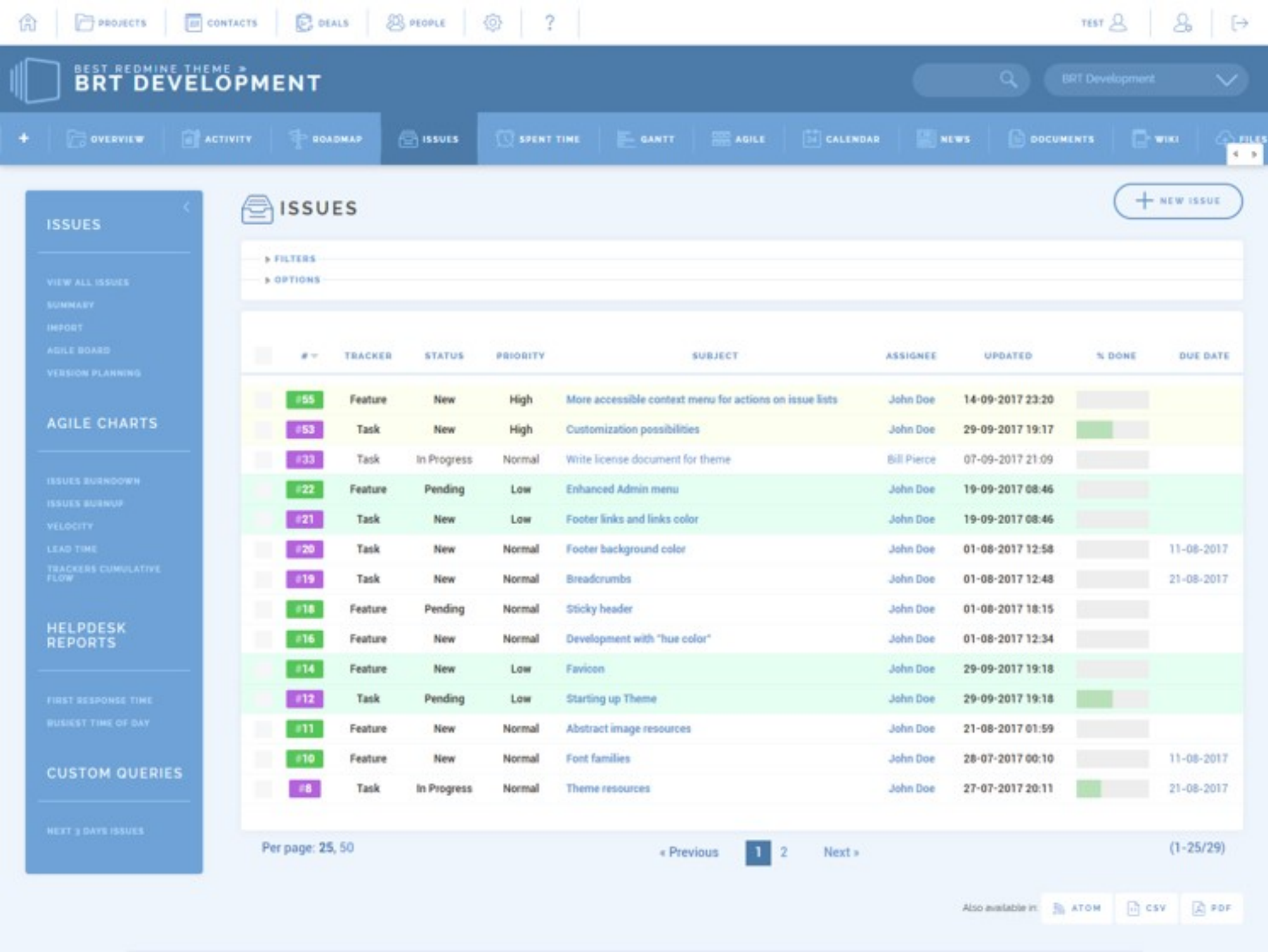


Figura 5: Visão da área de problemas no Redmine. Fonte: Redmine website.

# Gitlab

O Gitlab é primariamente uma ferramenta para controle de repositórios Git. Para isso, é constituído como uma camada construída sobre o Git, e similarmente ao Redmine, suporta desenvolvimento de software como um todo, destacando-se suas capacidades avançadas de controle de versão. O Gitlab pode inclusive ser uma alternativa completa a ferramentas de controle de versão, como o SVN, por exemplo.



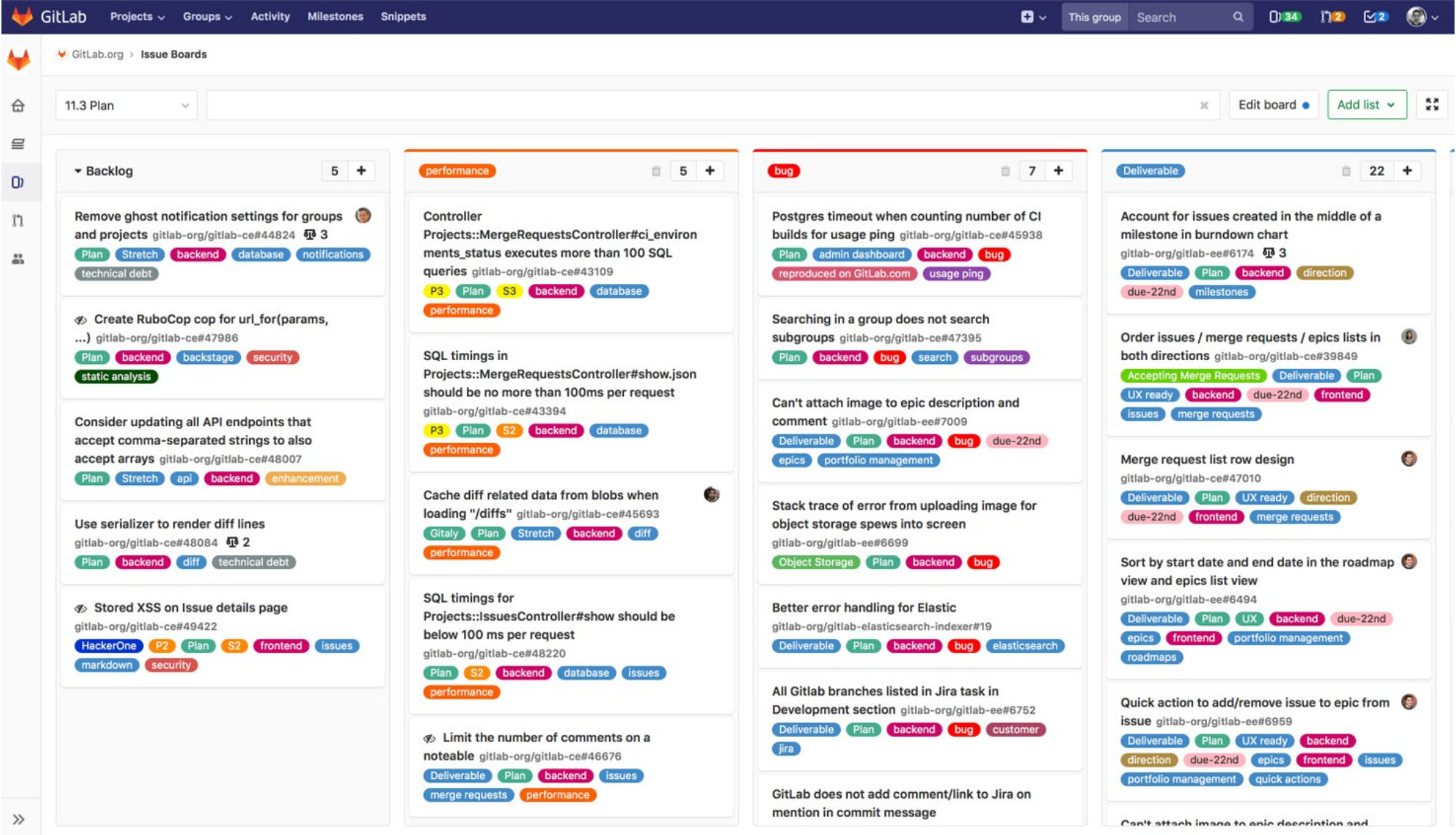


 Figura 6: Fotografias dos dados armazenados pelo Git ao longo do tempo. Fonte: Gitlab Website

Já que suporta o ciclo de desenvolvimento completo, o Gitlab fornece os mecanismos e controles para que, similarmente ao Redmine, exerça o Controle de Mudança de maneira adaptada.

**01** É interessante notar que, diferentemente do Redmine, existem vários níveis para membros, incluindo um que não custa nenhum valor mensal, até outros que custam até uma centena de dólares mensais.

**2** Similarmente ao Redmine, reclamações com relação à interface e experiência navegacional dos usuários são relativamente comuns, com o adendo de que muitos usuários tendem a reportar bugs nas funcionalidades disponibilizadas.

**3** Outro ponto que merece destaque são os reportes de suporte à comunidade não tão adequados, o que pode ser um fator importante a ser considerado antes da adoção dessa ferramenta.

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

# Atividade

---

1. O \_\_\_\_\_ pode ser usado como ferramenta de Controle de Mudança.

- a) Git.
  - b) Gitlab.
  - c) Jenkins.
  - d) SVN.
  - e) CVS.
- 

2. \_\_\_\_\_ é uma formalização para mudança:

- a) Comitê de Mudança.
  - b) Versão.
  - c) Linha de Base.
  - d) Solicitação de Mudança.
  - e) Mudança padrão.
- 

3. Mudança do tipo \_\_\_\_\_ normalmente implica em pré-aprovação. Estamos falando de:

- a) Padrão.
  - b) Normal.
  - c) Emergencial.
  - d) Urgente.
  - e) Solicitação.
- 

4. Não é uma boa prática de Controle de Mudança:

- a) Descartar metadados.
  - b) Checar recursos necessários.
  - c) Entender mudanças adicionais.
  - d) Entender a consequência de não implementar.
  - e) Realizar análise de impacto.
-



5. O Comitê de Mudanças precisa:

- a) Sempre ser envolvido em qualquer mudança.
  - b) Ser sempre envolvido em mudanças do tipo emergencial.
  - c) Nunca ser envolvido em mudanças.
  - d) Ser sempre envolvido em mudanças normais.
  - e) Ser sempre envolvido em mudanças padrão.
- 

## 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.

### Rastrear mudanças<sup>2</sup>

---

Esse rastreamento é bidirecional, ou seja, pode informar mudanças a todos os envolvidos quando o usuário altera o problema, ou quando o time de desenvolvimento o atualiza, de acordo com o progresso do trabalho.

## 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.

EVERTSE, J. **Mastering Gitlab 12**: Implement DevOps culture and repository management solutions. 1.ed. PACKT Publishing, 2019.

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

LESYUK, A. **Mastering Redmine**. 2.ed. PACKT Publishing, 2016.

## Próxima aula

---

- Conceituação e tipos de liberação;
- Boas práticas da Gestão de Liberação;
- Software / Ferramentas de Gerenciamento de Liberação.

Explore mais

---

Pesquise na internet sites, vídeos e artigos relacionados ao conteúdo visto.

Em caso de dúvidas, converse com seu professor online por meio dos recursos disponíveis no ambiente de aprendizagem.