

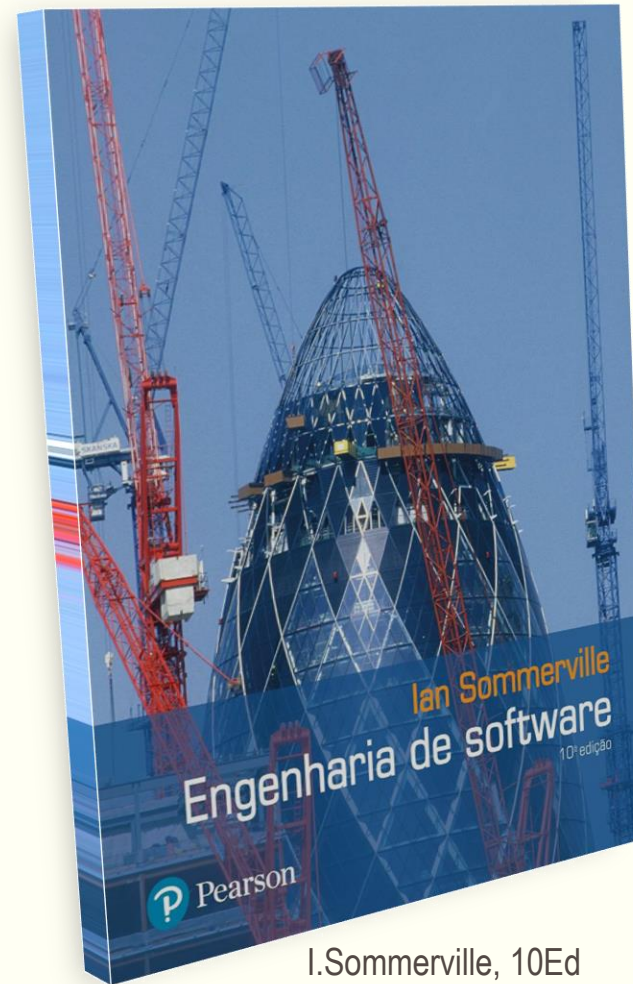
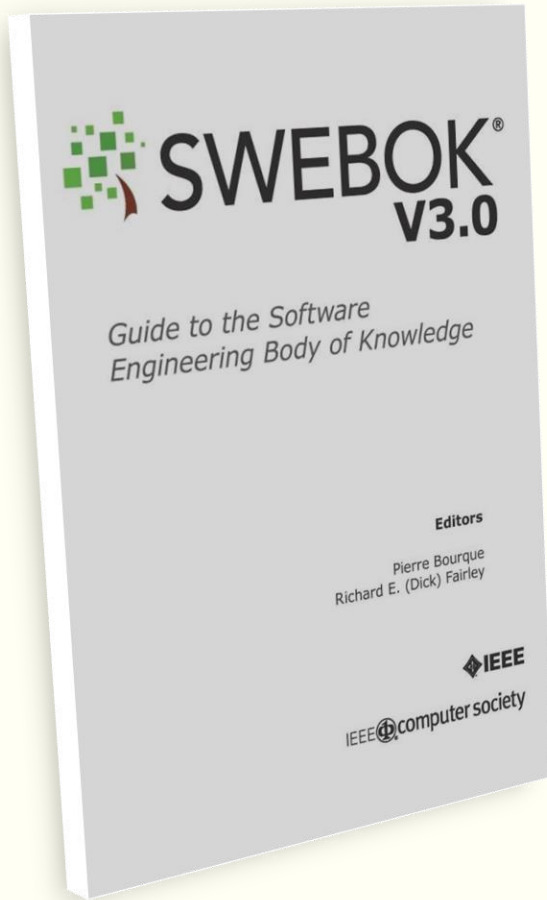
8

GERENCIAMENTO DA CONFIGURAÇÃO DE SOFTWARE

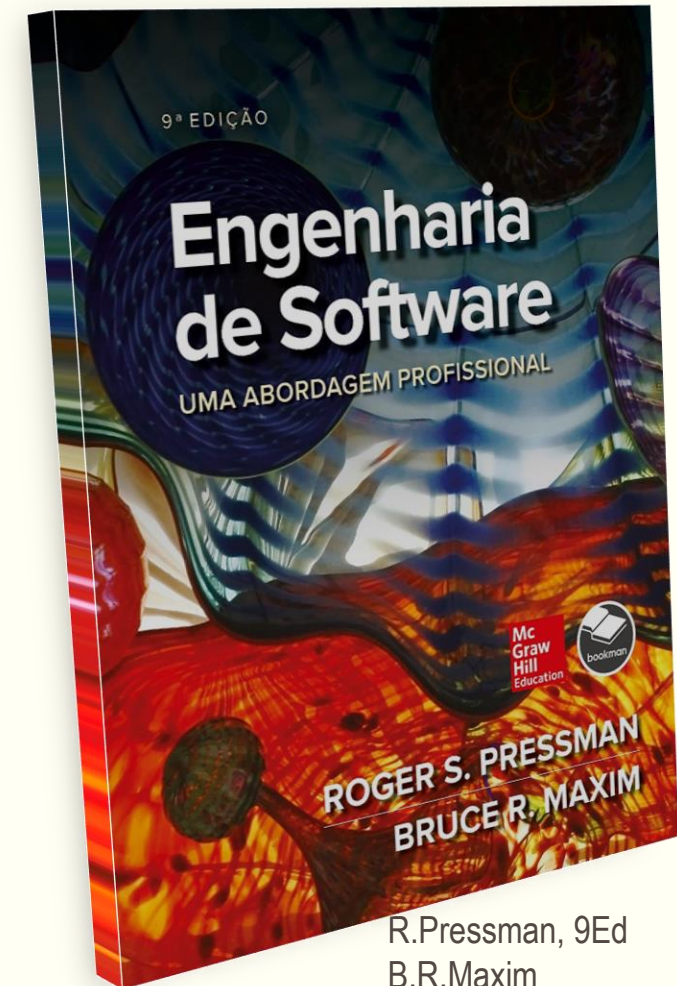
Prof. Ausberto S. Castro Vera
UENF – CCT – LCMAT
Ciência da Computação



Bibliografia Básica

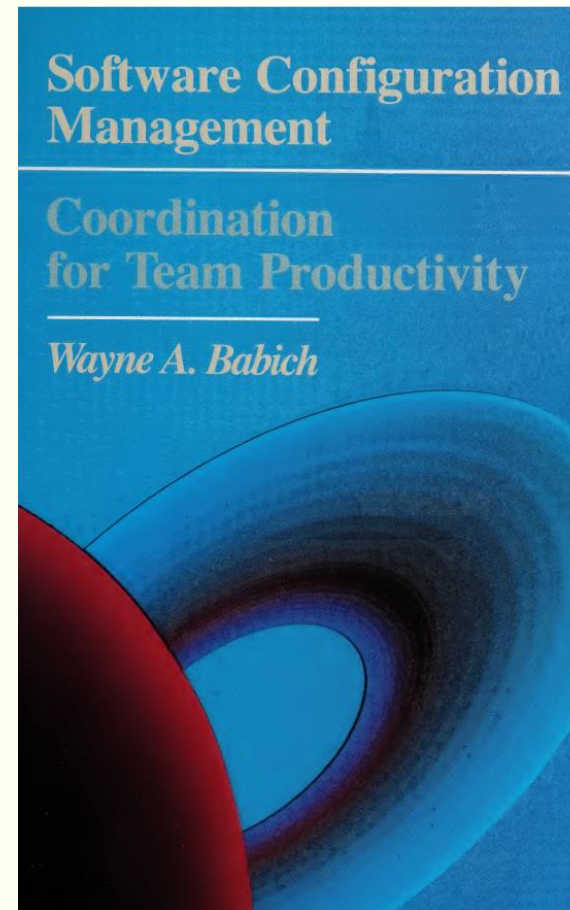
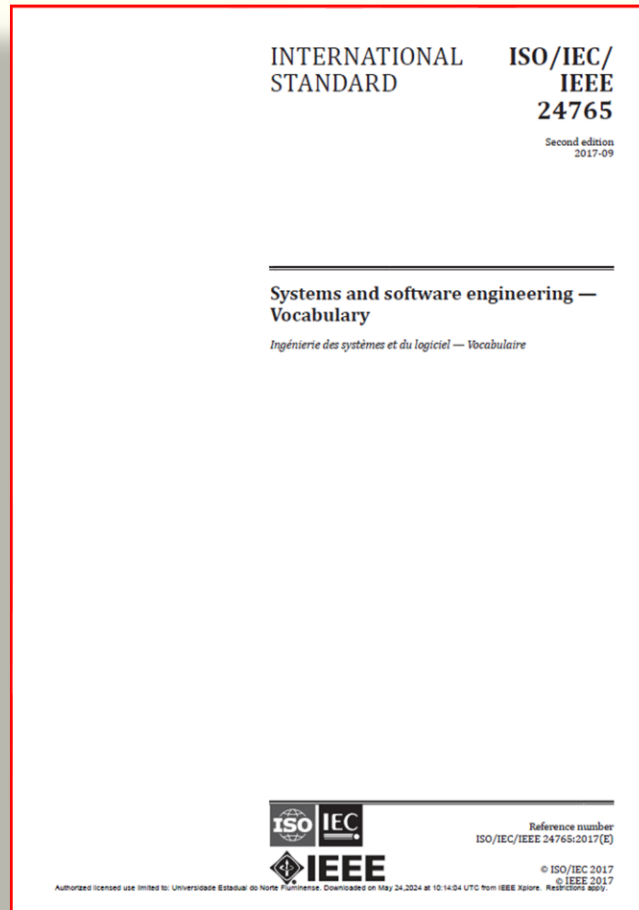


I.Sommerville, 10Ed
Pearson
2018
Cap 25



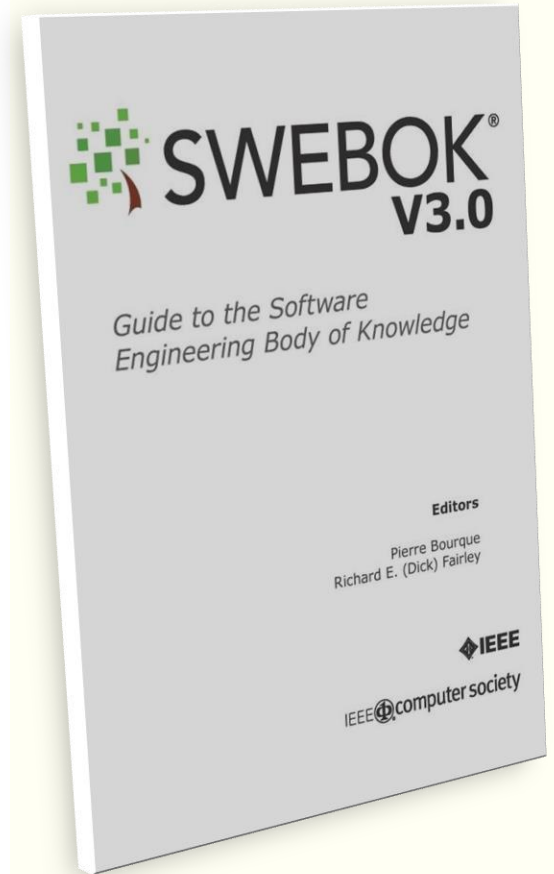
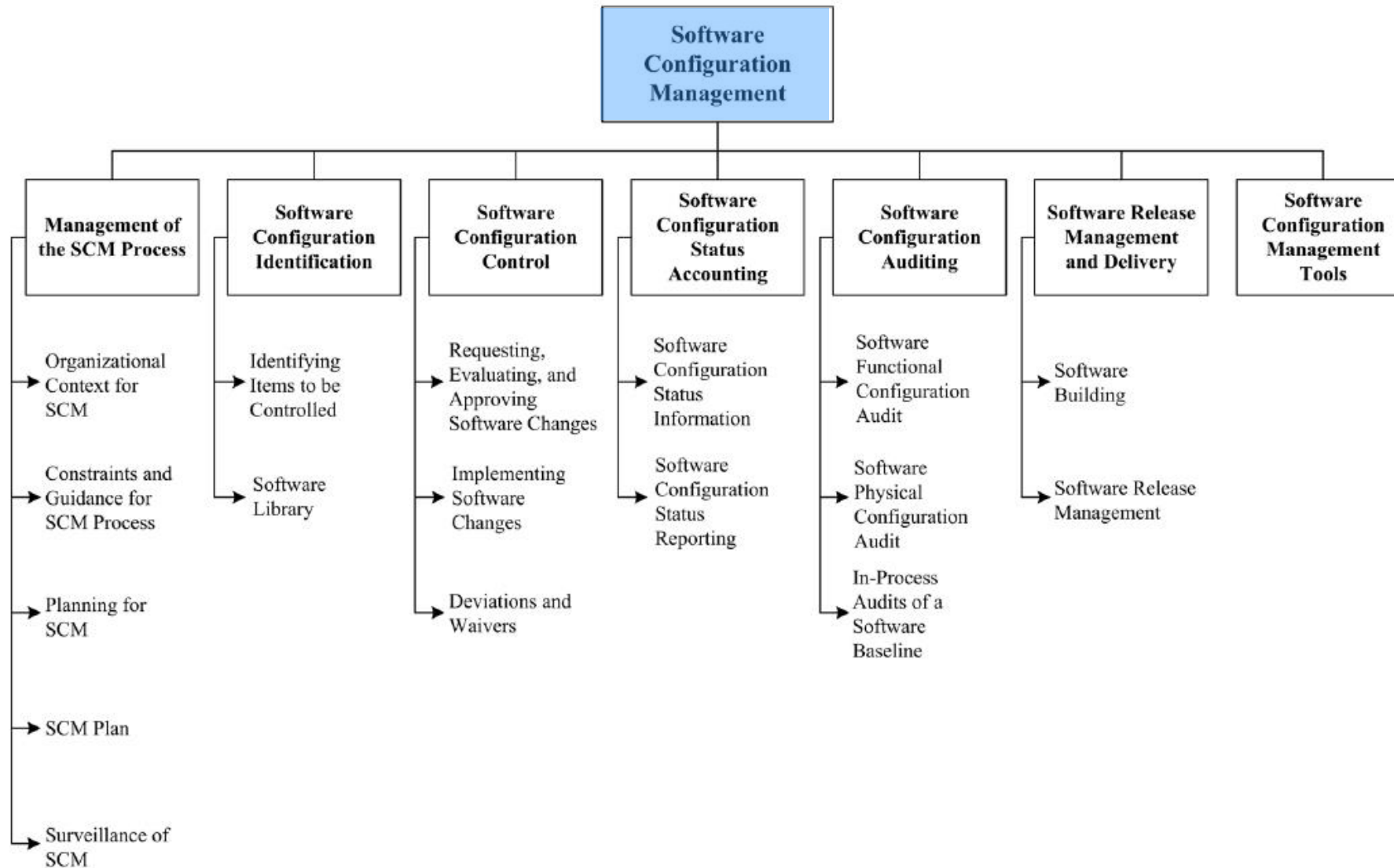
R.Pressman, 9Ed
B.R.Maxim
AMGH Editora
2021
Cap. 22

Bibliografia Complementar



24765-2017 - ISO/IEC/IEEE International Standard -
Systems and software engineering--Vocabulary

Software Configuration Management



8: Software Configuration Management (SCM)

- **1. Management of the SCM Process**

- **1.1. Definitions and Terminology**
- **1.2. Nature of Maintenance**
- **1.3. Need for Maintenance**
- **1.4. Majority of Maintenance Costs**
- **1.5. Evolution of Software**
- **1.6. Categories of Maintenance**

- **2. Software Configuration Identification**

- **2.1. Identifying Items to be Controlled**
- **2.2. Software Library**

- **3. Software Configuration Control**

- **3.1. Requesting, Evaluating, and Approving Software Changes**
- **3.2. Implementing Software Changes**
- **3.3. Deviations and Waivers**

- **4.**

- **4. Techniques for Maintenance**

- **4.1. Program Comprehension**
- **4.2. Reengineering**
- **4.3. Reverse Engineering**
- **4.4. Migration**
- **4.5. Retirement**

- **5. Software Maintenance Tools**

Sistema Computacional

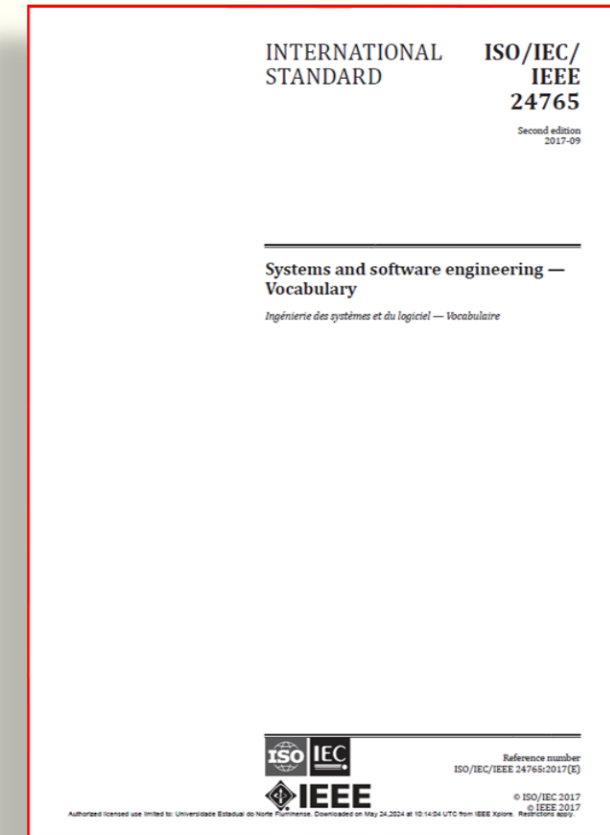
Um **sistema** pode ser definido como a combinação de elementos interativos organizados para atingir um ou mais propósitos declarados.

24765-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering--Vocabulary



Configuração

1. **Arranjo** de um sistema ou componente de computador, conforme definido pelo número, natureza e interconexões de suas partes constituintes
2. No gerenciamento de configuração, as **características funcionais e físicas** de hardware ou software conforme estabelecidas na documentação técnica ou obtidas em um produto
3. **Arranjo** de um sistema ou rede conforme definido pela natureza, número e características principais de suas unidades funcionais
4. **Requisitos, design e implementação** que definem uma versão específica de um sistema ou componente de sistema
5. A **maneira** pela qual o hardware e o software de um sistema de processamento de informações são organizados e interconectados
6. **Coleção** de objetos capazes de interagir em interfaces



Configuração de um sistema

- A **configuração de um sistema** são as *características funcionais e físicas de hardware ou software* conforme estabelecidas na *documentação técnica* ou obtidas em um produto.
- Uma **configuração de um sistema** é uma **coleção de versões** específicas de itens de hardware, firmware ou software combinados de acordo com *procedimentos de construção* específicos para servir a um *propósito específico*

24765-2017 - ISO/IEC/IEEE International Standard -
Systems and software engineering--Vocabulary



Item de Configuração - 24765-2017 - ISO/IEC/IEEE

item de configuração (CI, Configuration Item)

1. item ou agregação de hardware, software ou ambos designado para gerenciamento de configuração e tratado como uma entidade única no processo de gerenciamento de configuração
2. componente de uma infraestrutura ou item que está ou estará sob controle do gerenciamento de configuração
3. agregação de produtos de trabalho designados para gerenciamento de configuração e tratados como uma entidade única no processo de gerenciamento de configuração
4. entidade dentro de uma configuração que satisfaz uma função de uso final e que pode ser identificada exclusivamente em um determinado ponto de referência

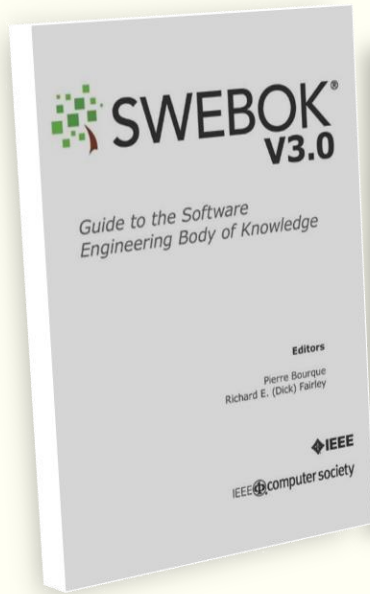
Gerenciamento da Configuração de Software

- **item de configuração de software** (SCI, Software Configuration Item)
entidade de software que foi estabelecida como item de configuração
Ex. componente de software de computador, item de configuração de software de computador, item de configuração de hardware, item de software
- **Gerenciamento de Configuração de Software** (SCM, Software Configuration Management)
processo de aplicação do gerenciamento de configuração ao longo do ciclo de vida do software para garantir a integridade e correção dos SCIs

24765-2017 - ISO/IEC/IEEE International Standard -
Systems and software engineering--Vocabulary

Gerenciamento da Configuração

O **gerenciamento de configuração** (CM, Configuration Management), é a disciplina de *identificação da configuração de um sistema* em momentos distintos com o propósito de *controlar sistematicamente as alterações* na configuração e *manter a integridade e rastreabilidade* da configuração ao longo do ciclo de vida do sistema.



É uma disciplina que aplica *direção e supervisão técnica e administrativa* para: *identificar e documentar* as características funcionais e físicas de um *item de configuração*, controlar alterações nessas características, registrar e relatar o status de processamento e implementação de alterações e verificar a conformidade com requisitos especificados

Gerenciamento da Configuração

1. **disciplina** que aplica direção e vigilância técnica e administrativa para: identificar e documentar as características funcionais e físicas de um item de configuração, controlar alterações nessas características, registrar e relatar o processamento de alterações e o status de implementação e verificar a conformidade com requisitos especificados
2. **atividades técnicas e organizacionais**, compreendendo identificação de configuração, controle, contabilidade de status e auditoria
3. **atividades coordenadas** para dirigir e controlar a configuração

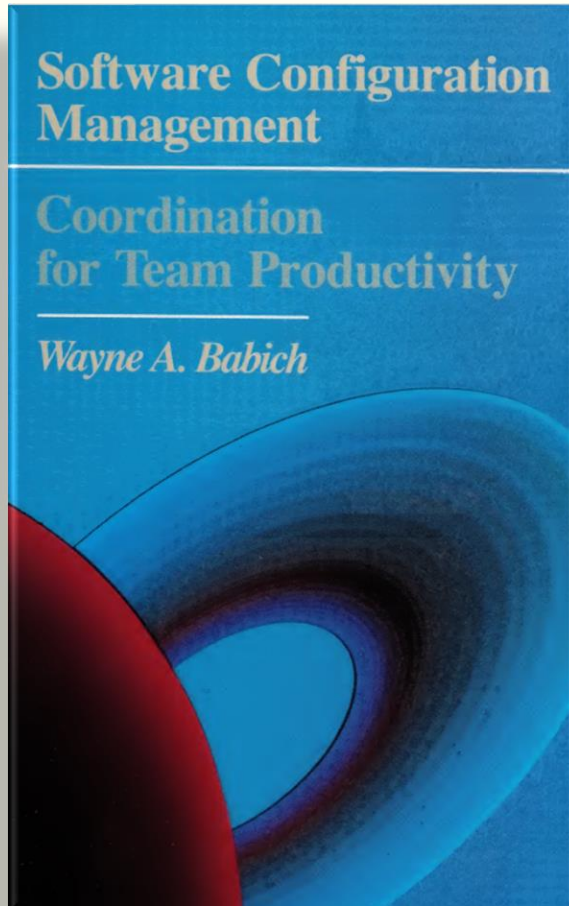
24765-2017 - ISO/IEC/IEEE International Standard -
Systems and software engineering--Vocabulary

Gerenciamento da Configuração

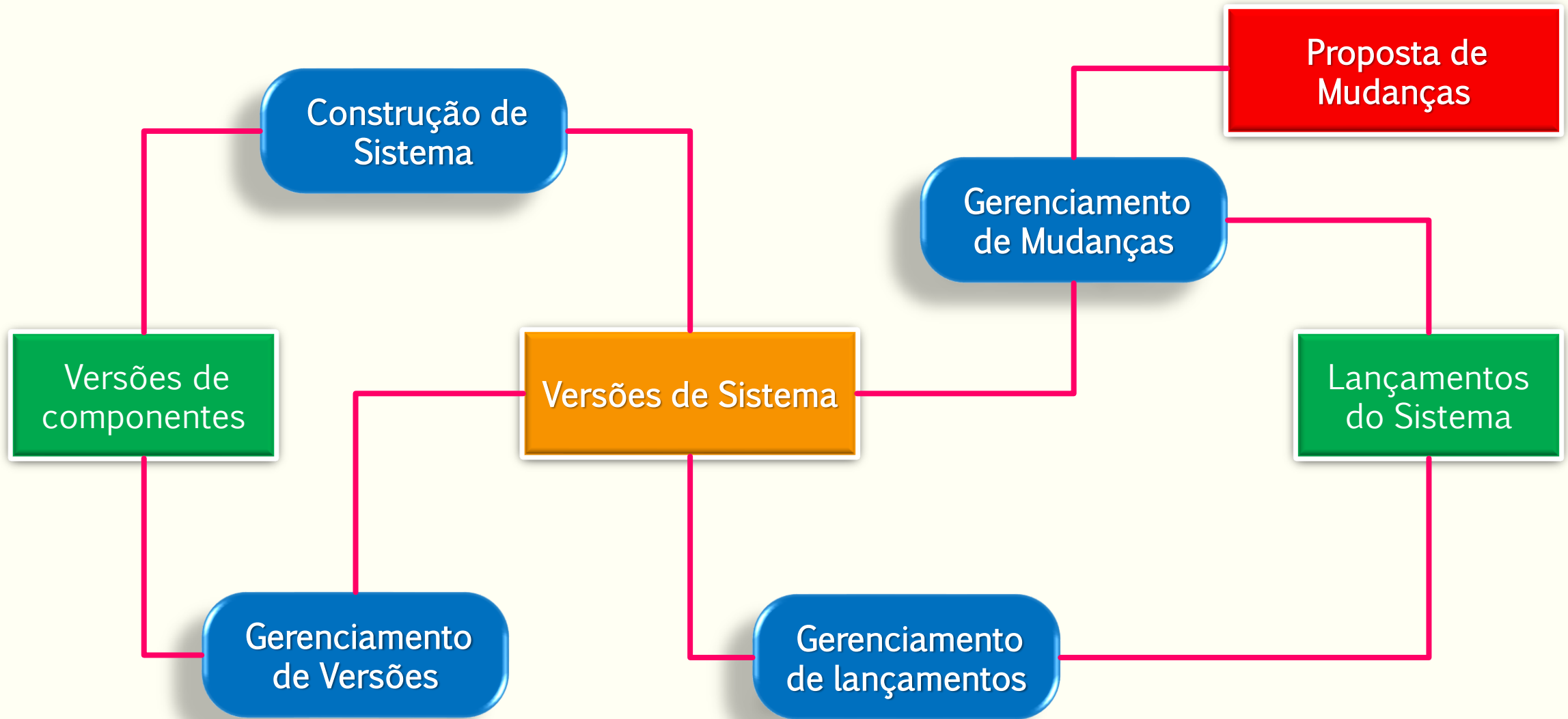
A **gestão de configuração**

é a arte de *identificar, organizar e controlar* modificações no *software* que está em construção por uma equipe de programação. O objetivo é maximizar a produtividade minimizando os erros

Wayne A. Babich, 1986



Atividades de Gerenciamento de Configuração



Atividades de Gerenciamento de Configuração

1. Controle de versão

Envolve manter o controle das várias versões dos componentes do sistema e garantir que as mudanças feitas em componentes por diferentes desenvolvedores não interfiram umas com as outras.

2. Construção de sistema

Processo de reunir componentes, dados e bibliotecas do programa, compilando-os eligando-os para criar um sistema executável.

3. Gerenciamento de Mudanças

Envolve manter o controle das solicitações de mudanças de clientes e desenvolvedores no software já entregue, elaborar os custos e o impacto de fazer essas mudanças e decidir se e quando as alterações devem ser implementadas.

4. Gerenciamento de lançamentos (releases)

Envolve a preparação de software para o lançamento externo e o acompanhamento das versões de sistema que foram lançadas para uso do cliente.



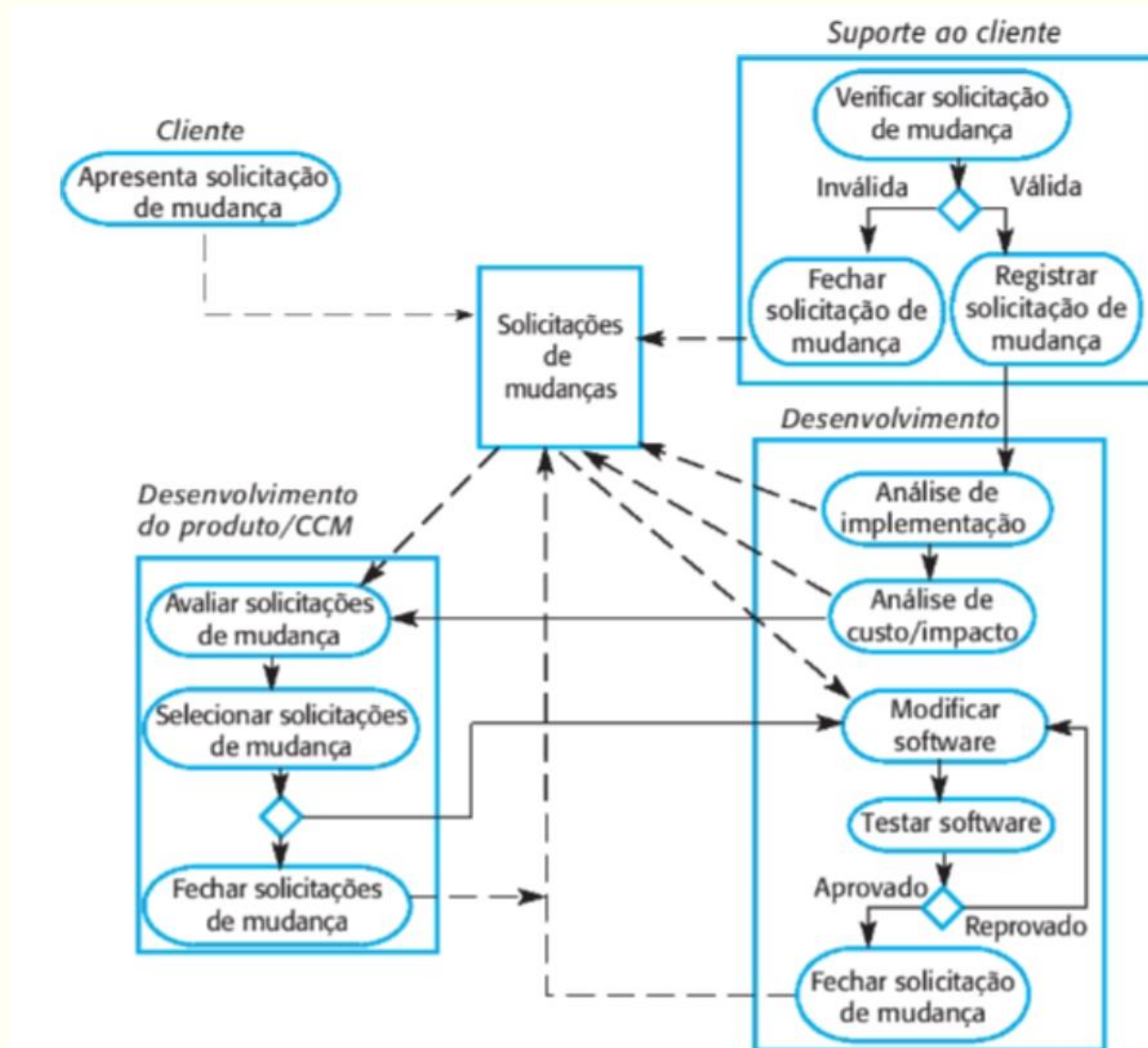
Palavra-chave em GCS



Gerenciamento de Mudanças

- A *mudança* é um fato da vida para os grandes sistemas de software.
- As *necessidades organizacionais* (cliente) e os *requisitos* (sistema) mudam durante a *vida útil de um sistema*. Os defeitos precisam ser reparados e os sistemas precisam se adaptar às *mudanças em seu ambiente*.
- Para garantir que as alterações sejam *aplicadas ao sistema de uma maneira controlada*, é necessário um conjunto de processos de *gerenciamento de mudanças* apoiados por ferramentas.
- O **gerenciamento de mudanças** visa a garantir que a evolução do sistema seja controlada e que as alterações mais urgentes e com bom custo-benefício sejam priorizadas.
- O **gerenciamento de mudanças** é o processo de analisar os custos e os benefícios das mudanças propostas, aprovando as que têm bom custo-benefício e controlando quais componentes do sistema foram modificados.
- **Processo de Gerenciamento de Mudanças**
 - Esse processo deve entrar em vigor quando o software for entregue para lançamento aos clientes ou para implantação dentro de uma organização.
 - Muitas variantes desse processo estão em uso, dependendo se o software é um sistema personalizado, uma linha de produtos ou um produto de prateleira.
 - Todos os processos de gerenciamento de mudanças devem incluir *alguma maneira* de verificar, precificar e aprovar alterações.

Gerenciamento de Mudanças



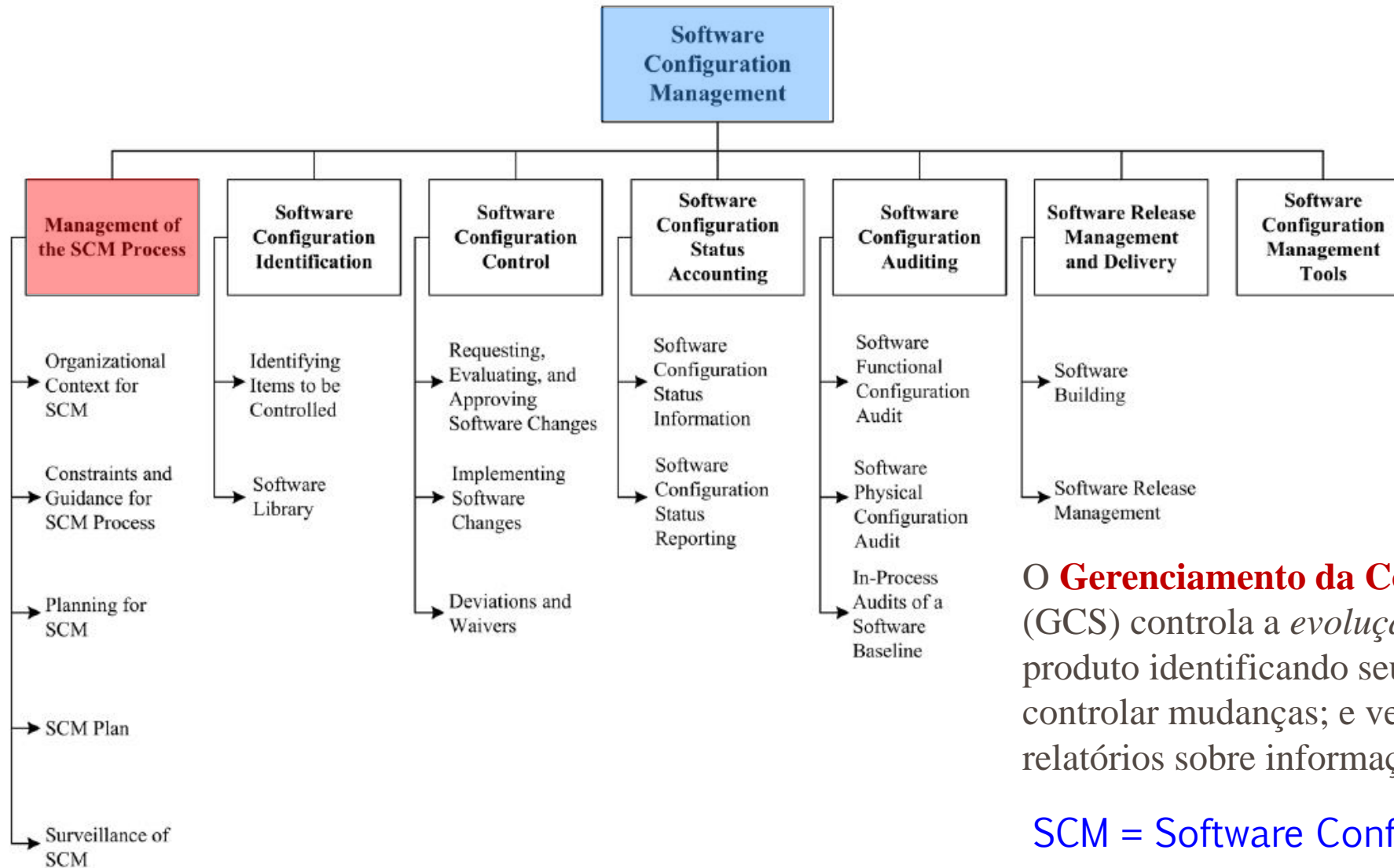
Contexto do Gerenciamento de Configuração de Software

Os conceitos do GCS só se aplicam a
Sistemas de Software

GRANDES



1. Gerenciamento do Processo GCS (SCM)



O **Gerenciamento da Configuração do Software** (GCS) controla a *evolução e integridade* de um produto identificando seus elementos; gerenciar e controlar mudanças; e verificar, registrar e gerar relatórios sobre informações de configuração.

SCM = Software Configuration Management

1. Gerenciamento do Processo GCS

- 1.1. Contexto Organizacional para o GCS
- 1.2. Restrições e Orientações para o Processo GCS
- 1.3. Planejamento para o GCS
 - 1.3.1. Organização e Responsabilidades do GCS
 - 1.3.2. Recursos e Cronogramas GCS
 - 1.3.3. Seleção e implementação de ferramentas
 - 1.3.4. Controle de fornecedor/subcontratado
- 1.4. Plano GCS
- 1.5. Vigilância do GCS (Gerenciamento de Configuração de Software)

GCS = Gerenciamento da Configuração de Software
SCM = Software Configuration Management

1.1. Contexto Organizacional para o GCS



1.1. Contexto Organizacional para SCM

1. Abrangência do GCS:

O GCS se aplica a diversos tipos de artefatos de software, desde o código-fonte, documentações e testes até ambientes de desenvolvimento, homologação e produção. Ele garante a rastreabilidade e o controle de todas essas peças ao longo do tempo.

2. Ciclo de Vida do Software:

O GCS está presente em todas as fases do ciclo de vida do software, desde o planejamento inicial até a desativação:

- **Planejamento:** Define quais artefatos serão gerenciados e como as mudanças serão controladas.
- **Desenvolvimento:** Rastreia as alterações no código, documentações e outros artefatos, garantindo a consistência e a colaboração entre os desenvolvedores.
- **Testes:** Assegura que os testes sejam executados nas versões corretas do software e dos ambientes configurados adequadamente.
- **Implantação:** Controla a liberação das novas versões do software para os ambientes de produção, minimizando riscos e falhas.
- **Operação:** Monitora as configurações do software em produção e facilita a identificação e resolução de problemas.
- **Desativação:** Garante a remoção ordenada e segura do software quando não for mais necessário.

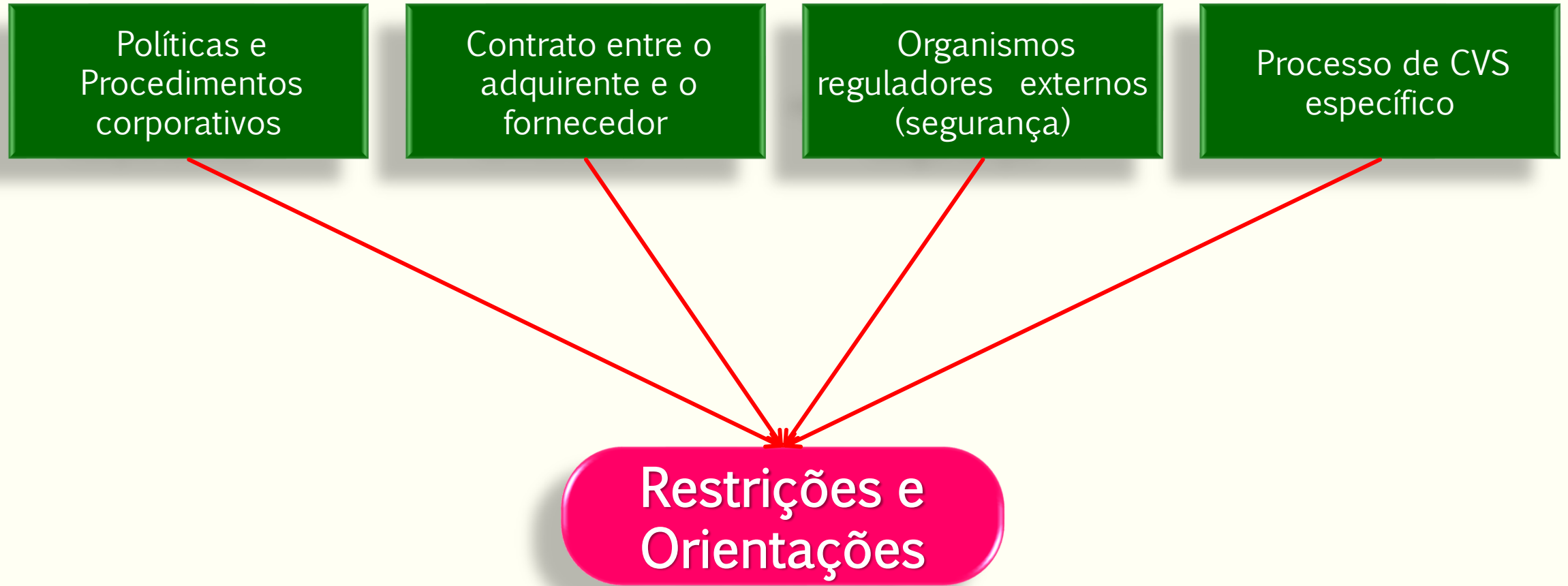
1.1. Contexto Organizacional para SCM

3. Integração com outras Práticas:

O GCS se integra e complementa outras práticas importantes no desenvolvimento de software:

- **Controle de Versão:** O GCS utiliza ferramentas de controle de versão, como Git ou SVN, para armazenar o histórico de alterações dos artefatos.
- **Integração Contínua e Entrega Contínua (CI/CD):** O GCS automatiza a integração das alterações no código e a entrega do software para os ambientes de produção, otimizando o processo de desenvolvimento.
- **DevOps:** O GCS é um elemento essencial da cultura DevOps, promovendo a colaboração entre equipes de desenvolvimento e operação.

1.2. Restrições e Orientações para o Processo GCS



1.2. Restrições e Orientações para o Processo GCS - Exemplos

1. Complexidade:

- Implementar e manter um sistema GCS eficaz pode ser **complexo e desafiador**, especialmente em projetos de grande porte ou com equipes distribuídas.
- A escolha da ferramenta GCS adequada, a definição de políticas e procedimentos adequados e a integração com outros sistemas de desenvolvimento podem ser tarefas trabalhosas e demoradas.
- A curva de aprendizado para usar ferramentas GCS pode ser alta, exigindo treinamento e suporte para os membros da equipe.

2. Escalabilidade:

- Gerenciar um grande número de arquivos, branches e versões do código pode se tornar **insustentável** com o crescimento do projeto.
- O desempenho do sistema GCS pode ser impactado negativamente à medida que a base de código aumenta, dificultando o acesso e a modificação do código.
- A integração com ferramentas de teste e implantação pode se tornar mais complexa em projetos de grande escala.

1.2. Restrições e Orientações para o Processo GCS - Exemplos

3. Integração:

- Integrar o GCS com outros sistemas de desenvolvimento, como ferramentas de teste, CI/CD e gerenciamento de projetos, pode ser **difícil e desafiador**.
- Diferentes ferramentas podem ter formatos de dados e interfaces incompatíveis, exigindo adaptadores ou soluções personalizadas.
- A integração incompleta pode levar a inconsistências de dados, duplicação de trabalho e fluxos de trabalho ineficientes.

4. Segurança:

- Garantir a **segurança** do código-fonte e dos dados de configuração é crucial para evitar acessos não autorizados, modificações mal-intencionadas e perda de dados.
- O GCS deve implementar medidas de controle de acesso robustas, criptografia de dados e mecanismos de auditoria para proteger os ativos de software.
- Vulnerabilidades no sistema GCS podem ser exploradas por agentes mal-intencionados para comprometer a integridade do código-fonte e causar danos à organização.

1.2. Restrições e Orientações para o Processo GCS - Exemplos

5. Mudança:

- Adaptar o GCS a **mudanças nos requisitos do projeto, na equipe ou na tecnologia** pode ser um processo lento e dispendioso.
- Atualizações nas ferramentas GCS, alterações nas políticas e procedimentos ou a integração com novos sistemas podem exigir tempo e esforço significativos.
- A rigidez do sistema GCS pode dificultar a adaptação a novas práticas de desenvolvimento e metodologias de trabalho

6. Visibilidade:

- Obter uma **visão completa e precisa** do *estado do código-fonte* e do *histórico de mudanças* pode ser um desafio em projetos complexos.
- A quantidade de dados e a complexidade das relações entre os arquivos podem dificultar a navegação e a análise do histórico de commits.
- A falta de ferramentas de visualização e relatórios adequadas pode dificultar a identificação de problemas e a tomada de decisões informadas

1.2. Restrições e Orientações para o Processo GCS - Exemplos

7. Custo:

- Implementar e manter um sistema GCS pode ser **custoso**, especialmente para ferramentas comerciais com recursos avançados.
- O custo das licenças, treinamento, suporte e infraestrutura deve ser considerado ao escolher uma solução GCS.
- Soluções open-source podem ser mais acessíveis, mas podem exigir mais tempo e esforço para serem configuradas e mantidas.

8. Conhecimento:

- Encontrar **profissionais qualificados** com experiência em GCS pode ser um desafio, especialmente para tecnologias e ferramentas menos comuns.
- A falta de conhecimento especializado pode dificultar a implementação, configuração e solução de problemas do sistema GCS.
- Investir em treinamento e desenvolvimento de equipe pode ser necessário para garantir a expertise necessária para gerenciar o GCS de forma eficaz.

9. Ferramentas:

- A escolha da **ferramenta GCS adequada** é crucial para o sucesso do processo.
- Diversas ferramentas estão disponíveis no mercado, cada uma com seus próprios pontos fortes e fracos.
- Fatores como tamanho do projeto, orçamento, recursos necessários, integrações com outros sistemas e preferências da equipe devem ser considerados na escolha da ferramenta.

1.3. Planejamento do GCS

- O *planejamento de um processo* de GCS para um determinado projeto deve ser
 - consistente com o contexto organizacional,
 - as restrições aplicáveis,
 - as orientações comumente aceitas e
 - a natureza do projeto (tamanho, criticidade de segurança e proteção).
- As *principais atividades de planejamento*:
 - são identificação de configuração de software,
 - controle de configuração de software,
 - contabilidade de status de configuração de software,
 - auditoria de configuração de software e
 - gerenciamento e entrega de versão de software.
 - organização e responsabilidades,
 - recursos e cronogramas,
 - seleção e implementação de ferramentas,
 - controle de fornecedores e subcontratados e
 - controle de interface
- Os *resultados da atividade de planejamento* são registrados em um **Plano GCS** (PGCS), que normalmente está sujeito à revisão e auditoria da SQA.

Planejamento

Atividades

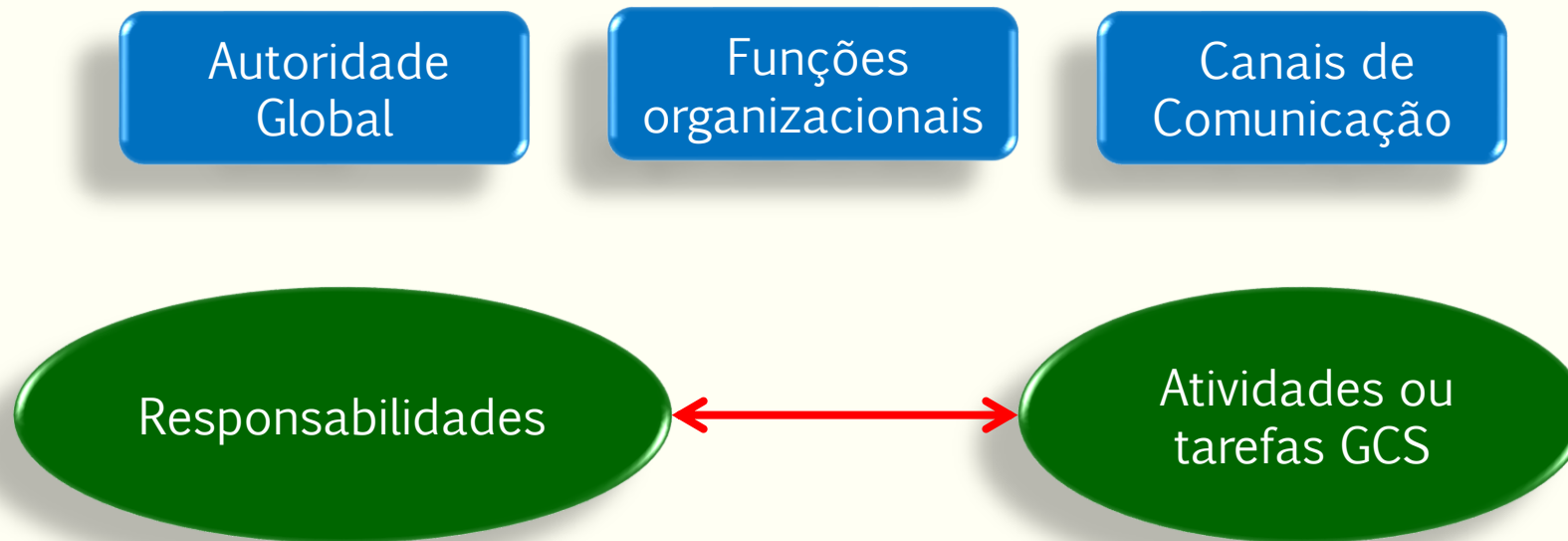
Plano
GCS

1.3. Planejamento do GCS

- 1.3.1. Organização e Responsabilidades do GCS
- 1.3.2. Recursos e Cronogramas GCS
- 1.3.3. Seleção e implementação de ferramentas
- 1.3.4. Controle de fornecedor/subcontratado
- 1.3.5 Controle de Interfaces

1.3.1. Organização e Responsabilidades do GCS

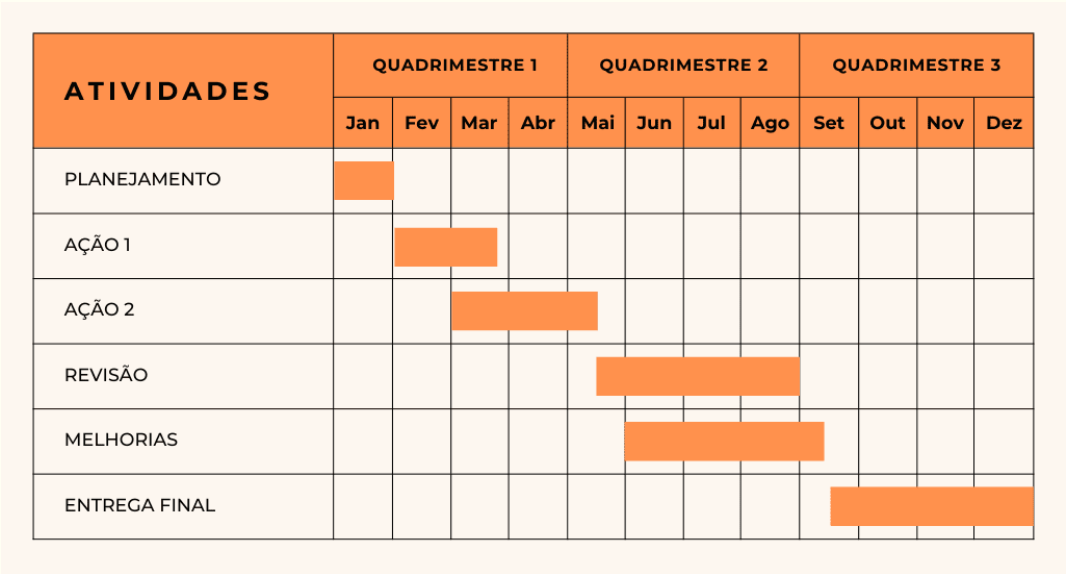
1. As *funções organizacionais* envolvidas no processo de GCS precisam ser claramente *identificadas*.
2. As *responsabilidades específicas* para determinadas *atividades ou tarefas* do GCS também precisam ser atribuídas às entidades organizacionais, seja por título ou por elemento organizacional.
3. A *autoridade global* e os *canais de comunicação* para o GCS também devem ser identificados, embora isto possa ser conseguido na fase de gestão do projeto ou de planejamento da garantia de qualidade.



1.3.2. Recursos e Cronogramas GCS

O *cronograma* é uma representação visual que organiza e estabelece a *sequência de atividades* a serem realizadas em um *determinado período*.

O GCS aborda questões de cronograma estabelecendo *sequências necessárias de tarefas* de GCS e identificando suas relações com os cronogramas e marcos do projeto estabelecidos na fase de planejamento do gerenciamento do projeto



1.3.3. Seleção e implementação de ferramentas

Como em qualquer área da *engenharia de software*, a seleção e implementação de ferramentas de GCS devem ser cuidadosamente planejadas.

- **Organização:** o que motiva a aquisição de ferramentas do ponto de vista organizacional?
- **Ferramentas:** podemos usar ferramentas comerciais ou desenvolvê-las nós mesmos?
- **Ambiente:** quais são as restrições impostas pela organização e seu contexto técnico?
- **Legado:** como os projetos utilizarão (ou não) as novas ferramentas?
- **Financiamento:** quem pagará pela aquisição, manutenção, treinamento e customização das ferramentas?
- **Escopo:** como as novas ferramentas serão implantadas – por exemplo, em toda a organização ou apenas em projetos específicos?
- **Propriedade:** quem é responsável pela introdução de novas ferramentas?

1.3.3. Seleção e implementação de ferramentas

Como em qualquer área da *engenharia de software*, a seleção e implementação de ferramentas de GCS devem ser cuidadosamente planejadas.

- **Futuro:** qual é o plano para o uso das ferramentas no futuro?
- **Mudança:** quão adaptáveis são as ferramentas?
- **Ramificação e fusão:** os recursos das ferramentas são compatíveis com as estratégias planejadas de ramificação e fusão?
- **Integração:** as diversas ferramentas de SCM integram-se entre si?
Com outras ferramentas em uso na organização?
- **Migração:** o repositório mantido pela ferramenta de controle de versão pode ser portado para outra ferramenta de controle de versão mantendo o histórico completo dos itens de configuração que contém?

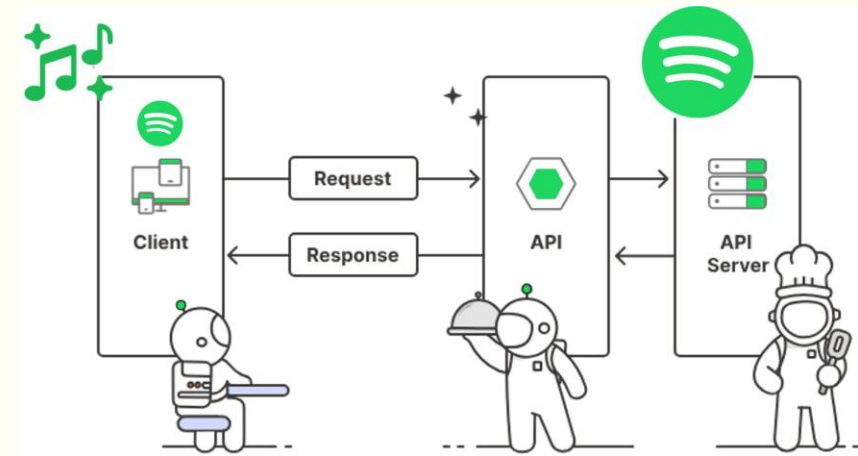
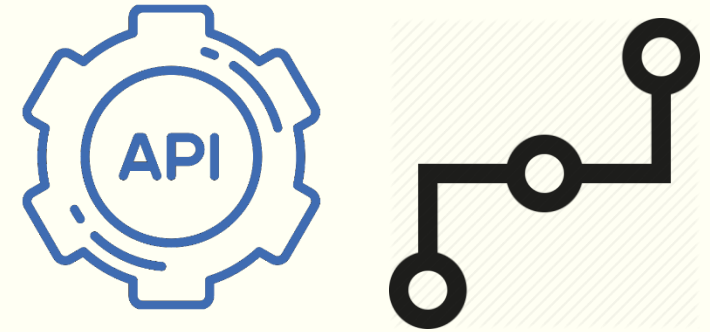
Determinar se a *bancada* de GCS será **aberta** (em outras palavras, ferramentas de diferentes fornecedores serão usadas em diferentes atividades do processo de GCS) ou **integrada** (onde os elementos da bancada são projetados para trabalharem juntos).

1.3.4. Controle de fornecedor/subcontratado

- Um projeto de software pode *adquirir ou utilizar* **produtos de software adquiridos**, como compiladores ou outras ferramentas. O planejamento SCM considera se e como esses itens serão colocados sob controle de configuração (por exemplo, integrados às bibliotecas do projeto) e como as alterações ou atualizações serão avaliadas e gerenciadas.
- Considerações semelhantes se aplicam ao **software subcontratado**. Ao utilizar software subcontratado, tanto os requisitos de GCS a serem impostos ao processo de GCS do subcontratado como parte do subcontrato quanto os meios para monitorar a conformidade precisam ser estabelecidos.

1.3.5 Controle de Interfaces

- Quando um *item de software* faz interface com outro item de software ou hardware, *uma alteração em qualquer um dos itens pode afetar o outro*.
- O planejamento do processo GCS considera como os itens de interface serão *identificados* e como as alterações nos itens serão *gerenciadas e comunicadas*.
- A função do GC pode fazer parte de um processo maior, em nível de sistema, para especificação e controle de interface; pode envolver
 - especificações* de interface,
 - planos de controle* de interface e
 - documentos* de controle de interface.
- Neste caso, o planejamento do GCS para controle de interface ocorre dentro do *contexto do processo* no nível do sistema.



1.4 O Plano GCS

- Os *resultados do planejamento* de GCS para um determinado projeto são registrados em um plano de gerenciamento de configuração de software (PGCS), um “documento vivo” que serve de referência para o processo de GCS.
- Ele é *mantido* (ou seja, atualizado e aprovado) conforme necessário durante o ciclo de vida do software.
- Na implementação do PGCS, normalmente é necessário desenvolver uma série de procedimentos subordinados mais detalhados que definem como os requisitos específicos serão executados durante as atividades do dia a dia - por exemplo, quais estratégias de ramificação serão usadas e com que frequência ocorrem as compilações e testes automatizados de todos os tipos são executados. Orientações sobre a criação e manutenção de um SCMP, com base nas informações produzidas pela atividade de planejamento, estão disponíveis em diversas fontes, como [2*].

1.5. Vigilância do GCS (Gerenciamento de Configuração de Software)

- **Necessidade:**

Após a implementação do processo do GCS, poderá ser necessário algum *grau de supervisão* para garantir que as disposições do PGCS sejam devidamente executadas.

- **Responsável:**

A *pessoa responsável* pelo GCS garante que aqueles com a responsabilidade atribuída executam corretamente as tarefas definidas do GCS. A autoridade de garantia da qualidade do software, como parte de uma atividade de auditoria de conformidade, também poderá realizar esta vigilância.

- **Ferramentas:**

O uso de *ferramentas* GCS integradas com capacidade de controle de processos pode facilitar a tarefa de vigilância (conformidade dos processos, adaptar os procedimentos). Os *requisitos de vigilância* e o *nível de flexibilidade* a ser fornecido ao engenheiro de software são considerações importantes na *seleção da ferramenta*.

1.5.1. Medidas e Medições do GCS

1.5.2. Auditorias em processo do GCS

1.5.1. Medidas e Medições do GCS

- As **medidas do GCS** podem ser projetadas para fornecer informações específicas sobre o *produto em evolução* ou para fornecer insights sobre o *funcionamento do processo* do GCS.
- Um **objetivo** relacionado ao monitoramento do processo GCS é descobrir *oportunidades de melhoria do processo*.
- As **medições dos processos** do GCS fornecem um bom meio para monitorar continuamente a *eficácia das atividades* do GCS. Estas medições são úteis para caracterizar o estado atual do processo, bem como para fornecer uma base para fazer comparações ao longo do tempo. A *análise das medições* pode produzir *percepções* que levem a *mudanças no processo e atualizações correspondentes* no Plano GCS.

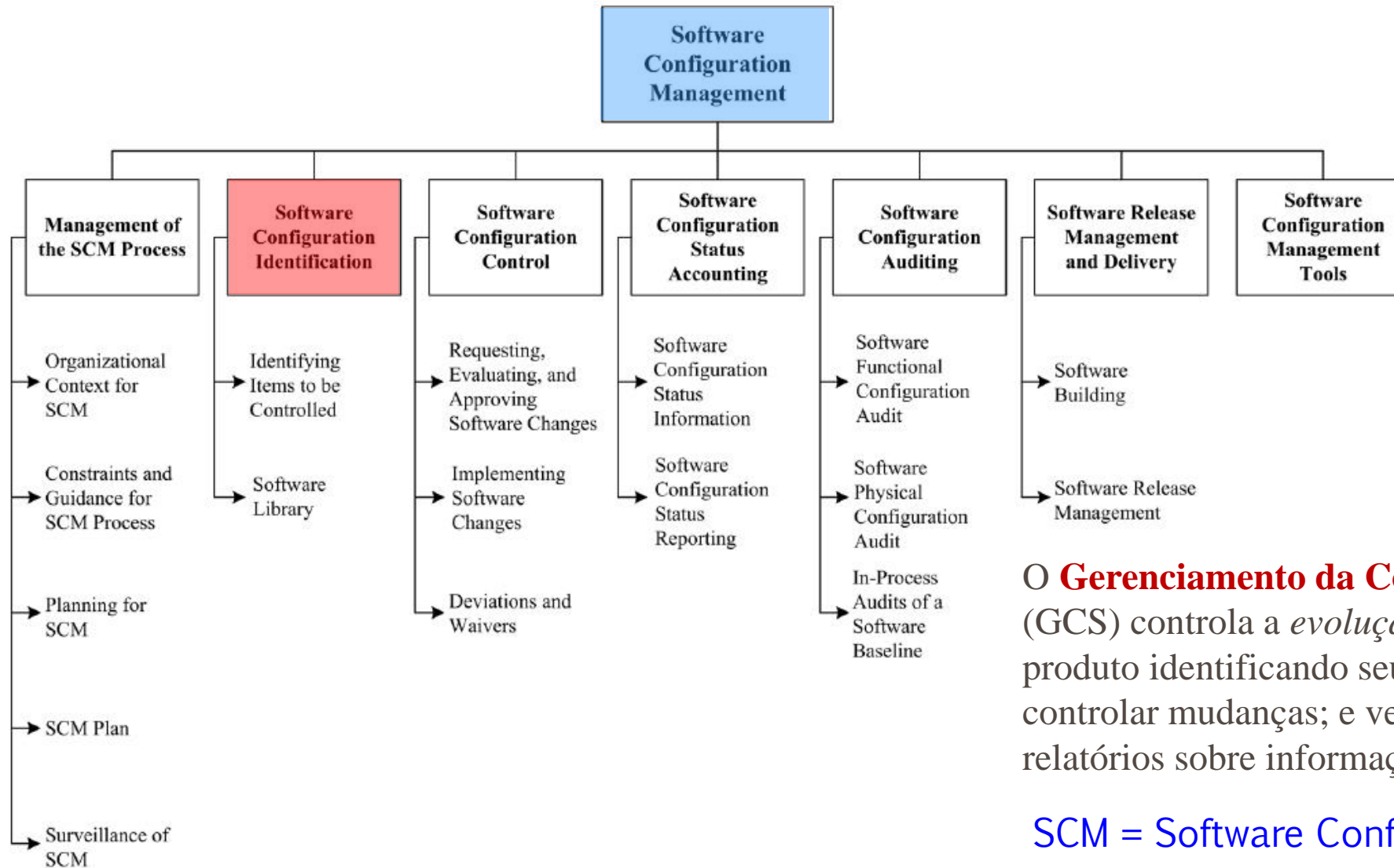


1.5.2. Auditorias em processo do GCS

- As *auditorias* podem ser realizadas durante o processo de engenharia de software
 - para investigar o *status atual* de *elementos específicos da configuração*, ou
 - para avaliar a *implementação do processo* GCS.
- A *auditoria in-processo* de GCS fornece um mecanismo mais formal para monitorar aspectos selecionados do processo e pode ser coordenada com a função SQA



2. Identificação da Configuração de Software



O **Gerenciamento da Configuração do Software** (GCS) controla a *evolução e integridade* de um produto identificando seus elementos; gerenciar e controlar mudanças; e verificar, registrar e gerar relatórios sobre informações de configuração.

SCM = Software Configuration Management

2. Identificação da Configuração de Software

A **identificação da configuração** de software

- identifica os *itens* a serem controlados,
- estabelece *esquemas de identificação* dos itens e suas versões e
- estabelece as *ferramentas e técnicas* a serem utilizadas na aquisição e gerenciamento dos itens controlados.

2.1. Identificando itens a serem controlados

2.1.1. Configuração de software

2.1.2. Item de configuração de software

2.1.3. Relacionamentos de itens de configuração de software

2.1.4. Versão do software

2.1.5. Linha de base

2.1.6. Adquirindo Itens de Configuração de Software

2.2. Biblioteca de software

2.1. Identificando itens a serem controlados

Um dos primeiros passos no controle da mudança é *identificar os itens de software a serem controlados*. Isso envolve:

- compreender a *configuração do software* dentro do contexto da configuração do sistema,
- selecionar *itens de configuração de software*,
- desenvolver uma *estratégia para rotular itens* de software e descrever seus relacionamentos, e
- identificar as *linhas de base* a serem usadas e o procedimento para a aquisição dos itens por uma linha de base.

"identificar itens a serem controlados" se refere ao processo crucial de *determinar quais elementos do sistema de software serão monitorados e gerenciados* ao longo do seu ciclo de vida. Essa etapa fundamental estabelece a base para um *controle eficaz da configuração*, garantindo a *consistência, rastreabilidade e integridade* do software.

2.1. Identificando itens a serem controlados

Para identificar os ICs de forma eficaz, é recomendável seguir estas **etapas**:

- A. Análise do Sistema:** Compreender profundamente a arquitetura, componentes e funcionamento do software.
- B. Definição de Critérios:** Estabelecer critérios para determinar quais elementos são considerados *críticos* e *precisam ser controlados*.
- C. Documentação dos ICs:** Criar um *inventário detalhado* dos ICs, incluindo sua *descrição, localização e mecanismos de controle*.
- D. Implementação do Controle:** Estabelecer *ferramentas e processos* para monitorar, registrar e gerenciar as mudanças nos ICs.

2.1. Identificando itens a serem controlados

2.1.1. Configuração de software

Configuração de software são as *características funcionais e físicas* de hardware ou software conforme estabelecidas na documentação técnica ou obtidas em um produto. Ele pode ser visto como parte de uma *configuração geral do sistema*.

2.1.2. Item de configuração de software

Um *item de configuração* (IC) é um item ou agregação de hardware ou software, ou ambos, projetado para ser gerenciado como uma *única entidade*. Um item de configuração de software (ICS) é uma entidade de software que foi estabelecida como um item de configuração. O GCS normalmente controla uma variedade de itens além do próprio código.

Itens de Configuração (ICs)

Os *itens a serem controlados*, também conhecidos como **Itens de Configuração (ICs)**, abrangem *diversos componentes* que definem o estado e o comportamento do software.

Alguns **exemplos** comuns incluem:

- **Código-fonte:** Todo o código que compõe o software, incluindo arquivos, módulos, bibliotecas e scripts.
- **Dados de configuração:** Parâmetros, variáveis e configurações que definem o comportamento do software em diferentes ambientes.
- **Documentação:** Manuais, guias do usuário e outros materiais que descrevem o funcionamento do software.
- **Executáveis:** Binários compilados prontos para serem executados.
- **Ambientes de desenvolvimento e teste:** Ferramentas, configurações e infraestrutura utilizados para o desenvolvimento, teste e implantação do software.

2.1. Identificando itens a serem controlados

2.1.3. Relacionamentos de itens de configuração de software

As relações estruturais entre os ICSs selecionados e suas partes constituintes afetam outras atividades ou tarefas do GCS, como a construção de software ou a análise do impacto das mudanças propostas..

2.1.4. Versão do software

- Os itens de software evoluem à medida que o projeto de software avança. Uma **versão** de um item de software *é uma instância identificada de um item*. Pode ser pensado como o estado de um item em evolução.
- No mundo do software, a **versão** representa um *identificador único* atribuído a um estado específico do software, marcando as modificações e o desenvolvimento ao longo do tempo.
- Cada **versão** *encapsula um conjunto definido de funcionalidades, correções de bugs e aprimoramentos*, fornecendo uma referência clara para diferenciar diferentes iterações do software.

2.1.4. Versão do software

O **sistema de versionamento** de software se assemelha a um sistema de numeração, geralmente composto por números e letras, seguindo uma estrutura lógica que reflete o tipo e o nível de mudança. Convenções comuns incluem:

- **Versão Principal:** Indica grandes mudanças funcionais ou na arquitetura do software.
- **Versão Secundária:** Representa adições incrementais de funcionalidades ou melhorias significativas.
- **Versão de Revisão:** Indica correções de bugs ou pequenas modificações.

Por exemplo, a **versão "1.2.3"** de um software pode significar:

- **Versão Principal 1:** Mudança fundamental na funcionalidade ou arquitetura.
- **Versão Secundária 2:** Adição de novas funcionalidades ou melhorias significativas.
- **Versão de Revisão 3:** Correção de bugs ou pequenas modificações.

2.1. Identificando itens a serem controlados

2.1.5. Linha de base

2.1.6. Adquirindo Itens de Configuração de Software

2.1.5 Linha de Base (baseline)

- Uma **linha de base de software** é uma versão formalmente aprovada de um item de configuração (independentemente da mídia) que é formalmente designada e fixada em um *momento específico* durante o *ciclo de vida do item de configuração*.
- A linha de base só pode ser alterada através de procedimentos formais de controle de alterações. Uma linha de base, juntamente com todas as alterações aprovadas na linha de base, representa a configuração aprovada atual.
- As linhas de base comumente usadas incluem linhas de base funcionais, alocadas, de desenvolvimento e de produto.
 - A *linha de base funcional* corresponde aos requisitos do sistema revisados.
 - A *linha de base alocada* corresponde à especificação de requisitos de software revisada e à especificação de requisitos de interface de software.
 - A *linha de base de desenvolvimento* representa a evolução da configuração do software em momentos selecionados durante o ciclo de vida do software.
 - A *linha de base do produto* corresponde ao produto de software completo entregue para integração do sistema.
- As linhas de base a serem utilizadas para um determinado projeto, juntamente com os níveis de autoridade associados necessários para a aprovação da mudança, são normalmente identificados no Plano de GCS.



Prof. Dr. Ausberto S. Castro Vera
Ciência da Computação
UENF-CCT-LCMAT
Campos, RJ

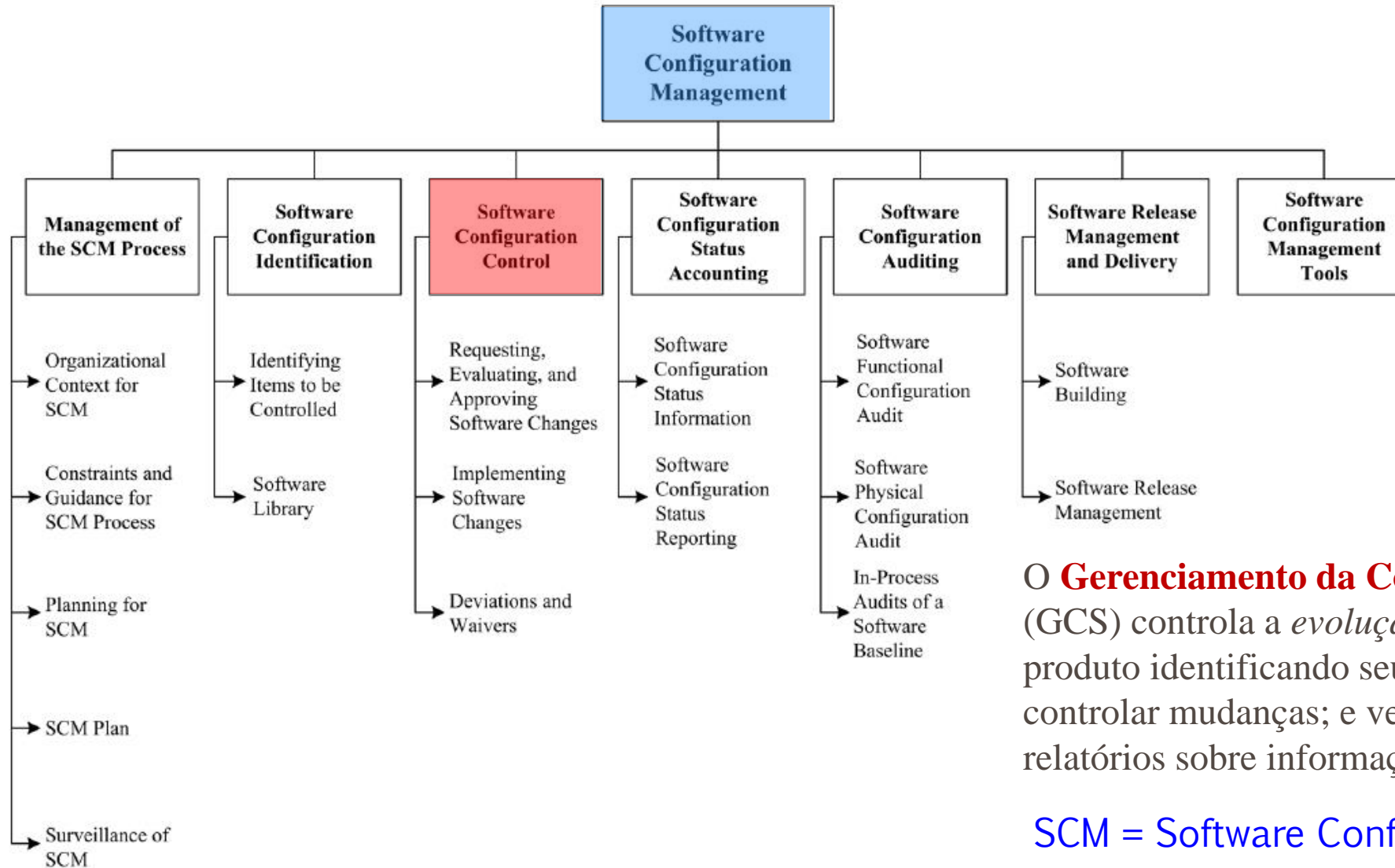
ascv@uenf.br



1.3. Planejamento do GCS

- Uma **ramificação** é definida como um conjunto de versões de arquivos fonte em evolução. Uma **ramificação**, também conhecida como **branch**, é um **caminho alternativo de desenvolvimento** dentro de um repositório de código-fonte. Ela representa uma **cópia de um ponto específico no tempo da base de código principal**, permitindo que os desenvolvedores trabalhem em **mudanças independentes** sem afetar o código principal.
- A **fusão** consiste em combinar diferentes alterações no mesmo arquivo. Isso normalmente ocorre quando mais de uma pessoa altera um item de configuração.
- Existem muitas **estratégias** de ramificação e fusão de uso comum. As estratégias de ramificação e fusão devem ser cuidadosamente planejadas e comunicadas, uma vez que têm impacto em muitas atividades de GCS.
- O **modelo de ciclo de vida de desenvolvimento de software** também impacta as atividades de GCS, e o planejamento de GCS deve levar isso em consideração.
 - Por exemplo, *a integração contínua* é uma prática comum em muitas abordagens de desenvolvimento de software. Geralmente é caracterizado por ciclos frequentes de *construção, teste e implantação*.

3. Controle da Configuração de Software



O **Gerenciamento da Configuração do Software** (GCS) controla a *evolução e integridade* de um produto identificando seus elementos; gerenciar e controlar mudanças; e verificar, registrar e gerar relatórios sobre informações de configuração.

SCM = Software Configuration Management

3. Controle da Configuração de Software

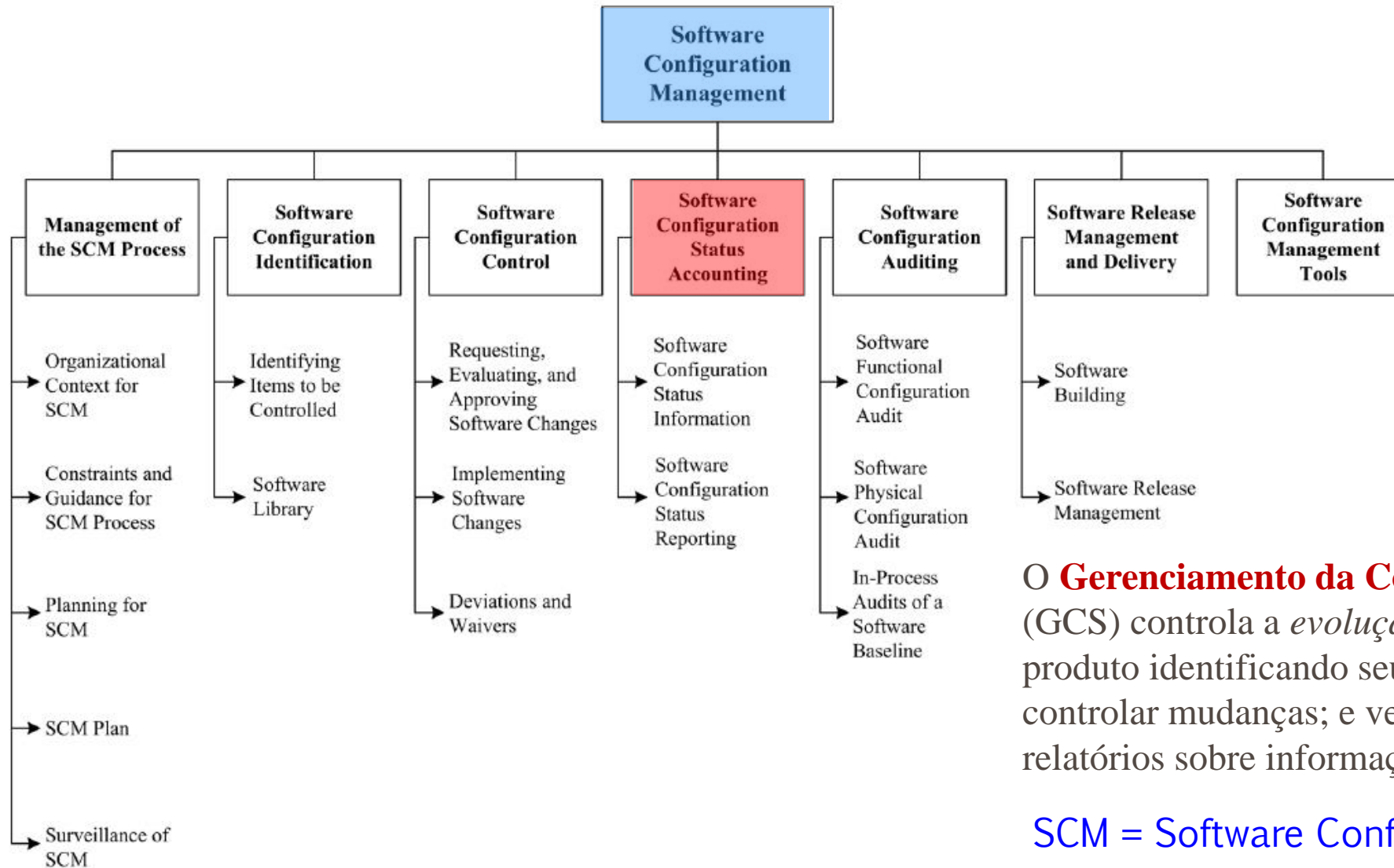
3.1. Solicitar, avaliar e aprovar alterações de software

3.1.1. Placa de controle de configuração de software

3.1.2. Processo de solicitação de alteração de software

3.2. Implementando Mudanças de Software

4. Contabilidade de Status de Configuração de Software



O **Gerenciamento da Configuração do Software** (GCS) controla a *evolução e integridade* de um produto identificando seus elementos; gerenciar e controlar mudanças; e verificar, registrar e gerar relatórios sobre informações de configuração.

SCM = Software Configuration Management

4. Contabilidade de status de Configuração de Software

4.1. Informações de status de configuração de software

4.2. Relatório de status de configuração de software