11ª Semana de Tecnologia Metroferroviária – Fórum Técnico

Associação dos Engenheiros e Arquitetos de Metrô - AEAMESP

IMPORTÂNCIA DA QUALIDADE DE SOFTWARE EM SEGURANÇA CRÍTICA

Introdução

O objetivo deste trabalho, frente ao atual cenário do setor Metroferroviário, é de conscientização e esclarecimento sobre o Controle da Qualidade de Software de Sistemas Sinalização e Controle, demonstrando providências exigíveis a Empresas Fornecedoras e Projetistas destes Sistemas.

No meio acadêmico, são ainda muito discutidas as problemáticas da realização de funções de segurança através de Software. Apesar do tímido número de evidências científicas de que equipamentos baseados em Software se comportam de maneira segura, a prática em diversas áreas demonstra um quadro irreversível de aplicação intensiva desses equipamentos nos mais diversos sistemas, conforme vivenciamos no nosso dia a dia.

A obtenção de um Software Seguro envolve inúmeros fatores, entre eles a experiência e o preparo de toda a equipe envolvida no Processo, o uso de técnicas e ferramentas apropriadas, uma criteriosa Análise de Segurança Independente e, dentre os atributos da Qualidade de Software, em especial sua Confiabilidade.

Confiabilidade de Software

A Confiabilidade de Software pode ser definida como a “probabilidade de operação livre de falhas por um período de tempo especificado em um determinado ambiente” [ANSI 1991]. A Confiabilidade é um dos atributos da Qualidade de Software, uma propriedade multidimensional que inclui outros fatores de satisfação do cliente tais como funcionalidade, facilidade de uso, desempenho, capacidade, documentação, facilidades de suporte, instalação e manutenção [Grad 1987, Grad 1992].

A Confiabilidade de Software é geralmente aceita como o fator principal da Qualidade de Software, pois quantifica as falhas de Software – que podem tirar de operação um poderoso sistema ou até mesmo trazer riscos às pessoas e patrimônio [Lyu 1996].

Os métodos mais respeitados e evoluídos para a avaliação da confiabilidade de software, tal como descrito em “Software Reliability Engineering” [Musa 1999], são baseados na observação de falhas do produto terminado, sustentada por análise estatística. Estes métodos são a única forma de medir a confiabilidade do produto, pois são baseadas na observação direta de seu comportamento. As importantes desvantagens destas abordagens são [Dale 1990]:

- estão preocupadas puramente com a medida da confiabilidade, enquanto outras informações sobre, por exemplo, os métodos que foram utilizados no desenvolvimento do produto, são ignoradas; e
- só podem ser utilizadas quando o produto final existe, portanto sua aplicação só pode ser iniciada a partir da obtenção de dados durante a fase de testes do desenvolvimento.

Uma alternativa pesquisada para a obtenção de Software de alta confiabilidade é a especificação e análise dos Processos de Desenvolvimento de Software adotados em um projeto, em conformidade com as normas técnicas adequadas a aplicação, além de modelos de ampla aceitação.

Adotando-se uma abordagem deste tipo, a avaliação da confiabilidade deve ser realizada sobre os produtos intermediários de cada fase do processo de desenvolvimento, antes de se obter o produto final.

Segurança de Software (SW Safety)

A Segurança de um sistema S(t) é a probabilidade de um sistema efetuar suas operações de forma correta, ou descontinuar seu funcionamento de forma a não comprometer a operação de outros sistemas ou comprometer a Segurança [Johnson 1989]. As técnicas de “falha segura” (fail-safe) têm sido empregadas desde em sistemas eletrônicos convencionais e até mesmo nas principais rotinas de diagnóstico de funções de Software e componentes de Hardware, bem como nos conceitos de programação defensiva.

Especificação de Requisitos Técnicos de Software

A Especificação Técnica de Software abordada a seguir foi elaborada, a princípio, tendo em vista um horizonte de aplicação da tecnologia “Communications Based Train Control” (CBTC) na

implantação e modernização de Sistemas de Sinalização e Controle. O emprego desta nova tecnologia traz desafios tais como:

- uso intensivo de Software para a realização de funções de Tempo Real e Segurança Crítica;
- fatores de aumento natural do risco e a conseqüente necessidade de sua mitigação através da tecnologia, tais como a diminuição do intervalo entre trens (“headway”), operação automática sem condutor (“driverless”), funções de diagnóstico e recuperação remotas;
- transmissão intensiva por meio aberto de sinais vitais, trazendo preocupações típicas da Segurança da Informação (“security”), além do envelhecimento de informações de tempo real, que devem ser tratadas pelo Software de comunicação;
- determinação da posição das composições e distâncias percorridas utilizando múltiplos sensores, e tratamento por algoritmos que incluem informações sobre margens de segurança para determinação e cumprimento de pontos de parada com segurança;
- emprego limitado de hardware intrinsecamente “fail-safe” e componentes com modos de falha bem definidos; e
- os conceitos consagrados de Segurança de um Sistema de Sinalização e Controle devem ser implementados de forma completa e claramente verificáveis [IEEE 2005].

No entanto, devido a incorporação da experiência obtida pelo METRÔ-SP no projeto e implantação dos sistemas de Sinalização e Controle, somada a estudos realizados em conjunto com a Fundação para o Desenvolvimento Tecnológico da Engenharia - FDTE, foi considerado adequado o emprego desta metodologia para a aquisição de Software do Sistema de Sinalização e Controle de Trens – Estações e Vias da Extensão da Linha 2 – Verde, trecho Ana Rosa – Ipiranga, mesmo tratando-se de um Sistema Microprocessado Convencional, e dentre seus principais requisitos, serão enfatizados:

❖ Conformidade com CMMi

A Capacidade do Processo [CMU/SEI 2002] estabelece o grau de refinamento e institucionalização com que o processo é realizado em uma organização, em cinco níveis discretos, em sua representação por estágios, conforme representado na Figura 1.

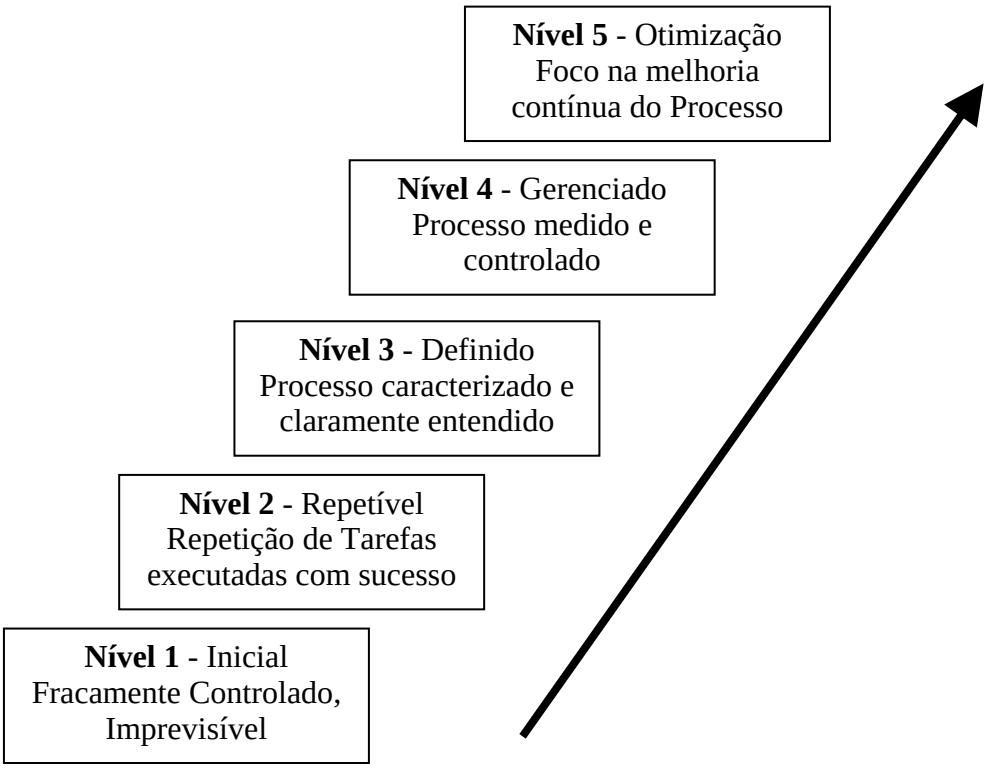


Figura 1 - Estágios do CMMi

Os atributos de processo pertinentes ao nível 2 do CMMi, pelo qual estamos iniciando a utilização desta metodologia, ilustrados na Figura 2, podem ser assim resumidos:

- o Processo é Executado;
- o Processo atinge seus resultados definidos; e
- os produtos de trabalho do Processo são gerenciados.

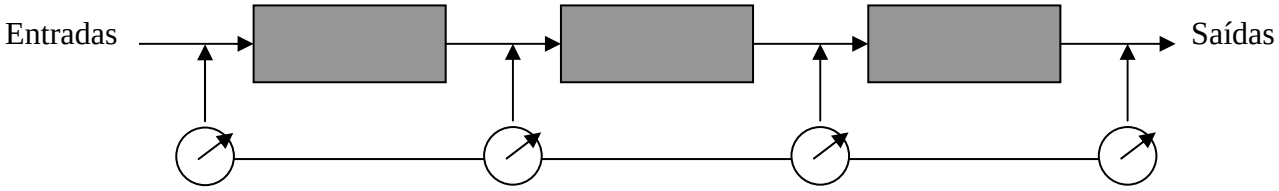


Figura 2 - CMMi nível 2 - Visibilidade

A Especificação Técnica de Software apresenta o seguinte requisito:

A Contratada deve adotar a metodologia estabelecida no "Capability Maturity Model Integration for Systems Engineering and Software Engineering" (CMMI-SE/SW), versão 1.1 ou superior, do "Software Engineering Institute / Carnegie Mellon University" (SEI/CMU) para todo o Software fornecido, sendo que:

Deve ser adotado CMMI-SE/SW nível dois (2) ou superior, na representação em estágios "Staged Representation";

A Contratada deve comprovar pelo "Standard CMMI Appraisal Method for Process Improvement" (SCAMPI) [CMU/SEI 2001], o emprego desta metodologia por toda equipe envolvida no projeto.

- **Nível de Integridade de Segurança (SIL)**

A Especificação Técnica de Software apresenta o seguinte requisito com relação ao Nível de Integridade de Segurança de Software, que juntamente com os parâmetros de Segurança do Hardware irão determinar a Segurança do Sistema, traduzido em valores na Figura 3.

Os conceitos de "Fail Safe" devem ser utilizados no Software relacionado às funções de Segurança Crítica ou Vitais. Deve atender ao "Software Safety Integrity Level" (SIL) quatro (4), conforme a norma IEC 61508 "Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems".

SIL	Falhas inseguras / hora
4	$10^{-9} < \lambda_u < 10^{-8}$
3	$10^{-8} < \lambda_u < 10^{-7}$
2	$10^{-7} < \lambda_u < 10^{-6}$
1	$10^{-6} < \lambda_u < 10^{-5}$

Figura 3 – Nível de Integridade de Segurança [IEC 1999]

- **Conformidade com a Norma CENELEC 50128**

Um importante requisito na Especificação Técnica de Software é a conformidade com a Norma CENELEC 50128 [CENELEC 2001], específica para o Software aplicado a Sistemas de Sinalização e Controle:

Devem ser atendidas as recomendações da norma CENELEC 50128 "Railway applications – Communications, Signaling and Processing Systems – Software for Railway Control and Protection Systems".

Conforme podemos observar no diagrama seguinte, presente na Figura 4, as atividades específicas para a obtenção do Software estão vinculadas às especificações do Sistema onde ele é empregado.

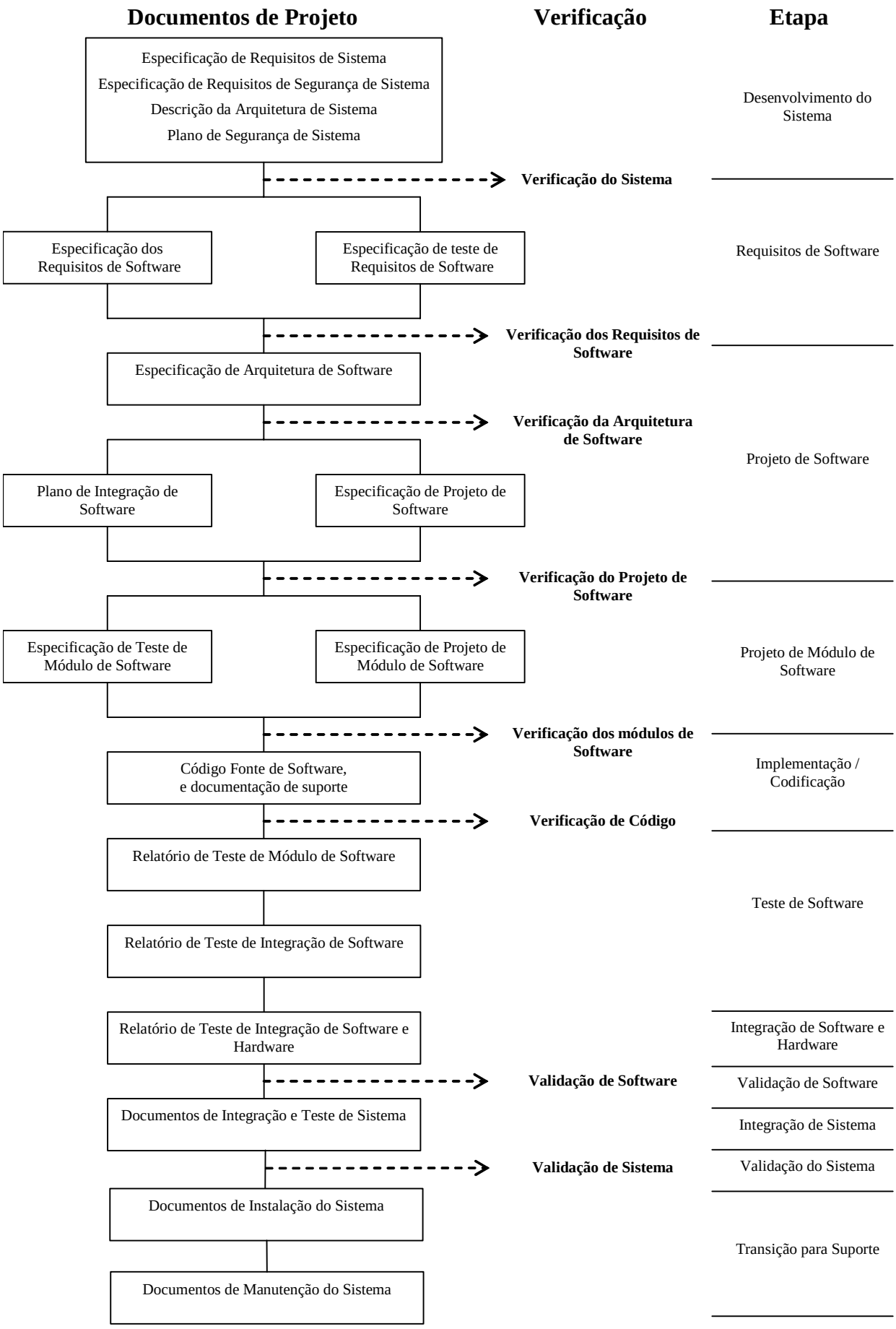


Figura 4 - Ciclo de Vida e Artefatos [CENELEC 2001]

❖ **Planejamento de Software**

Antes do início do desenvolvimento do Software, foi detectada a necessidade de solicitar à empresa fornecedora e projetista a elaboração de Planos específicos aos processos de obtenção de Software com o nível de Segurança adequado à aplicação no sistema de Sinalização e Controle:

- Plano de Gerenciamento de Software;
- Plano de Garantia da Qualidade de Software;
- Plano de Gerência de Configuração de Software;
- Plano de Verificação de Software de Aplicação Crítica e de Segurança; e
- Plano de Validação de Software.

Estes documentos deverão ser submetidos à aprovação, incluindo eventuais alterações ao longo do fornecimento.

❖ **Ciclo de Vida**

A Especificação Técnica de Software apresenta, ainda, uma descrição das fases de um Ciclo de Vida do Software de referência, listadas a seguir:

- Especificação de Requisitos de Software;
- Arquitetura de Software;
- Projeto de Software;
- Implementação de Software;
- Integração e Testes de Software;
- Validação de Software;
- Aceitação de Software; e
- Operação e Manutenção.

Conclusões

Este trabalho tem por objetivo obter uma maior garantia do nível de Segurança do Software e, ainda, a redução do re-trabalho e conseqüente otimização dos recursos empregados, possibilitando também um maior controle sobre o processo e as tecnologias em implantação.

Não deixando de observar as limitações impostas pelo estágio de desenvolvimento atual da indústria nacional, é importante vislumbrar esta oportunidade e necessidade de progresso,

aproveitando a experiência e capacidade de inovação, e incorporando a iniciativa dos maiores consumidores de Software mundiais e a produção científico-acadêmica disponível nesta área.

Referência Bibliográfica

[CMU/SEI 2002] Carnegie Mellon University Software Engineering Institute: “CMMI-SE/SW – Capability Maturity Model Integration (CMMI), Version 1.1: CMMI for Systems Engineering and Software Engineering, Staged Representation”, CMU/SEI-2002-TR-002, 2002.

[CMU/SEI 2001] Carnegie Mellon University Software Engineering Institute: “SCAMPI – Standard CMMI Appraisal Method for Process Improvement Version 1.1: Method Definition Document”, CMU/SEI-2001-HB-001, 2001.

[CENELEC 2001] European Committee for Electrotechnical Standardization: “Railway Applications – Communications, Signaling and Processing Systems – Software for Railway Control and Protection Systems”, BS-EN 50128, 2001.

[Dale 1990] Dale, C.; Paul Rook (ed.) – “Software Reliability Handbook” – Elsevier Applied Science, 1990.

[Johson 1989] Barry W. Johnson –“Design and Analyses of fault-Tolerant Digital Systems” – Addison-Wesley, 1989.

[Lyu 1996] Michael R. Lyu (ed.) – “Handbook of Software Reliability Engineering”, McGraw-Hill, 1996.

[Musa 1999] Musa J.D. – “Software Reliability Engineering”, McGraw-Hill, 1999.

[ANSI 1991] ANSI/IEEE, "Standard Glossary of Software Engineering Terminology", STD-729-1991, ANSI/IEEE, 1991.

[GRAD 1987] Grady, Robert B.; Caswell, Deborah L.- “Software Metrics: Establishing a Company-Wide Program.” Prentice Hall, 1987.

[GRAD 1992] Grady, Robert B. - “Practical Software Metrics for Project Management and Process Improvement”. Prentice Hall, 1992.

[IEEE 2005] IEEE Standard for Communications-Based Train Control (CBTC) Performance and Functional Requirements, Institute of Electrical and Electronics Engineers, IEEE 1474-1, 2005.

[IEC 1999] Functional safety of electrical/electronic/programmable electronic safety-related systems - General requirements, IEC 61508-1, 1999.

Eng. Christian Becker Bueno de Abreu

e-mail: cbecker@metrosp.com.br – tel. 55-11-3371-7222

Engenheiro de Projeto da Companhia do Metropolitano de São Paulo – METRÔ

Departamento de Projeto de Sistemas de Controle e Comunicações – GEP/EPS

Prof. Dr. Paulo Sérgio Cugnasca

Escola Politécnica da Universidade de São Paulo (EPUSP)

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)

Grupo de Análise de Segurança (GAS)

Importância da
Qualidade de Software
em
Segurança Crítica

Eng. Christian *Becker* Bueno de Abreu

Prof. Dr. Paulo Sérgio Cugnasca

Natureza do Software ???

C

pro

ta

Todas

ção de um

[NBR ISO 9000-3]

Software - Definições

“Software”:

Criação intelectual compreendendo os programas, procedimentos, regras e qualquer documentação correlata de um sistema de processamento de dados

NOTA: O “software” não depende do meio no qual é registrado.

Desenvolvimento:

Todas as atividades a serem realizadas para a criação de um produto de Software.

[NBR ISO 9000-3]

Qualidade & Segurança - Definições

Qualidade - indicadores

- **Confiabilidade**
- Funcionalidade
- Facilidade de uso
- Desempenho
- Capacidade
- Documentação
- Suporte, Instalação e Manutenção

Segurança - S(t)

Probabilidade de um sistema efetuar suas operações de forma correta ou descontinuar seu funcionamento de forma a não comprometer a operação de outros sistemas ou comprometer a Segurança

Qualidade ∩ Segurança - Confiabilidade

QUALIDADE

CONFIABILIDADE

Confiabilidade - R(t)

Probabilidade de **operação livre de falhas** por um período de tempo especificado em um determinado ambiente

$$MTTF = \int_0^{\infty} R(t).dt$$

Qualidade & Segurança - Definições

Qualidade - indicadores

- **Confiabilidade**
- Funcionalidade
- Facilidade de uso
- Desempenho
- Capacidade
- Documentação
- Suporte, Instalação e Manutenção

Segurança - S(t)

Probabilidade de um sistema **efetuar suas operações de forma correta** ou descontinuar seu funcionamento de forma a não comprometer a operação de outros sistemas ou **comprometer a Segurança**

Qualidade & Segurança - Definições

- Con… sistema
- Func… **ações de**
- Facil… **a** ou
- Dese… seu
- Capa… forma a
- Docu… operação
- Supo… **nas** ou
- Manu… **egurança**

Medição da Confiabilidade

Medir a Confiabilidade:

Observação de falhas

+

Análise estatística.

Problemas:

- Início na etapa de testes:
Produto deve estar terminado

- Tempo necessário:
Software Ultra-confiabilidade

- Informações ignoradas:
Métodos empregados

Alternativa:

Especificação e Análise dos Processos de Desenvolvimento de Software.

Conformidade com Normas Técnicas adequadas e Modelos de amplo aceitação.

Avaliação da Confiabilidade sobre os Produtos Intermediários de cada fase do Processo de Desenvolvimento.

Medição da Confiabilidade

Medir a Confiabilidade:

Observação de falhas

+

Análise estatística.

Problemas:

- Início na etapa de testes: Produto deve estar terminado
- Tempo necessário: Software Ultra-confiabilidade
- Informações ignoradas: Métodos empregados

Alternativa:

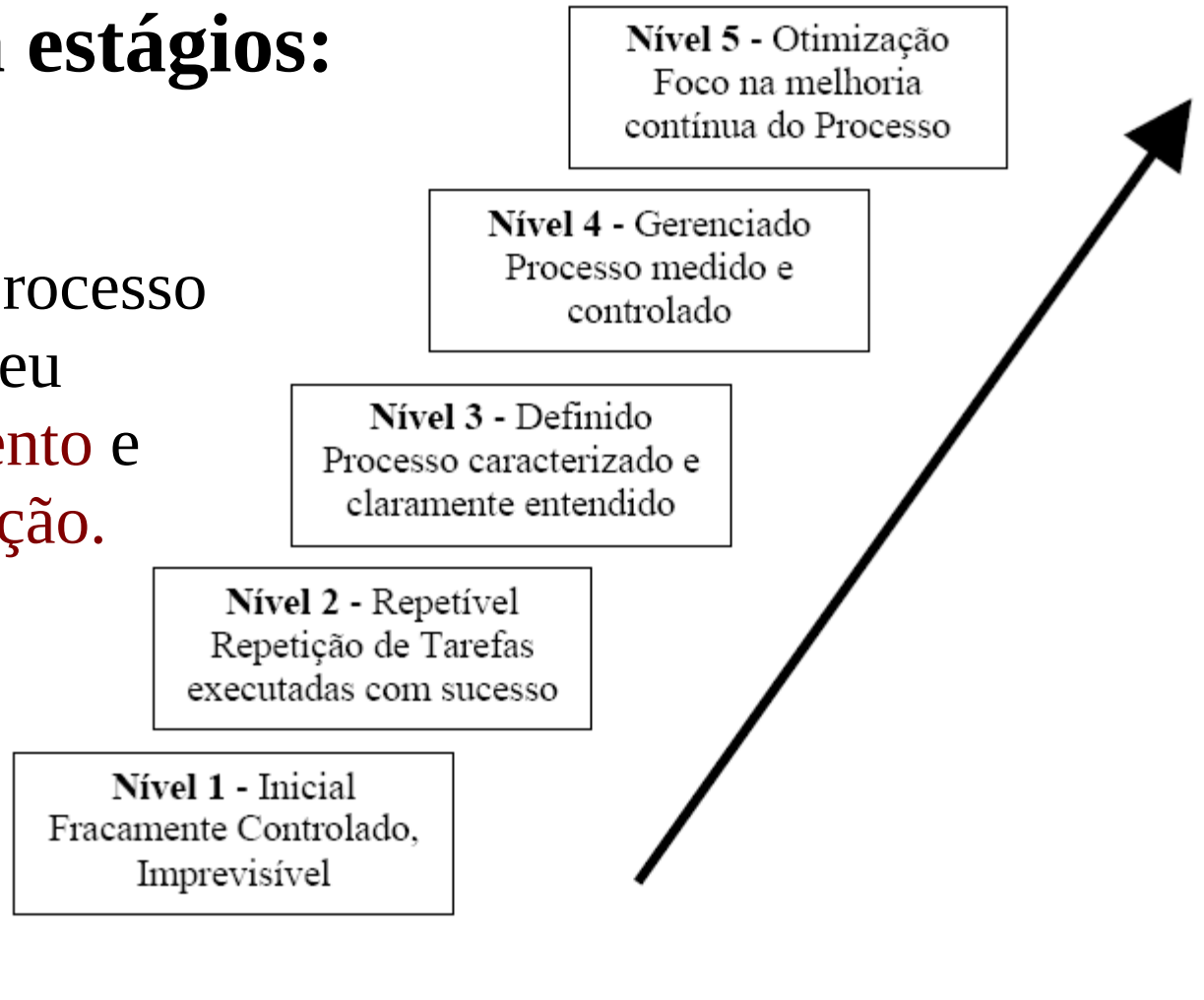
Especificação e Análise dos Processos de Desenvolvimento de Software.

Conformidade com Normas Técnicas adequadas e Modelos de ampla aceitação.

Avaliação da Confiabilidade sobre Artefatos Intermediários de cada fase do Processo de Desenvolvimento.

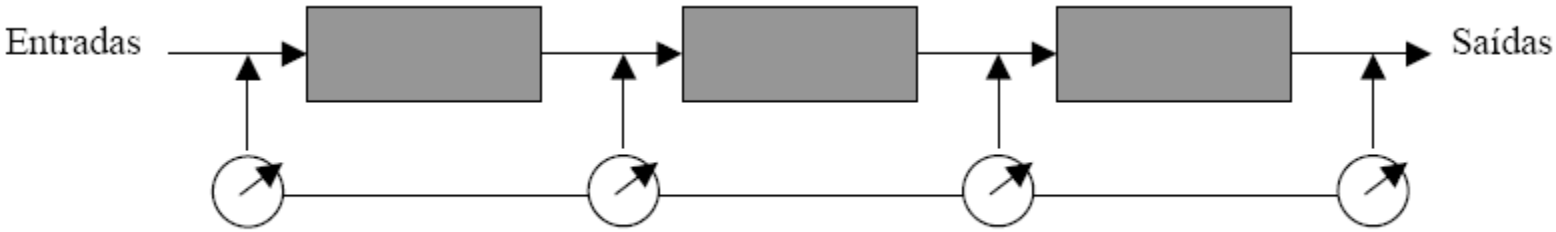
Representação em estágios:

A Capacidade do Processo estabelece o seu grau de refinamento e institucionalização.



Visibilidade:

CMMi nível 2



Requisito Técnico de Software - Qualidade

Deve ser adotada a metodologia estabelecida no “Capability Maturity Model Integration for Systems Engineering and Software Engineering” (CMMI-SE/SW), versão 1.1 ou superior, do “Software Engineering Institute / Carnegie Mellon University” (SEI/CMU) para todo o Software fornecido, sendo que:

- Deve ser adotado CMMI-SE/SW **nível dois** (2) ou superior, na representação em estágios “Staged Representation”;
- Deve ser comprovado pelo “Standard CMMI Appraisal Method for Process Improvement” (SCAMPI), o emprego desta metodologia por toda equipe envolvida no projeto.

[EPS 2004]

Fail Safe (Falha Segura)

A validade deste conceito é, em geral, baseada na experiência, mas tem aplicação limitada ao desenvolvimento e uso de sistemas grandes e complexos empregando microprocessadores comerciais.

O crescimento exponencial no número de combinações de falhas a serem consideradas quando usando tais componentes significam que uma *abordagem determinística* é, geralmente impraticável. Com tais sistemas complexos, a *abordagem probabilística* pode ser usada efetivamente. **[CENELEC 50126]**

Exemplos de aplicação em Software:

- Rotinas de diagnóstico
- Programação Defensiva

Níveis de Risco

Determinação dos Níveis de Risco

* Freqüência da ocorrência de um evento de ameaça	Níveis de Risco			
Freqüente	Indesejável	Intolerável	Intolerável	Intolerável
Provável	Tolerável	Indesejável	Intolerável	Intolerável
Ocasional	Tolerável	Indesejável	Indesejável	Intolerável
Remoto	Negligenciável	Tolerável	Indesejável	Indesejável
Improvável	Negligenciável	Negligenciável	Tolerável	Tolerável
Inacreditável	Negligenciável	Negligenciável	Negligenciável	Negligenciável
	Insignificante	**Marginal**	**Crítico**	**Catastrófico**
	Níveis de gravidade da conseqüência da ameaça.			

IEC 61508 - Parâmetros para determinar SIL

Conseqüência do Risco

C1 Perdas Pequenas.

C2 Perdas sérias e permanentes para uma ou mais pessoas, ou morte de uma pessoa.

C3 Morte de várias pessoas.

C4 Muitas pessoas assassinadas.

Freqüência e Tempo de Exposição

F1 Exposição de rara a freqüente na região de perigo.

F2 Exposição de freqüente a permanente na região de perigo.

Possibilidade de evitar o evento

P1 Possível sob certas condições.

P2 Quase impossível.

Probabilidade de ocorrências indesejáveis

W1 Probabilidade muito pequena de que as ocorrências acontecerão.

W2 Probabilidade pequena de que as ocorrências acontecerão.

W3 Probabilidade relativamente alta de que as ocorrências acontecerão.

IEC 61508 - Parâmetros para determinar SIL

Conseqüência do Risco

C1 Perdas Pequenas.

C2 Perdas sérias e permanentes para uma ou mais pessoas, ou morte de uma pessoa.

C3 Morte de várias pessoas.

C4 Muitas pessoas assassinadas.

Freqüência e Tempo de Exposição

F1 Exposição de rara a freqüente na região de perigo.

F2 Exposição de freqüente a permanente na região de perigo.

Possibilidade de evitar o evento

P1 Possível sob certas condições.

P2 Quase impossível.

Probabilidade de ocorrências indesejáveis

W1 Probabilidade muito pequena de que as ocorrências acontecerão.

W2 Probabilidade pequena de que as ocorrências acontecerão.

W3 Probabilidade relativamente alta de que as ocorrências acontecerão.

IEC 61508 - Parâmetros para determinar SIL

Conseqüência do Risco

C1 Perdas Pequenas.

C2 Perdas sérias e permanentes para uma ou mais pessoas, ou morte de uma pessoa.

C3 Morte de várias pessoas.

C4 Muitas pessoas assassinadas.

Freqüência e Tempo de Exposição

F1 Exposição de rara a freqüente na região de perigo.

F2 Exposição de freqüente a permanente na região de perigo.

Possibilidade de evitar o evento

P1 Possível sob certas condições.

P2 Quase impossível.

Probabilidade de ocorrências indesejáveis

W1 Probabilidade muito pequena de que as ocorrências acontecerão.

W2 Probabilidade pequena de que as ocorrências acontecerão.

W3 Probabilidade relativamente alta de que as ocorrências acontecerão.

IEC 61508 - Parâmetros para determinar SIL

Conseqüência do Risco

C1 Perdas Pequenas.

C2 Perdas sérias e permanentes para uma ou mais pessoas, ou morte de uma pessoa.

C3 Morte de várias pessoas.

C4 Muitas pessoas assassinadas.

Freqüência e Tempo de Exposição

F1 Exposição de rara a freqüente na região de perigo.

F2 Exposição de freqüente a permanente na região de perigo.

Possibilidade de evitar o evento

P1 Possível sob certas condições.

P2 Quase impossível.

Probabilidade de ocorrências indesejáveis

W1 Probabilidade muito pequena de que as ocorrências acontecerão.

W2 Probabilidade pequena de que as ocorrências acontecerão.

W3 Probabilidade relativamente alta de que as ocorrências acontecerão.

IEC 61508 - Determinar Redução de Risco e SIL

			W3	W2	W1
	C1		A	-	-
C2	F1	P1	B	A	-
		P2	C	B	A
	F2	P1	D	C	B
		P2	E	D	C
C3	F1		F	E	D
	F2		G	F	E
	C4		H	G	F

SIL	Falhas inseguras / hora
4	$10^{-9} < \lambda_u < 10^{-8}$
3	$10^{-8} < \lambda_u < 10^{-7}$
2	$10^{-7} < \lambda_u < 10^{-6}$
1	$10^{-6} < \lambda_u < 10^{-5}$

Integridade de Segurança

Requisitos de Segurança

Requisitos Especiais de Segurança

B, C	SIL 1
D	SIL 2
E, F	SIL 3
G	SIL 4
H	Sistemas de Proteção não Suficientes

IEC 61508 - Determinar Redução de Risco e SIL

		W3	W2	W1
C1		A	-	-
F1	P1	B	A	-
	P2	C	B	A
C2				
F2				
C3				
C4				

SIL	Falhas inseguras / hora
4	$10^{-9} < \lambda_u < 10^{-8}$
3	$10^{-8} < \lambda_u < 10^{-7}$

Redução de Risco	*Nível de Integridade de Segurança*
-	Sem Requisitos de Segurança
A	Sem Requisitos Especiais de Segurança
B, C	SIL 1
D	SIL 2
E, F	SIL 3
G	SIL 4
H	Sistemas de Proteção não Suficientes

IEC 61508 - Determinar Redução de Risco e SIL

		W3	W2	W1
C1		A	-	-
F1	P1	B	A	-
	P2	C	B	A
C2	P1	D	C	B
F2	P2	E	D	C
	F1	F	E	D
C3	F2	G	F	E
C4		H	G	F

SIL	Falhas inseguras / hora
4	$10^{-9} < \lambda_u < 10^{-8}$
3	$10^{-8} < \lambda_u < 10^{-7}$
2	$10^{-7} < \lambda_u < 10^{-6}$
1	$10^{-6} < \lambda_u < 10^{-5}$

Redução de Risco	*Nível de Integridade de Segurança*
-	Sem Requisitos de Segurança
A	Sem Requisitos Especiais de Segurança
B, C	SIL 1
D	SIL 2
E, F	SIL 3
G	SIL 4
H	Sistemas de Proteção não Suficientes

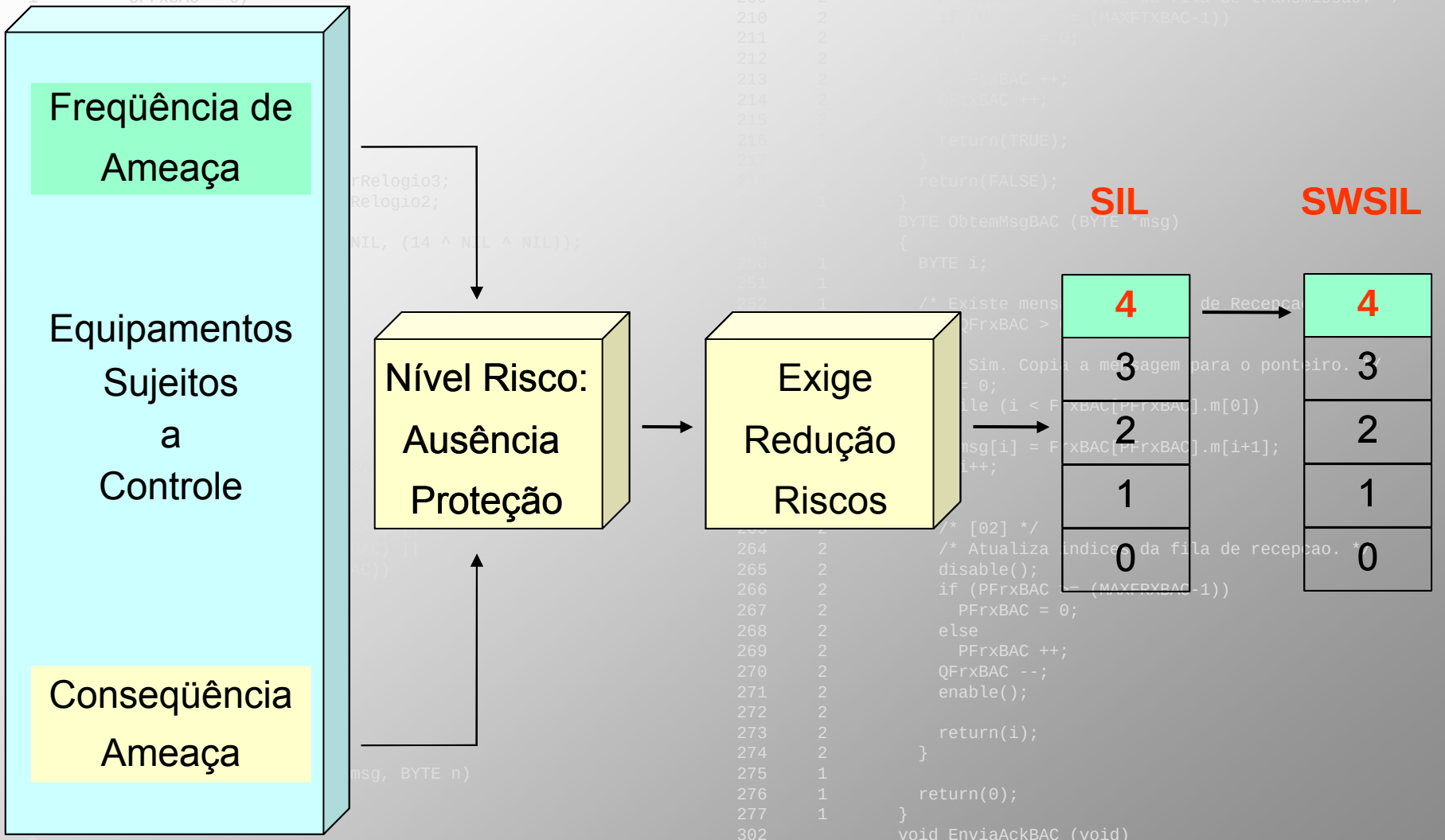
Software & Segurança: EN50128 SWSIL

5.2.2 Os requisitos de nível de integridade de segurança de software (SWSIL) deverão ser decididos com base no nível de risco associado com a **utilização do software no sistema** e no nível de integridade de segurança de software.

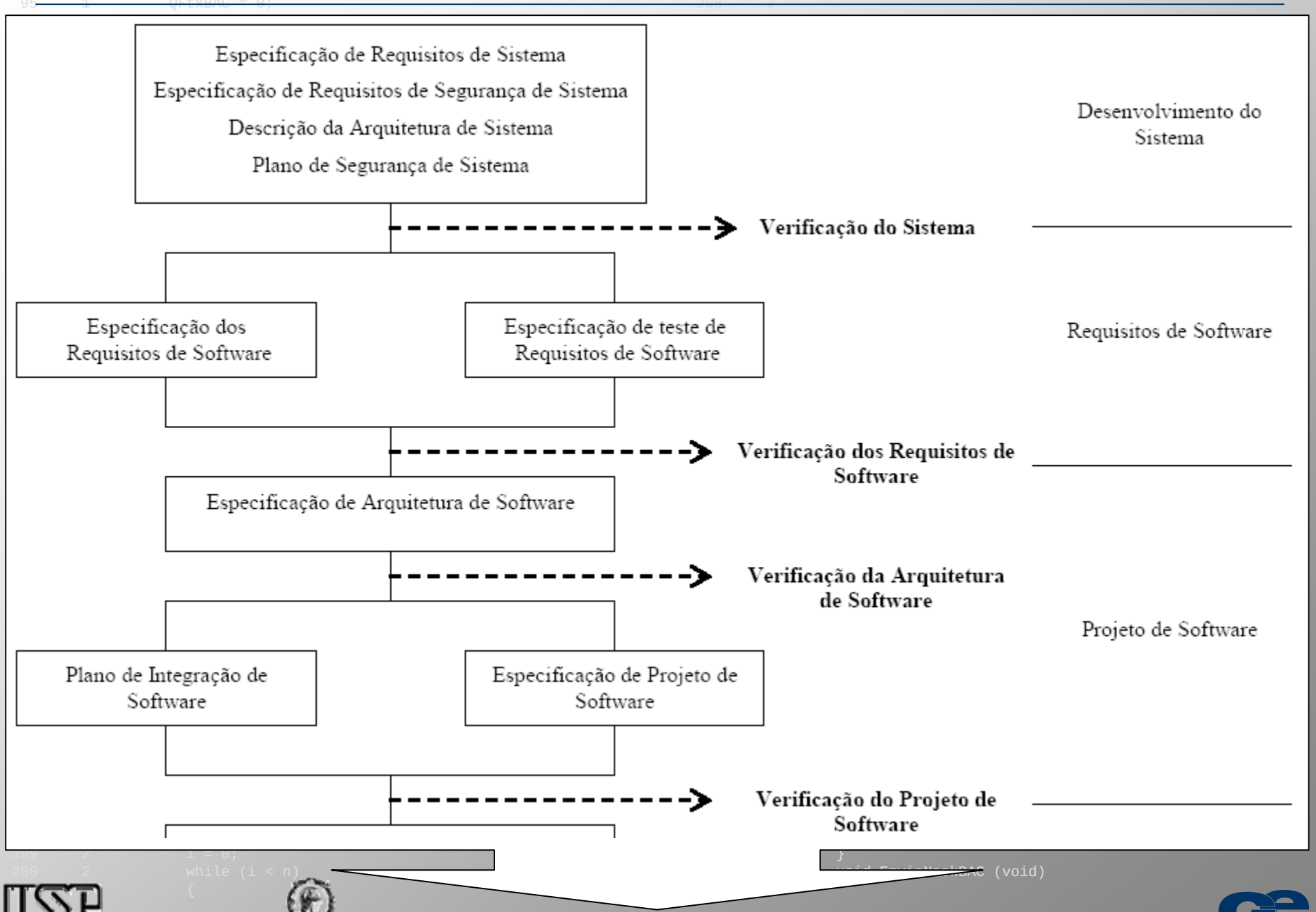
[CENELEC 50128]

Tendo caracterizado o Nível de Integridade de Segurança (SIL) para o Sistema Metro-ferroviário, o Software na área de Sinalização para tal Sistema deverá possuir o mesmo valor.

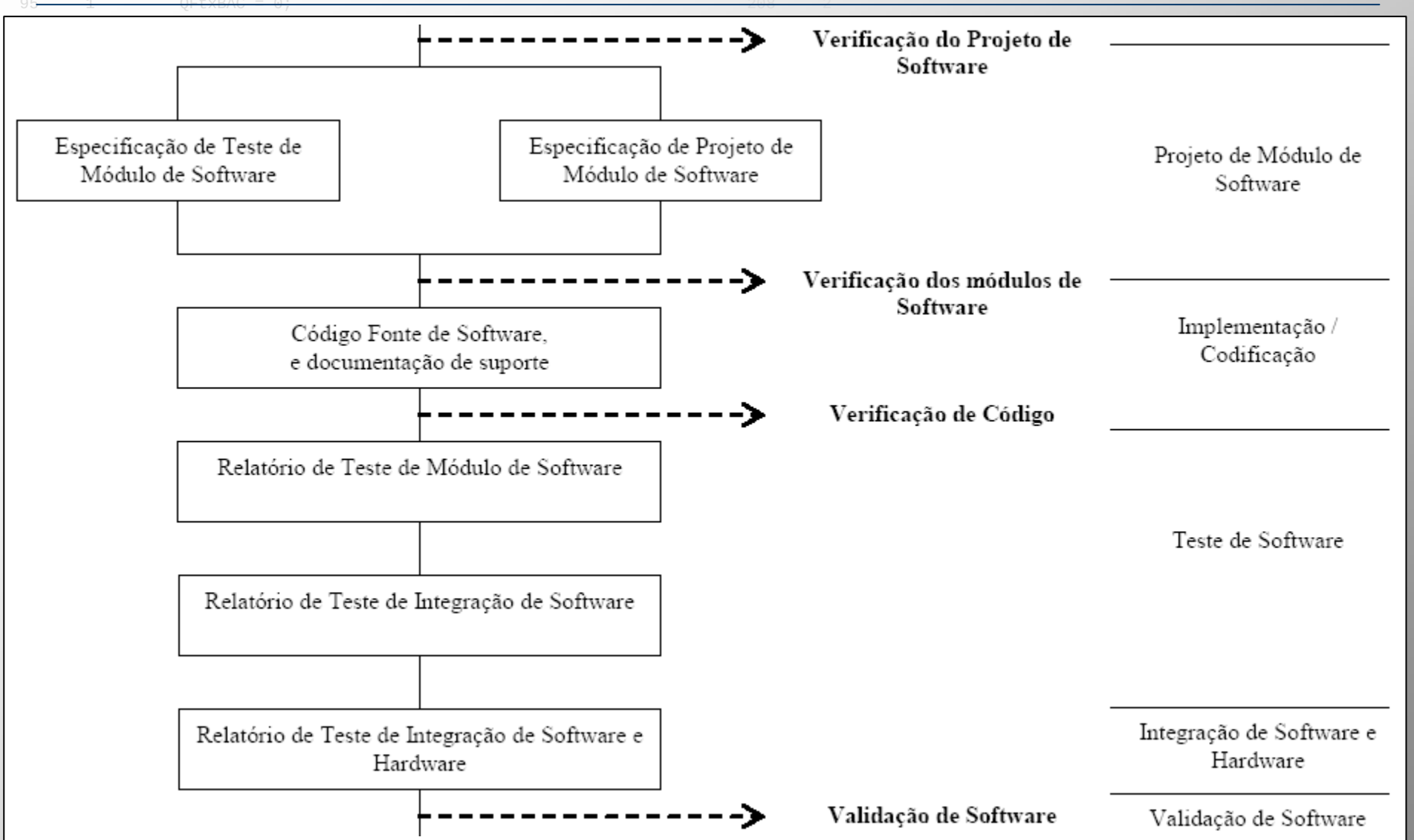
CENELEC 50128



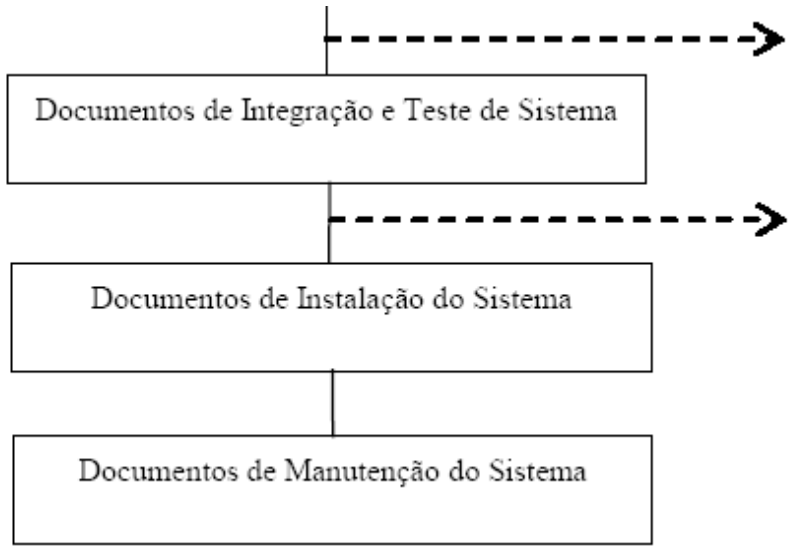
CENELEC 50128 - Ciclo de Vida I



CENELEC 50128 - Ciclo de Vida II



CENELEC 50128 - Ciclo de Vida III



Validação de Software

Validação de Sistema

Validação de Software

Integração de Sistema

Validação do Sistema

Transição para Suporte

Requisito Técnico de Software - Segurança

Os conceitos de “**Fail Safe**” devem ser utilizados no Software relacionado às funções de Segurança Crítica ou Vitais.

Deve atender ao **“Software Safety Integrity Level” (SIL) quatro (4)**, conforme a norma IEC 61508 “Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems”.

Devem ser atendidas as recomendações da norma **CENELEC 50128** “Railway applications – Communications, Signaling and Processing Systems – Software for Railway Control and Protection Systems”.

[EPS 2004]

Emprego de Software em Sinalização - Horizonte

Considerações:

- Uso Intensivo de SW em Tempo Real e Segurança Crítica;
- Fatores de aumento natural do risco;
- Transmissão por meio aberto de Sinais Vitais;
- Determinação de distância por múltiplos sensores;
- Emprego limitado de Hardware fail-safe;

Os **conceitos consagrados** de Segurança de um Sistema de Sinalização e Controle devem ser implementados de forma **completa** e claramente **verificáveis**. [IEEE 1474-1]

O Software com Qualidade:

- Maior Garantia do Nível de Segurança
- Redução do Retrabalho
- Otimização de Recursos
- Controle sobre o Processo
- Controle sobre Tecnologias em Implantação
- Integra Experiência e Capacidade de Inovação
- Incorporação de Iniciativas dos maiores Consumidores de Software mundiais e da Produção Científico-acadêmica na área.

+ + +

“Nada disso é tudo,
Tudo isso é fundamental.”

[H Gessiger, *Engenheiros do Hawaii*]

Christian Becker Bueno de Abreu

cbecker@metrosp.com.br