1. Falha x Causa x Prevenção

Therac-25 (1985–1987)

Falha

Pacientes de radioterapia receberam doses letais de radiação.

Causa

Condições de corrida (race condition) no software de controle, falhas de interface e ausência de intertravamentos de segurança.

prevenção

- Testes rigorosos de software crítico.
- Validação independente de segurança.
- Inclusão de hardware de segurança redundante.
- Adoção de práticas de engenharia de software seguro.

2. Ligação com ISO/IEC 25010 – Qualidade de produto

Característica ISO/IEC 25010	Problema Observado no Caso	Consequência
Confiabilidade (Reliability)	O sistema não foi capaz de lidar com exceções de dados fora do intervalo esperado (overflow na conversão de número de ponto flutuante).	Falha total do sistema de navegação e destruição do foguete segundos após o lançamento.
Manutenibilidade (Maintainability)	Reutilização de código do Ariane 4 sem análise adequada das novas condições operacionais do Ariane 5.	Código legado não revisado causou comportamento inesperado em um novo ambiente.
Segurança (Security)	Ausência de mecanismos de isolamento de falhas entre sistemas redundantes (ambos falharam simultaneamente).	Nenhum sistema pôde assumir o controle quando o principal falhou.
Compatibilidade (Compatibility)	Software herdado não compatível com os parâmetros de voo do novo foguete.	Dados incompatíveis provocaram erro crítico durante a execução.

3. Ligação com CMMI/MPS.BR – Qualidade de processo.

Área de Processo CMMI/MPS.BR	Problema Observado no Caso	Como o Processo Poderia Evitar a Falha
Gerência de Requisitos (REQM)	Requisitos críticos de segurança e intertravamento não foram especificados de forma detalhada.	Especificar requisitos funcionais e de segurança com rastreabilidade completa, garantindo que todos sejam implementados e testados.
Verificação (VER)	Testes de software incompletos, sem simulação de todas as condições de operação.	Implementar processos formais de verificação, incluindo simulações de cenários extremos e testes automatizados.
Validação (VAL)	Falha em validar o software em ambiente real antes da liberação.	Validar o sistema com pacientes simulados ou modelos físicos antes da implantação real.
Gestão de Configuração (CM)	Código legado foi reutilizado sem controle adequado.	Aplicar controle de versões, revisão de código e auditorias formais para garantir integridade e rastreabilidade do software.
Gerência de Riscos (RSKM)	Falhas críticas não foram antecipadas nem mitigadas.	Identificar riscos críticos, implementar planos de mitigação e medidas preventivas para evitar falhas fatais.

4. Importância dos testes e auditorias

Tipo de Teste	Como Ajudaria no Caso Therac-25
Testes de Integração	Detectariam falhas na comunicação entre os módulos de controle e hardware, evitando que comandos incorretos fossem enviados ao acelerador de partículas.
Testes de Regressão	Garantiriam que alterações no software não reintroduzissem erros antigos de controle de dose ou de segurança já corrigidos em versões anteriores.

Testes em Ambiente Simulado	Permitiriam simular tratamentos reais com pacientes virtuais, detectando comportamentos perigosos antes da liberação do sistema.
Testes Automatizados	Executariam rotinas de verificação contínua para identificar falhas críticas de cálculo ou de sequência de operação, reduzindo o risco de erro humano nos testes.
Auditorias de Código e Processos	Revisões independentes teriam identificado a ausência de verificações de segurança e falhas de rastreabilidade, assegurando a conformidade com padrões médicos e regulatórios.