Sistemas Críticos

Centro de Informática - Universidade Federal de Pernambuco
Engenharia da Computação
Kiev Gama
kiev@cin.ufpe.br

Slides originais elaborados por Ian Sommerville e adaptado pelos professores Marcio Cornélio e Kiev Gama

UNIVERSIDADE FEDERAL DE PERNAMBUCO

Sistemas Críticos

Sistemas críticos de segurança

- A falha pode resultar em perda de vida, prejuízo ou dano para o ambiente;
- Sistema de proteção de planta química.

Sistemas críticos de missão

- A falha pode resultar na deficiência de alguma atividade dirigida a metas;
- Sistema de navegação de nave espacial.

Sistemas críticos de negócios

- A falha pode resultar em altas perdas econômicas;
- Sistema de contas de cliente em um banco.

washingtonpost.com > Technology > Special Reports > Cyber-Security

Cyber Incident Blamed for Nuclear Power Plant Shutdown

By Brian Krebs
washingtonpost.com Staff Writer
Thursday, June 5, 2008; 1:46 PM

A nuclear power plant in Georgia was recently forced into an emergency shutdown for 48 hours after a software update was installed on a single computer.

The incident occurred on March 7 at Unit 2 of the Hatch nuclear power plant near Baxley, Georgia. The trouble started after an engineer from Southern Company, which manages the technology operations for the plant, installed a software update on a computer operating on the plant's business network.

http://www.washingtonpost.com/wp-dyn/content/article/2008/06/05/AR2008060501958.html

Software Bug Triggered Airplane Dive Emergency

When an airplane system monitoring Airbus jet's altitude and position output incorrect data, flight computers failed to compensate.

Investigators have released their final report into a 2008 Qantas flight QF72 from Singapore to Perth, Australia, in which 110 people were injured after a computer component failed. Interestingly, investigators have now found that a programming error was partly to blame for the incident.

Here's what happened: On October 7, 2008, aircraft-monitoring systems in the Airbus A330-303--flying at 37,000 feet--failed, causing the autopilot to automatically disconnect. But pilots were still at the mercy of a flight computer that was receiving incorrect data.

Roughly two minutes after the failure of the computer component, the flight computer initiated two deep dives, the first for 20 seconds, the second for 16 seconds. Each dive slammed passengers into ceilings and walls. Dozens of alarms, most of them false, also began sounding in the cockpit. Luckily, pilots were able to switch to fully manual controls and execute an emergency landing at a nearby Australian military base.

http://www.darkreading.com/risk-management/software-bug-triggered-airplane-dive-emergency/d/d-id/1101952?

TECH 8/19/2013 @ 3:50PM | 15.316 views

Amazon.com Goes Down, Loses $66,240 Per Minute

+ Comment Now + Follow Comments

It's been a bad week for ecommerce. On Friday, Google GOOG -1.01% temporarily went dark, causing a 40% drop in web traffic. Today Amazon.com AMZN -0.22% went down for approximately 30 minutes, preventing shoppers from accessing the site via Amazon.com, mobile and Amazon.ca.

http://www.forbes.com/sites/kellyclay/2013/08/19/amazon-com-goes-down-loses-66240-per-minute/

Confiança no sistema

"Dependability"

Para sistemas críticos é, em geral, o caso em que a propriedade de sistema mais importante é a confiança.

A confiança de um sistema reflete o grau de confiança do usuário nesse sistema, bem como a extensão de confiança do usuário que o operará conforme suas expectativas e que não 'falhará' durante o uso normal.

Utilidade e confiança não são a mesma coisa. Um sistema não tem de ser confiável para ser útil.

A importância da confiança

Sistemas que não são confiáveis, inseguros ou desprotegidos, podem ser rejeitados pelos seus usuários.

Os custos com falha de sistema podem ser muito altos.

Sistemas não confiáveis podem causar perda de informação e, conseqüentemente, um alto custo de recuperação.

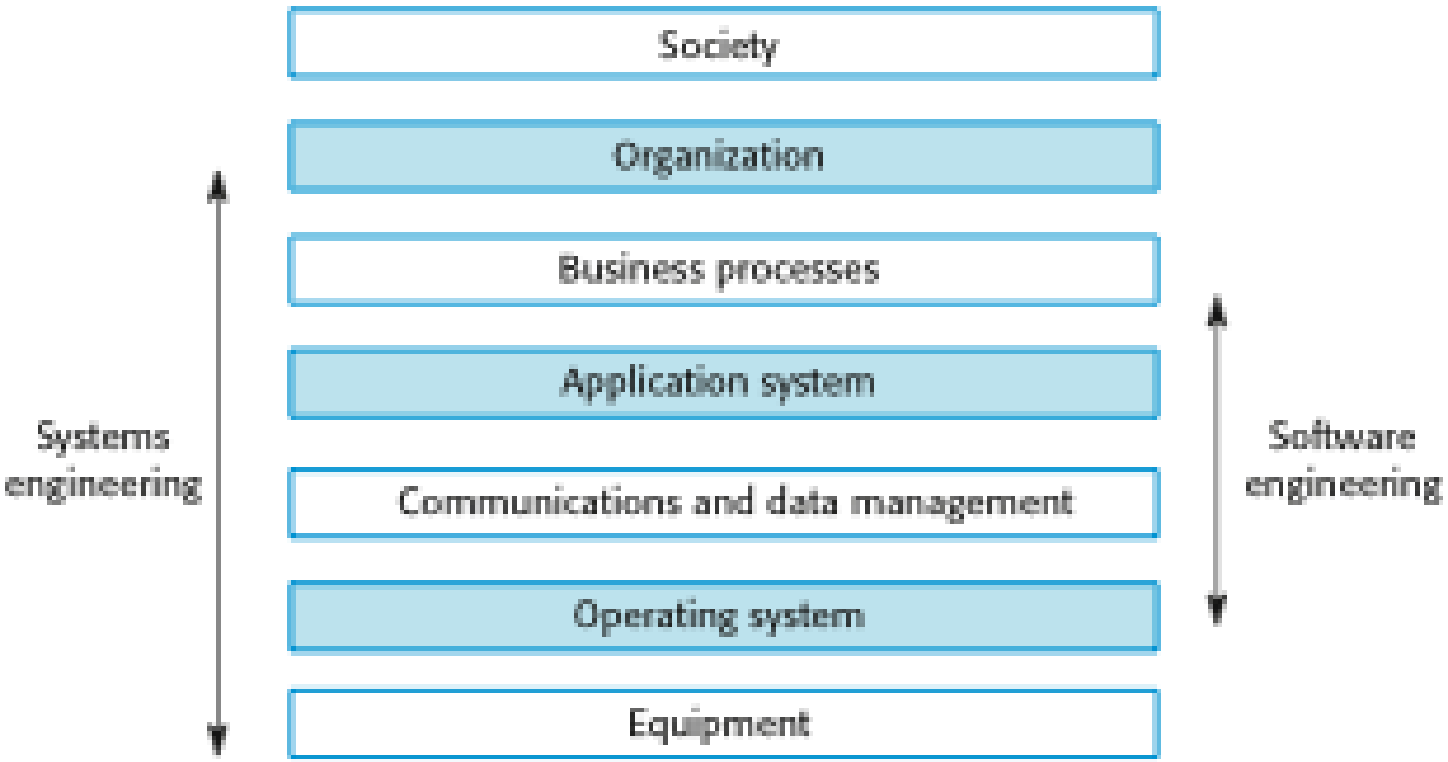
Métodos de desenvolvimento para sistemas críticos

Os custos com falha de sistema crítico são tão altos que os métodos de desenvolvimento podem ser usados, embora não sejam eficazes em termos de custo para outros tipos de sistema.

Exemplos de métodos de desenvolvimento

- Métodos formais de desenvolvimento de software
- Análise estática
- Garantia de qualidade externa

Sistemas sociotécnicos



Sistemas sóciotécnicos críticos

Falhas de hardware

O hardware pode falhar por causa de erros de projeto e de produção, ou pelo fato de os componentes terem atingido o fim de sua vida natural.

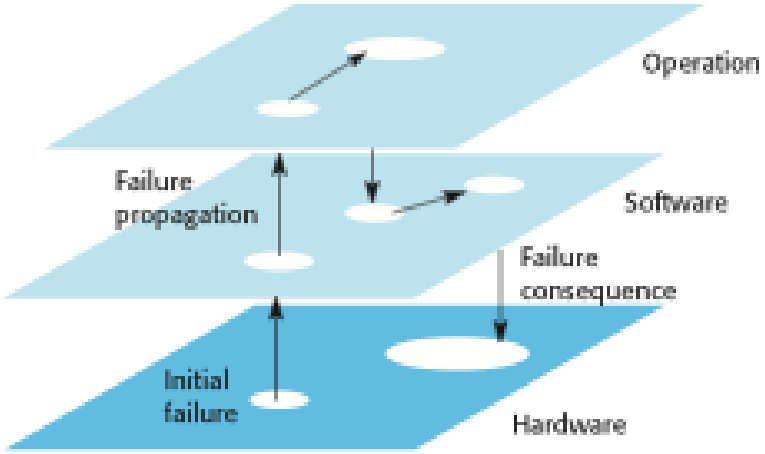
Falhas de software

Software falha devido a erros na sua especificação, projeto ou implementação.

Falha operacional

Operadores humanos cometem erros. Atualmente, talvez sejam a maior causa de falhas de sistema.

Propagação de falhas



Uma bomba de insulina controlada por software

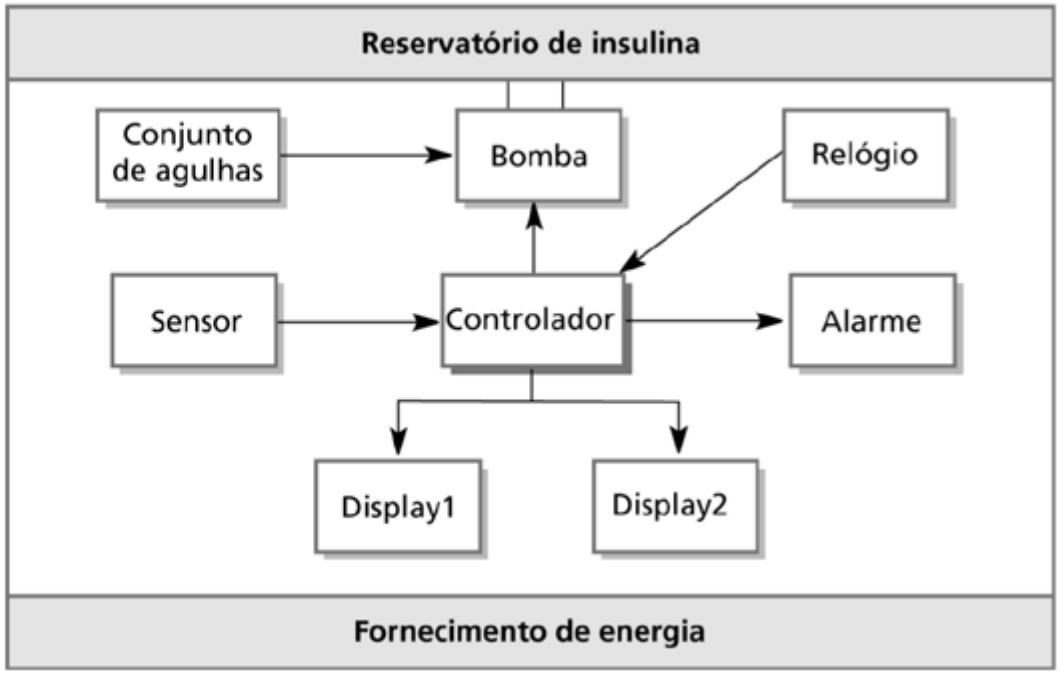
Usada por diabéticos para simular a função do pâncreas, que produz insulina, um hormônio essencial que com a função de metabolizar o açúcar do sangue.

Mede a glicose do sangue (açúcar) usando um micro sensor e calcula a dose de insulina necessária para metabolizar a glicose.

Organização da bomba de insulina

Figura 3.1

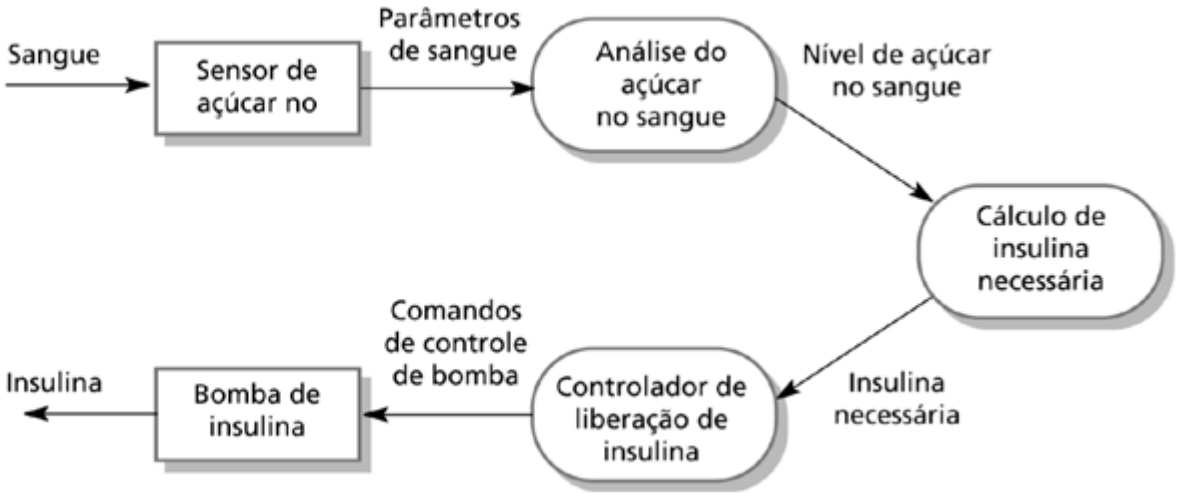
Estrutura de bomba de insulina



Fluxo de dados da bomba de insulina

Figura 3.2

Modelo de fluxo de dados de bomba de insulina



Requisitos de confiança

O sistema deverá estar disponível para liberar insulina quando requisitado.

O sistema apresentará confiabilidade e liberará a quantidade correta de insulina para neutralizar o nível corrente de açúcar no sangue.

O requisito essencial de segurança é que doses excessivas de insulina nunca devem ser liberadas, já que isso é, potencialmente, uma ameaça de vida.

Confiança

A confiança em um sistema equivale ao seu merecimento de confiança.

Um sistema confiável é aquele em que os usuários depositam sua confiança.

Figura 3.3

Dimensões de confiança

Outras propriedades de confiança

- Facilidade de reparo
  - Reflete a amplitude em que o sistema pode ser reparado no caso de uma eventual falha;
- Facilidade de manutenção
  - Reflete a amplitude em que o sistema pode ser adaptado para novos requisitos;
- Capacidade de sobrevivência
  - Reflete a amplitude na qual o sistema pode fornecer serviços quando está sob ataque hostil;
- Tolerância a erros
  - Reflete a amplitude na qual os erros de entrada de usuário podem ser evitados e tolerados.

Confiança no sistema de bomba de insulina

Disponibilidade

- Funciona quando solicitado

Confiabilidade

- Dose correta de insulina

Segurança

- Não libera uma dose perigosa de insulina

Custos de confiança

Custos de confiança tendem a aumentar exponencialmente quando níveis cada vez mais altos de confiança são requisitados.

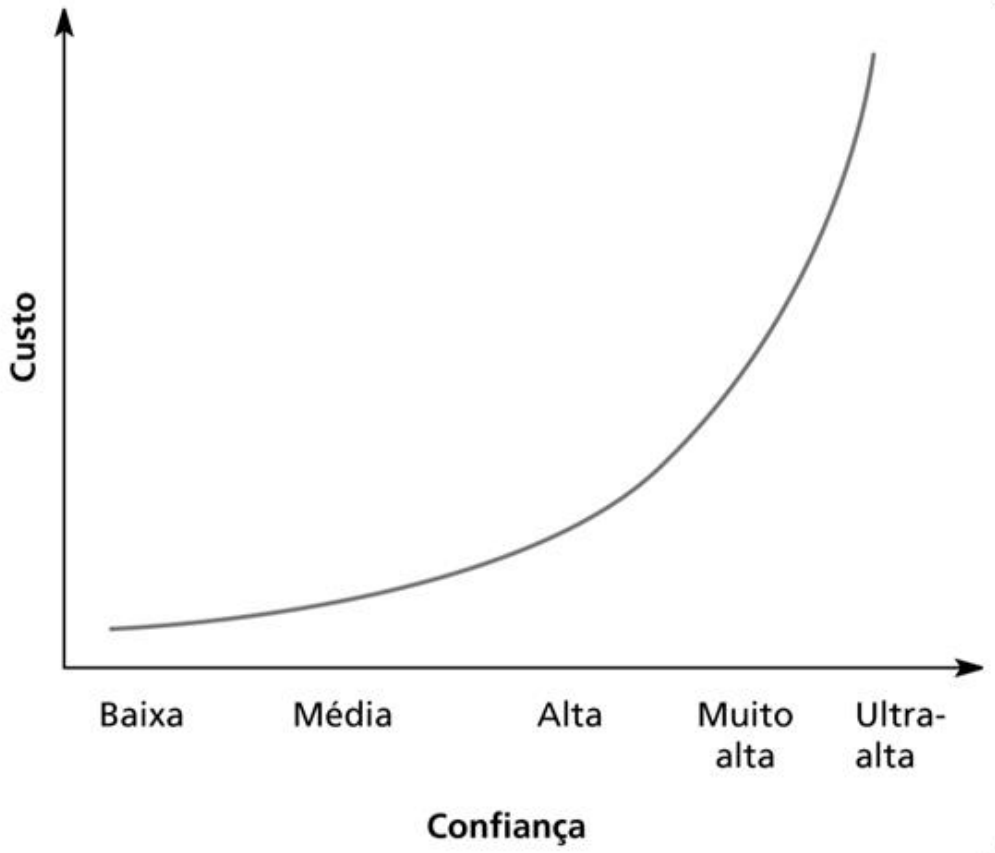
Existem duas razões para isso:

- O uso de técnicas de desenvolvimento mais onerosas e de hardware que são requisitados para atingir os níveis mais altos de confiança
- O aumento de testes e a validação de sistema que é necessária para convencer o cliente do sistema de que os níveis de confiança requisitados foram atingidos.

Custos de aumento de confiança

Figura 3.4

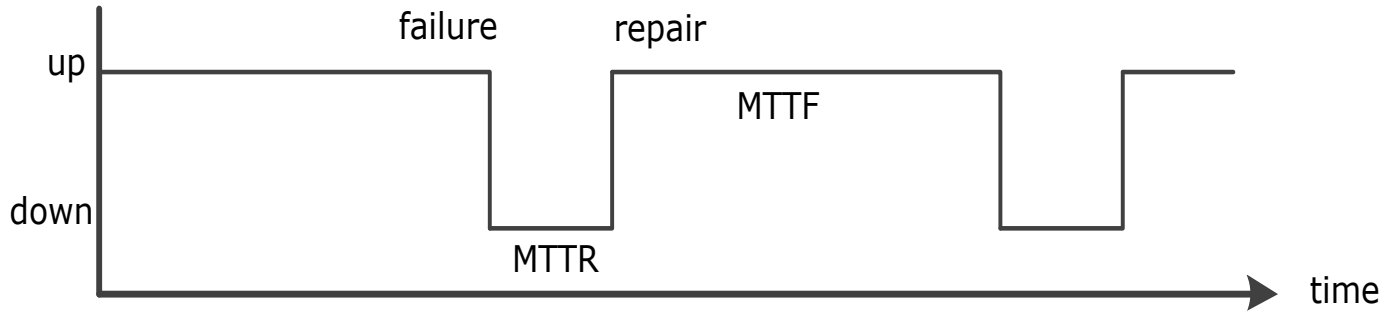
Curva custo/confiança



Disponibilidade e confiabilidade

- Confiabilidade
  - Probabilidade de operação de sistema livre de falha durante um período especificado, em um dado ambiente para um objetivo específico.
  - Serviços fornecidos corretamente (consistente com especificação)
- Disponibilidade
  - Probabilidade que um sistema, em um determinado instante, estará operacional e será capaz de fornecer os serviços requisitados.
- Ambos os atributos podem ser expressos quantitativamente.

Disponibilidade e confiabilidade



$$MTBF = MTTF + MTTR$$

$$A = \frac{MTTF}{MTTF + MTTR}$$

MTTF – Mean Time To Failure
MTBF – Mean Time Between Failure
MTTR – Mean Time To Repair

Disponibilidade e confiabilidade

Algumas vezes é possível incluir a disponibilidade de sistema sob a confiabilidade de sistema

- Obviamente, se um sistema está indisponível, não está fornecendo os serviços de sistema especificados.

É possível haver sistemas com baixa confiabilidade disponíveis.

- Contanto que as falhas de sistema sejam rapidamente reparadas e não causem danos aos dados, a baixa confiabilidade talvez não seja um problema.

A disponibilidade leva em conta o tempo de reparo.

Disponibilidade

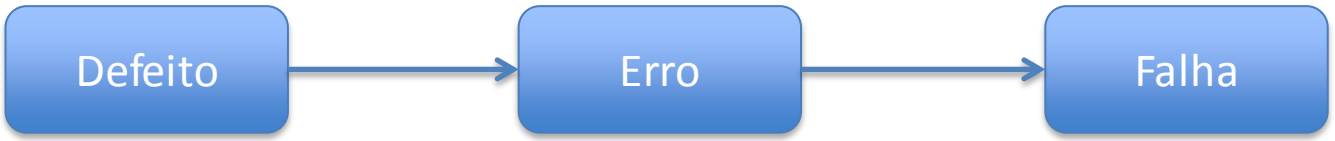
Class	System type	Availability	Unavailability (min/year)
1	Unmanaged	90%	52560
2	Managed	99	5256
3	Well-managed	99.9	526
4	Fault-tolerant	99.99	52
5	High-availability	99.999	5
6	Very-high-availability	99.9999	0.5
7	Ultra-high-availability	99.99999	0.05

Gray, J., Reuter, A. Transaction Processing: Concepts and Techniques. Morgan Kaufman, 1993

Terminologia de confiabilidade

Tabela 3.1 Terminologia de confiabilidade.

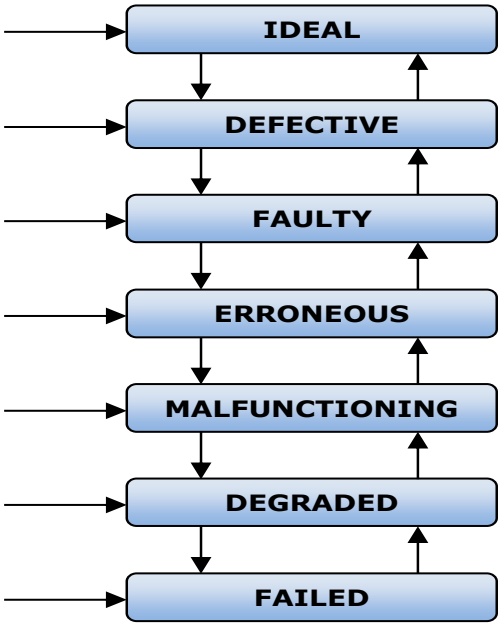
Termo	Descrição
Falha de sistema	Um evento que ocorre em algum momento, quando o sistema não fornece um serviço conforme esperado por seus usuários.
Erro de sistema	Um estado errôneo de sistema que pode levá-lo a um comportamento inesperado pelos seus usuários.
Defeito de sistema	Uma característica do sistema de software que pode levar a um erro de sistema. Por exemplo, a falha em iniciar uma variável pode levar a um valor errado quando esta for usada.
Erro humano ou engano	Comportamento humano que resulta na introdução de defeitos em um sistema.



Relação de causalidade entre as diferentes ameaças à confiança (dependability)

Avižienis, A., Laprie, J., Randell, B., and Landwehr, C. 2004. Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing 1, 1 (Jan. 2004), 11-33

Níveis de confiança (Parhami, 1997)



Parhami, B. Defect, Fault, Error, . . . , or Failure. IEEE Transactions on Reliability, December 1997, pp. 450—45

Defeitos e falhas

Falhas são, em geral, resultado de erros derivados de defeitos no sistema.

No entanto, os defeitos não resultam necessariamente em erros no sistema

- O estado defeituoso do sistema pode ser transitório, e 'corrigido' antes do aparecimento de erros.

Os erros não conduzem, necessariamente, a falhas de sistema

- O erro pode ser corrigido por detecção de erros *built-in* e pela recuperação

Realização de confiabilidade

Prevenção de defeitos

- Técnicas de desenvolvimento são usadas para minimizar a possibilidade de erros e/ou detectá-los antes que causem a introdução de defeitos no sistema (ex. evitar LP que tenha manipulação de ponteiros)

Detecção e remoção de defeitos

- O uso de técnicas de verificação e validação que aumentam a probabilidade de detecção e correção (teste e depuração)

Tolerância a defeitos

- Técnicas de *run-time* são usados para assegurar que defeitos não resultem em erros de sistema e/ou que esses erros não resultem em falhas.

Modelagem de confiabilidade

Você pode modelar um sistema como um mapeamento de entradas-saídas, onde algumas entradas resultarão em saídas errôneas.

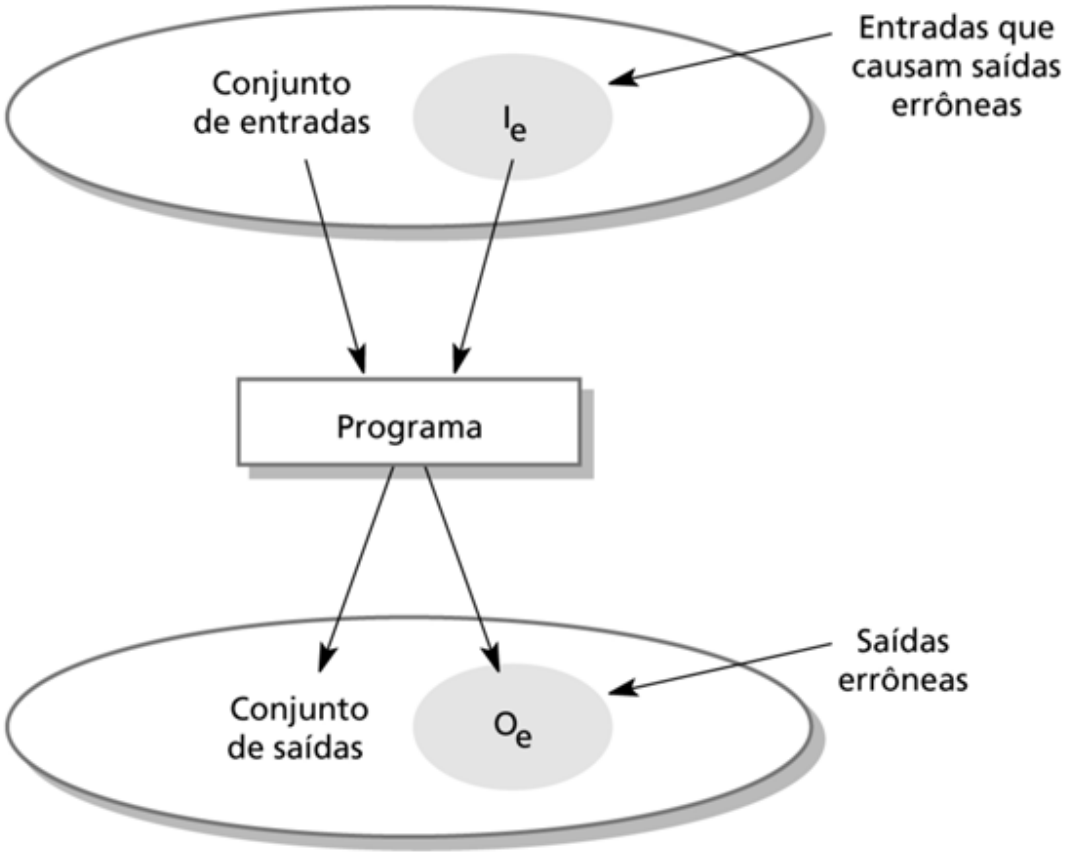
A confiabilidade do sistema é a probabilidade de uma entrada particular pertencer a um conjunto de entradas que causam saídas errôneas.

Pessoas diferentes usarão o sistema de maneiras diferentes e, sendo assim, essa probabilidade não é um atributo estático de sistema, mas sim, dependente do ambiente do sistema.

Mapeamento entradas/saídas

Figura 3.5

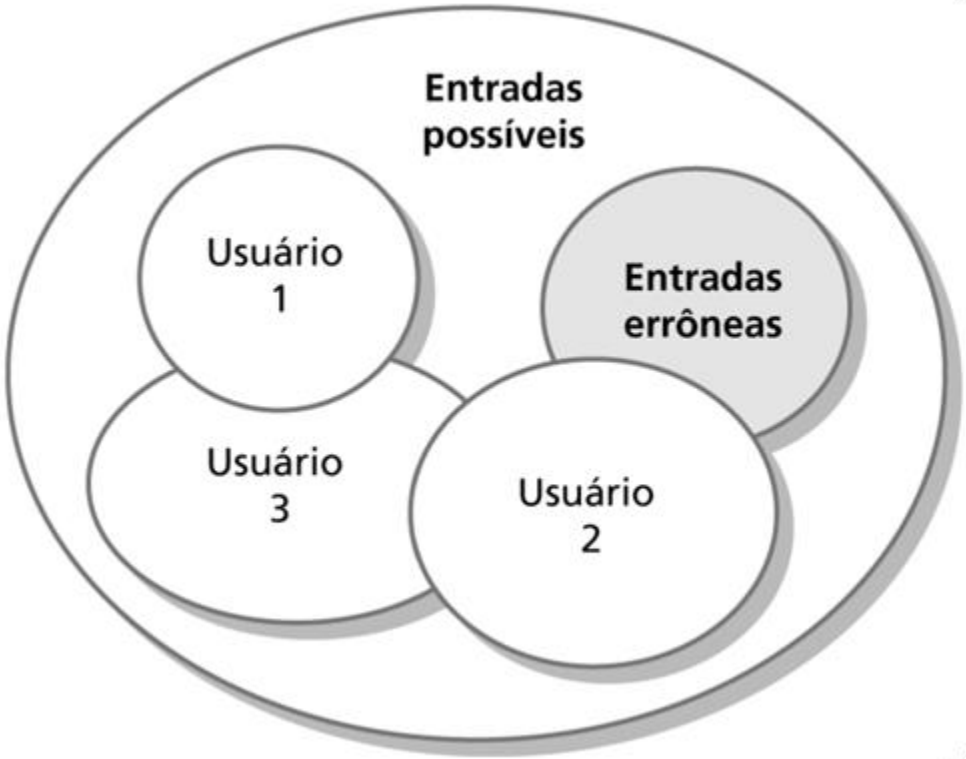
Sistema de mapeamento de entrada/saída



Percepção de confiabilidade

Figura 3.6

Padrões de uso de software



Melhoria de confiabilidade

A remoção de X% de defeitos de um sistema não melhorará, necessariamente, a confiabilidade em X%. Um estudo da IBM mostrou que a remoção de 60% de defeitos de produtos resultaram em uma melhoria de 3% da confiabilidade.

Defeitos de programa podem estar em seções raramente executadas do código e, desse modo, nunca serem encontrados pelos usuários. A remoção destes defeitos não afetam a percepção da confiabilidade.

Um programa com defeitos conhecidos podem, portanto, ainda ser vistos como confiáveis pelos seus usuários.

Segurança

Propriedade do sistema que reflete a habilidade do sistema para operar, normalmente ou não (pode falhar), sem perigo de causar **prejuízo ou morte a pessoas** e sem **danos para o ambiente**

É cada vez mais importante considerar a segurança do software à medida que **mais e mais sistemas incorporam sistemas de controle** baseados em software.

Requisitos de segurança são requisitos exclusivos, isto é, **excluem situações indesejáveis** ao invés de especificar serviços de sistema requisitados.

Aspectos críticos de segurança

Sistemas críticos primários de segurança

Sistemas de software embutidos cuja falha pode causar a falha do hardware associado e ameaçar diretamente as pessoas.

Sistemas críticos secundários de segurança

Sistemas cuja falha resulta em defeitos em outros sistemas que podem ameaçar as pessoas.

Segurança e confiabilidade

Segurança e confiabilidade estão relacionadas, mas são distintas

Em geral, confiabilidade e disponibilidade são condições necessárias, porém, não suficientes para a segurança de sistemas.

A confiabilidade está relacionada à conformidade com uma dada especificação e com o fornecimento de serviço.

A segurança está relacionada à garantia de que sistemas não podem causar danos, independente de estarem ou não de acordo com sua especificação.

Sistemas confiáveis inseguros

Erros de especificação

Se a especificação de sistema está incorreta, o sistema pode se comportar conforme especificado e, no entanto, causar acidentes.

Falhas de hardware gerando entradas espúrias

Difíceis de prever na especificação.

Comandos sensíveis ao contexto, isto é, emitindo um comando certo no momento errado

Em geral, é o resultado de um erro de operador.

Terminologia de segurança

Tabela 3.2 Terminologia de segurança.

Termo	Descrição
Acidente (ou desgraça)	Evento ou seqüência de eventos não planejados que resulta em morte ou ferimento de humanos, danos à propriedade ou ao ambiente. Uma máquina controlada por computador que fere seu operador é um exemplo de um acidente.
Perigo	Condição com potencial para causar ou contribuir para um acidente. A falha de um sensor que detecta um obstáculo em frente de uma máquina é um exemplo de perigo.
Dano	Medida de perda resultante de um acidente. Um dano pode variar desde a morte de várias pessoas como resultado de um acidente até ferimentos de pouca importância ou danos à propriedade.
Severidade do perigo	Avaliação do pior dano possível que poderia resultar de determinado perigo. A severidade do perigo pode variar de catastrófica, na qual várias pessoas são mortas, até somente danos de pouca importância.
Probabilidade de perigos	Probabilidade de ocorrência de eventos que criam um risco. Valores de probabilidade tendem a ser arbitrários, mas variam de *provável* (digamos, chance de 1/100 de ocorrência de um risco) a *implausível* (não existem situações concebíveis nas quais o perigo possa ocorrer).
Risco	É a medida da probabilidade de que o sistema causará um acidente. O risco é avaliado considerando-se a probabilidade do perigo, a severidade do perigo e a probabilidade de que o perigo resultará em um acidente.

Realização de segurança

Prevenção de perigos

O sistema é projetado de tal modo que algumas classes de perigos sejam evitadas. Ex: uso de dois botões separados em um sistema de corte

Detecção e remoção de perigos

O sistema é projetado de tal forma que os perigos são detectados e removidos antes de causarem um acidente. Ex: pressão excessiva e uso de válvula de escape para evitar explosão

Limitação de danos

O sistema inclui recursos de proteção que minimizem os danos resultantes de um acidente.

Proteção

A proteção é uma propriedade que reflete a habilidade do sistema de se **proteger de um ataque externo** acidental ou deliberado.

A proteção está se tornando cada vez mais importante à medida que os sistemas são colocados em rede e, desse modo, o acesso externo ao sistema por meio da Internet é possível.

Proteção é um **pré-requisito essencial** para disponibilidade, confiabilidade e segurança.

Proteção fundamental

Se um sistema opera em rede e é inseguro, então, as declarações sobre sua confiabilidade e segurança não são confiáveis.

Essas declarações dependem da execução do sistema e de que o sistema desenvolvido seja o mesmo. No entanto, uma intrusão pode mudar a execução do sistema e/ou seus dados.

Portanto, a confiabilidade e a garantia de segurança não são mais válidas.

Danos de falta de proteção

Recusa de serviço

O sistema é forçado para um estado em que serviços normais tornam-se indisponíveis, ou então, a provisão de serviços é significativamente degradada.

Corrupção de programa ou dados

Os programas ou dados de um sistema podem ser alterados de uma maneira não autorizada.

Abertura de informação confidencial

A informação gerenciada pelo sistema pode ser exposta a pessoas que não estão autorizadas a ler ou usar essa informação.

Terminologia de proteção

Tabela 3.3 Terminologia de proteção.

Termo	Descrição
Exposição	Possível perda ou dano no sistema computacional. Pode ser perda ou danos nos dados ou pode ser perda de tempo ou esforço, se a recuperação é necessária após uma brecha na proteção.
Vulnerabilidade	Uma fraqueza no sistema baseado em computador que pode ser explorada para causar perda ou dano.
Ataque	Uma exploração da vulnerabilidade do sistema. Geralmente parte de fora do sistema e é uma tentativa deliberada para causar algum dano.
Ameaças	Circunstâncias que têm potencial para causar perda ou dano. Você pode pensar nelas como uma vulnerabilidade do sistema que está sujeita a um ataque.
Controle	Uma medida de proteção que reduz uma vulnerabilidade do sistema. Criptografia pode ser um exemplo de controle que reduz a vulnerabilidade de um sistema fraco de controle de acesso.

Garantia de proteção

Prevenção de vulnerabilidade

O sistema é projetado de maneira que as vulnerabilidades não ocorram. Por exemplo, se não há uma conexão de rede externa, um ataque externo é impossível.

Detecção e neutralização de ataque

O sistema é projetado de tal modo que ataques à vulnerabilidades são detectados e neutralizados antes de causarem uma exposição. Um exemplo disso são os verificadores de vírus, que encontram e removem os vírus antes de infectarem um sistema.

Limitação de exposição

O sistema é projetado de tal modo que as consequências adversas de um ataque bem sucedido sejam minimizadas. Por exemplo, uma política de *backup* permite que informações danificadas sejam restauradas.

Bibliografia

- Sommerville, Ian. Engenharia de Software, 9ª edição. Pearson Education