

[< VOLTAR](#)

Desenvolvimento de sistemas críticos: implicações e cuidados no processo de desenvolvimento de software

O objetivo desta aula é definir e apresentar os principais conceitos relacionados a sistemas críticos.

NESTE TÓPICO

- > O que são sistemas críticos?
- > O que são softwares de sistemas críticos?
- > Engenharia de software para sistemas críticos
- > Referências



O que são sistemas críticos?



Sistemas críticos são sistemas que se falharem poderão ter impacto dramático sobre a vida humana, sobre o ambiente ou a ativos significativos.

Tais sistemas devem satisfazer a uma variedade de critérios de qualidade específicos, tais como confiabilidade, disponibilidade, segurança e proteção.

Com a inserção cada vez maior de computadores e softwares em todos os aspectos do mundo moderno, sistemas críticos dependem cada vez mais da funcionalidade do software.

Sistemas de controle de tráfego aéreo, sistemas de controle e monitoração de usinas nucleares, sistemas metroviários, sistemas de semáforos, sistemas médicos de tratamento de tumores, sistemas que controlam ABS em automóveis, são exemplos de sistemas críticos.

O que são softwares de sistemas críticos?

- São softwares que implementam processos de tomada de decisões críticas.
- São softwares que controlam e monitoram funções críticas de segurança.

- São softwares que intervêm quando uma condição insegura está presente ou na iminência de acontecer.
- São softwares que interferem em sistemas que executam software crítico de segurança.
- São softwares que tratam dados de segurança críticos incluindo a exibição de informações críticas de segurança.
- São softwares usados para verificar e validar softwares críticos de segurança.

Engenharia de software para sistemas críticos

Engenharia de software para sistemas críticos é uma área particularmente complexa. Há três aspectos que podem ser aplicados para auxiliar no desenvolvimento de sistemas críticos.

- O primeiro aspecto é o processo de engenharia e gerenciamento.
- O segundo aspecto está relacionado a selecionar as ferramentas e ambiente apropriados para desenvolver o sistema, o que permitirá que o desenvolvedor teste efetivamente o sistema.
- O terceiro aspecto envolve abordar todos os requisitos legais e regulamentares pertinentes ao domínio da aplicação em questão.

Seguir os padrões estabelecidos para sistemas críticos de determinada área, força o desenvolvedor a se ater aos requisitos. A indústria aeronáutica, nuclear e espacial, por exemplo, tem vários métodos e padrões estabelecidos e que devem ser seguidos no desenvolvimento de software para essas áreas.

A abordagem padrão é codificar cuidadosamente, inspecionar, documentar, testar, verificar e analisar o sistema que está sendo desenvolvido.

Um exemplo de paradigma para o desenvolvimento de sistemas críticos é o denominado "Métodos Formais". Nessa abordagem, todo o desenvolvimento de software é feito em cima de representações e formalismo matemático.

Métodos formais é uma abordagem que promete software livre de defeitos. Mas tem algumas desvantagens, como a demora para desenvolver o software, a dificuldade na comunicação com o cliente, já que exige um conhecimento da complexa abordagem matemática utilizada, além de exigir um rigoroso treinamento da equipe de desenvolvedores.

O objetivo das abordagens associadas ao desenvolvimento de sistemas críticos é melhorar a qualidade do software por meio de testes e/ou eliminação de passos manuais no processo de desenvolvimento, principalmente porque pessoas cometem erros, e esse tipo de erro (falha humana) é a causa mais comum de potenciais erros fatais.

Agora que você já estudou esta aula, resolva os exercícios e verifique seu conhecimento. Caso fique alguma dúvida, leve a questão ao Fórum e divida com seus colegas e professor.

EXERCÍCIO (https://ead.uninove.br/ead/disciplinas/impressos/_g/pdsoft80_100/a20ex)

Referências

PRESSMAN, R. S. *Engenharia de software*. 7. ed. São Paulo: McGraw-Hill, 2010.

SOMMERVILLE, Ian. *Engenharia de software*. São Paulo: Addison-Wesley, 2007.



Avalie este tópico



ANTERIOR

Definição e conceituação de ferramentas
Biblioteca
CASE: ferramentas para automatizar o processo de desenvolvimento de software
([https://www.uninove.br/conheca-](https://www.uninove.br/conheca-uninove/biblioteca/sobre-a-biblioteca/apresentacao/)



Índice

uninove/biblioteca/sobre-a-biblioteca/apresentacao/)
Portal Uninove
(<http://www.uninove.br>)
Mapa do Site

© Todos os direitos reservados

Ajuda?
(<https://ava.uninove.br/ava/ajuda/>)

