

MO409/MC926 – Introdução à Engenharia de Software

1º Semestre/2012

Prof.: Eliane Martins

Horário da disciplina: 2ª e 4ª das 14h-16h

Local: IC-316

Onde me achar:

eliane_at_ic.unicamp.br

Pavilhão do IC - sala 36

Horário de atendimento:

Marcar por email.

Apresentação

Conforme assinala o documento “Grandes Desafios da Pesquisa em Computação no Brasil”, editado pela SBC em 2006, sistemas de software são cada vez mais *onivalentes*. Em outros termos, estamos cercados por sistemas computacionais que têm grande influência na nossa vida diária, pois controlam a distribuição de energia, transportes, comunicações, entre outros. Sistemas embarcados, interconectados por redes sem fio, tornam possível o controle distribuídos de sistemas automotivos, aviônicos, e saúde, com as redes de área corporal. A dependabilidade (de *dependability*, em inglês) é uma propriedade que vem ganhando cada vez mais importância para sistemas computacionais. Essa propriedade define a capacidade dos sistemas computacionais de prestar um serviço em que se pode justificadamente confiar. Entre os atributos da dependabilidade podemos citar a confiabilidade, disponibilidade, segurança operacional (*safety*), segurança da informação (*security*), entre outras. Sistemas com requisitos de dependabilidade devem ser desenvolvidos cuidadosamente. Em especial, deve ser garantido, ao longo do desenvolvimento, que as propriedades desejadas são efetivamente satisfeitas pelo sistema.

Uma parte da dificuldade no desenvolvimento de sistemas críticos é o conflito entre custo e garantia de que a qualidade desejada está sendo atendida. Quando os métodos para garantir a qualidade requerem alto custo de treinamento de pessoal e alto esforço para seu uso, esses métodos acabam sendo evitados.

A UML (Unified Modeling Language) pode oferecer uma boa relação custo-benefício no desenvolvimento de sistemas com requisitos de disponibilidade, dado que suas notações são bastante utilizadas na indústria. Além disso, existe uma grande quantidade de ferramentas, comerciais ou livres, que dão apoio ao desenvolvimento de sistemas usando UML.

O objetivo da disciplina é estudar e utilizar formas de se desenvolver sistemas com requisitos de dependabilidade. Será visto o desenvolvimento dirigido por modelos utilizando a UML. A disciplina também abordará formas de determinar se o sistema realmente apresenta os atributos de dependabilidade desejados. Os métodos e técnicas estudados serão aplicados em estudos de caso.

A disciplina não aborda a codificação de sistemas, que pode ser coberta em cursos de algoritmos ou de linguagens de programação. O projeto de bancos de dados e de interfaces com usuários também não serão cobertos, por ser tópicos de disciplinas oferecidas pelo programa de pós-graduação do IC, o mesmo acontecendo com projeto de sistemas distribuídos. Aspectos de planejamento também não serão cobertos.

Programa da disciplina

Apresentação do curso

- Descrição do programa

Noção de dependabilidade

- Sistemas críticos
- Terminologia e conceitos
- Porquê ocorrem defeitos? Qual sua consequência?

Introdução à Engenharia de Software

- Engenharia de software: princípios
- Atividades de apoio: Controle de Configuração, Controle de Qualidade, Verificação e Validação, auditorias
- Verificação e validação: conceito e classificação das abordagens
- Processos de software

Desenvolvimento dirigido por modelos

- Elementos de MDA e modelos executáveis
- MDA e ciclo de vida do software

Requisitos

- Requisitos de software: funcionais e não-funcionais, requisitos do usuário, requisitos do sistema
- Casos de uso e cenários

Análise

- Modelagem de domínio
- UML:
 - modelo estrutural: diagrama de classes
 - modelos de comportamento: Diagramas de Interação, Diagrama de Atividades
 - Modelos de comportamento: Diagrama de Estados
- Validação da modelagem

Arquitetura de software

- Da Análise para o Projeto
- Projeto da arquitetura

Projeto de software

- Projeto OO
- Reutilização de software: padrões de projeto. Linhas de produto. Componentes. Serviços.

Testes de software

- Abordagem estática: revisões técnicas
- Abordagem dinâmica: testes. Classificação das técnicas. Noção de modelo de testes e de cobertura.
- Uso de grafos em testes. Testes baseados em modelos de estado.

Atividades do Curso

O programa será coberto por algumas aulas ministradas pelo professor e de seminários sobre diferentes tópicos, a serem apresentados pelos alunos. Os alunos deverão ler artigos sobre os tópicos dados, analisá-los e apresentá-los em classe. Será dado também um estudo de caso, que servirá para ilustrar os tópicos estudados.

☞ Os trabalhos poderão ser feitos em grupos de no MÁXIMO quatro alunos.

Durante o curso o aluno deverá realizar as seguintes tarefas:

- apresentação de seminários;
- aplicação ao estudo de caso;
- escrita de um relatório sobre a aplicação das técnicas estudadas ao estudo de caso.

Seminários

Os seminários tratarão da aplicação do desenvolvimento baseado em modelos ao projeto utilizado como estudo de caso. Cada seminário está dividido em:

- apresentação: 20 a 40 min
- questões: ~10 min

Os seguintes quesitos serão considerados na avaliação das apresentações: **estruturação** (ordem lógica das transparências, sem saltos), **legibilidade** (transparências pouco “carregadas”), **originalidade** (mínimo de cópia de figuras ou textos de artigos, tutoriais, etc, de outros autores; se houver, dar referencia), **conteúdo** (quão bem o assunto foi coberto e atende aos aspectos solicitados), **referências** (livros, artigos, *sites* consultados), **duração** (dentro do limite de tempo estabelecido).

☞ **As transparências devem ser postadas no Teleduc no dia da apresentação. Veja a seguir os descontos por atraso.**

Os seminários servirão para mostrar a evolução do desenvolvimento do sistema escolhido como estudo de caso. Para a modelagem do sistema será utilizado o perfil SysML da OMG. Cada grupo deverá apresentar mais do que um seminário, referentes à aplicação desse perfil na modelagem do estudo de caso.

Os detalhes sobre os seminários serão divulgados no Teleduc.

Estudo de caso

Para aprender como os conceitos de desenvolvimento baseado em modelos para sistemas críticos devem ser utilizados, ao longo do curso será utilizado um estudo de caso. Ao final, será escrito um relatório, mostrando as etapas do desenvolvimento. Será utilizado o perfil SysML, da OMG, para o desenvolvimento. Ferramentas de apoio ao desenvolvimento e testes serão usadas durante o curso.

Os projetos a serem usados como estudo de caso serão disponibilizados no Teleduc. Os estudos de caso cobrirão diferentes aspectos de dependabilidade:

- Segurança operacional (safety)
- Segurança (security)
- Tempo real
- Sistemas embarcados

Cada grupo deverá escolher UM estudo de caso, dentro de um dos tópicos acima. Mais detalhes sobre o estudo de caso estarão disponíveis no Teleduc.

Escrita de Relatório

Ao final do curso, o grupo deve entregar um texto, relatando a experiência adquirida no desenvolvimento do estudo de caso.

- ☞ O relatório não deve ultrapassar 40 páginas, incluindo figuras, tabelas e anexos. Um modelo encontra-se disponível no Teleduc.
- ☞ As mesmas restrições quanto ao formato aos descontos por atraso, descritos acima, também se aplicam nesse caso.

O relatório deverá conter os tópicos apresentados nos seminários, mostrando o desenvolvimento do sistema. Deve conter também uma análise da aplicação do SysML, mostrando possíveis omissões ou dificuldades no uso do SysML na realização do estudo de caso escolhido, bem como levantar os benefícios com a sua utilização. A pergunta de pesquisa a ser respondida é: “As diretrizes para modelagem propostas pelo SysML foram realmente úteis e aplicáveis, cumprindo o propósito de melhorar a expressividade e facilitar o desenvolvimento do sistema usado como estudo de caso?” Para responder a essa pergunta, use as questões e métricas a seguir:

- **Q1: As recomendações do perfil utilizado foram realmente úteis e necessárias para modelagem do estudo de caso¹?**

Para responder a essa questão, a seguinte métrica pode ser utilizada:

- PRC: Porcentagem de Requisitos Cobertos pela modelagem, e pode ser calculada como:

$$PRC = R_c / R_t$$

- R_c : número de requisitos cobertos
- R_t : número total de requisitos

- **Q2: Os artefatos produzidos atendem ao grau de qualidade desejado?**

Para responder a essa questão, usar como métricas:

- Quantidade de falhas encontradas durante as atividades de verificação e geração de testes
- Estimar alguma medida do atributo de dependabilidade mais importante para o estudo de caso escolhido.

Entregas

Cada grupo deverá disponibilizar o material produzido no Teleduc, que consiste em:

- Arquivo contendo as apresentações dos seminários
- Relatório final

☞ Descontos por atraso: **20%** (até 3 dias), **35%** (de 3 a 7 dias), **70%** (de 7 a 10 dias) e **100%** (acima de 10 dias).

☞ Só serão aceitos arquivos em **formato PDF**. Verifique se o seu arquivo pode ser aberto no Teleduc. Arquivos com formato inválido ou com problemas de leitura serão considerados como não entregues e sujeitos a descontos por atraso.

Análise de artigo (opcional)

Os alunos que não tenham atingido a nota necessária terão um trabalho extra, que valerá como um exame. Esse trabalho consiste em fazer a revisão de um artigo recente (de 2006, no mínimo) sobre um dos temas abordados no curso. Na revisão deve constar:

- Descrição sumária do artigo.
- Qual o problema tratado (motivação do artigo).
- Qual a solução dada.
- Qual a forma de avaliar se a solução realmente atendeu ao esperado
- Pontos fortes do artigo (do que você gostou no artigo?).
- Pontos fracos do artigo (do que você não gostou no artigo?).

Postar o artigo lido e a revisão no Teleduc até a data estipulada.

☞ **Esse trabalho é individual.**

☞ O trabalho não deve ultrapassar 5 páginas.

☞ As mesmas regras quanto a formato e descontos por atraso se aplicam nesse caso.

¹ E.L.G.Alves. “Usando MDA e MDT para modelagem e geração automática de arquiteturas de teste para sistemas de tempo real”. Dissertação de mestrado. UFCG. 2011.

Áreas do curso:

Área para comunicações do curso, disponibilização de trabalhos e aviso em geral:

http://ggte.unicamp.br/~teleduc/cursos/aplic/index.php?cod_curso=30

Inscrições até: 30/3

Pontuação

A média (**M**) do aluno será calculada conforme os pesos indicados na tabela abaixo.

	Peso
Seminários	30%
Relatório	70%

Observações:

1. **Não** serão dados provas ou trabalhos substitutivos.
2. Para as atividades que requeiram apresentação a presença é obrigatória. Caso um membro do grupo não esteja presente, a nota do grupo terá um desconto de **1/n**, onde **n** é o número de membros do grupo.
3. O **trabalho extra** conta como se fosse um exame, ou seja, a nota final do aluno será dada pela média entre a média parcial (Seminários e Relatório) e a nota da revisão.

Avaliação

O conceito final será dado de acordo com a tabela abaixo:

Média Final	Conceito
91-100	A
80-90	B
60-79	C
0-59	D

Conduta ética

- ☞ Os trabalhos deverão ser de autoria do aluno ou do grupo UNICAMENTE. Discussões e troca de idéias com colegas, professor ou assistente são saudáveis e bem vindas, mas a solução final deve ser exclusivamente do autor, ou dos autores, quando for trabalho em grupo.
- ☞ Consultas a fontes externas (Web, artigos e livros) são válidas e altamente recomendadas desde que explicitamente referenciados no trabalho.
- ☞ Qualquer outro tipo de conduta será considerado como plágio, e implicará em pontuação zero no semestre para todos os envolvidos.

Leitura

Livros de referência

Bruce Powel Douglass. *Doing Hard Time: Developing Real-Time Systems with UML, Objects, Frameworks and Patterns*. Addison Wesley. Object Technology Series, 1999.

Ian Sommerville. *Software Engineering*, 8ª edição. Addison Wesley, 2006.

Eduardo Bezerra. *Princípios de Análise e Projeto de Sistemas com UML*. Editora Campus, 2ª edição, 2007.

Neil Storey. *Safety-Critical Computer Systems*. Pearson/Prentice Hall, 1996.

Sobre SysML

Especificação:

<http://www.omg.sysml.org/>

<http://www.sysml.org/specs/>

Tutorial:

<http://www.sysml.org/specs/>

