



Universidade Estadual de Campinas Instituto de Computação

Prof. Dr. Bruno Barbieri de Pontes Cafeo
cafeo@ic.unicamp.br
<https://ic.unicamp.br/~cafeo/>



MC646 - Verificação, Validação e Teste de Software (2024s2)

1 Objetivos da Disciplina

Ao final do curso, o aluno deve ser capaz de:

- Compreender a importância da verificação e validação (V&V) na qualidade do software e os custos associados aos defeitos.
- Aplicar técnicas de verificação estática, como revisões, inspeções e análise estática de código.
- Utilizar técnicas de verificação dinâmica, incluindo execução simbólica e testes caixa branca e caixa preta.
- Implementar e utilizar ferramentas de análise de cobertura e de mutantes para avaliação de testes.
- Realizar testes de diferentes níveis, como testes de unidades, de integração, de sistemas e de aceitação.
- Criar e utilizar componentes de testes, como drivers e dublês.
- Conduzir testes de requisitos de qualidade, como testes de robustez e de segurança.
- Aplicar práticas de testes em métodos ágeis, como TDD (Test-Driven Development) e BDD (Behavior-Driven Development).

2 Horário de Aulas

As aulas ocorrerão nos seguintes dias e horários:

Dia da Semana	Horário	Local	Tipo
Segunda-feira	16h00 - 18h00	CB11	Teórica
Quarta-feira	16h00 - 18h00	CB11	Teórica

3 Atendimento

Atendimento presencial ou via Google Meet agendado com, pelo menos, 24h de antecedência via e-mail. Dias específicos serão combinados com a turma ao longo do semestre.

4 Programa da Disciplina

Os tópicos a serem apresentados no curso incluem:

- Visão geral de Processos de Teste.
- Conceitos básicos de teste: defeito, falha, casos de teste, critérios de teste.
- Objetivos e restrições de VV&T (Verificação, Validação e Teste).
- Atividades de V&V ao longo do ciclo de vida de um produto.
- Inspeção.
- Geração de casos de teste baseado em documentos de requisitos.
- Fases do teste.
- Medidas e Métricas de V&V.

- Técnica Funcional: partições em classes de equivalência, análise do valor limite, combinações.
- Técnica Estrutural: critérios de teste baseado em fluxo de controle, critérios de teste baseado em fluxo de dados.
- Refatoração e testes (Teste de regressão).
- Documentação de estratégias de VV&T e outros artefatos.
- Planejamento de VV&T e planos de testes.

5 Página do Curso

O material da disciplina ficará disponível na sala do Google Classroom. O código de inscrição será enviado pelo professor por e-mail, caso necessário.

6 Avaliações e Composição da Nota

A média da disciplina M será calculada da seguinte forma:

$$M = (0.8 * (E_1 + E_2 + E_3 + E_4 + E_5)) + (0.2 * S)$$

onde E_x são os exercícios, e S é a nota do Seminário. Todas as atividades e projeto final serão realizados em grupos.

Um aluno poderá fazer o exame SOMENTE se a média M estiver entre 2.5 (inclusive) e 4.9 (inclusive). O exame consiste em refazer as atividades avaliativas E_x já corrigidas e que o aluno não obteve sucesso. Então, é sugerido que o aluno refaça apenas atividades em que a nota foi inferior a 5,0. Após entregar as reavaliações, a média será recalculada conforme a equação definida anteriormente e, se o aluno atingir pelo menos 5,0 na média final, este será aprovado e terá 5,0 como teto da nota final. Assim, não há possibilidade de um aluno de exame ter uma nota final maior que 5,0 quando aprovado.

7 Datas das Avaliações

A seguir estão as datas importantes para o curso:

- E_1 : 02/09/2024
- E_2 : 09/09/2024
- E_3 : 23/09/2024
- E_4 : 02/10/2024
- E_5 : 16/10/2024
- S : a partir de 30/10/2024
- Exame Final: 09/12/2024

8 Referências

Algumas das referências consideradas importantes para o cumprimento do conteúdo proposto encontram-se listadas a seguir. O material complementar a ser utilizado será indicado na página da disciplina.

1. Delamaro, M. E.; Maldonado, J. C.; Jino, M. *Introdução ao Teste de Software*. Elsevier, 2016.
2. Myers, G. J.; Sandler, C. *The Art Of Software Testing*. John Wiley & Sons, 2004.
3. Copeland, L. *a Practitioner's Guide To Software Test Design*. Artech House, 2004.
4. Mauro Pezzè, Michal Young. *Teste e Análise de Software: processos, princípios e técnicas*. Tradução: Bernardo Copstein e Flávio Moreira de Oliveira. Bookman, 2008.

5. Paul Ammann and Jeff Offutt. *Introduction to Software Testing*, Cambridge University Press, Cambridge, UK, ISBN 0-52188-038-1, 2008.
6. Robert Binder. *Testing object-oriented systems: models, patterns, and tools*. Editora Addison-Wesley Professional, 2000.

Observações

- Não haverá provas ou trabalhos substitutivos que não constem nesse PDD.
- Qualquer tentativa de fraude nas atividades ou nos trabalhos/projetos implicará em nota final $M = 0$ (zero) para todos os envolvidos.
- As datas estipuladas para a entrega das avaliações encontram-se detalhadas no cronograma da disciplina (Slide 1). **O prazo limite para submissão é invariavelmente até as 23:59 do dia estipulado.**