

FIAP GRADUAÇÃO



TECNOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

COMPLIANCE & QUALITY ASSURANCE

PROF. Me. PAULO SAMPAIO

I CONTEÚDO DO CURSO

EMENTA

Visão conceitual e de mercado da importância e aplicabilidade da **qualidade de SW** na governança de TI e nos projetos de software, para formação de profissionais *Full Stack*.

Projeto integrado de sistemas que atendam expectativas de Governança através de uma arquitetura de solução alinhada com **estratégias de negócio**.

I CONTEÚDO DO CURSO

EMENTA

Aplicação de **métricas** na avaliação quantitativa da qualidade e em estimativas de software. **Modelos de maturidade** e capacidade para desenvolvimento e avaliação da qualidade e em desenvolvimento de software com base em CMMi e MPS-br.

Modelos de inspeção de software, princípios e prática de **testes de software**, desenvolvimento orientado a testes e a comportamento, automação dos testes, casos de teste e **gerenciamento ágil** do processo de testes, integrado com Scrum.

- Desenvolver as competências técnicas sobre o processo de produção de SW, com objetivos de garantir a **qualidade do produto final** e o desempenho adequado do processo de negócio envolvido.
- Conhecer e aplicar controles sobre os processos de produção de SW para administrar adequadamente a eficácia, eficiência e efetividade e gerar um produto de software que cumpra com os **objetivos da qualidade**, atendendo as necessidades de patrocinadores, clientes e equipes que cuidarão da sustentação do produto.

- Adquirir conhecimento fundamental para participar de processos de **auditoria da qualidade** para certificação ou dentro de um programa de melhoria contínua.
- Oferecer os fundamentos preparatórios para **certificações internacionais** de profissionais especializados em testes e qualidade de software.

- Desenvolver o **plano de qualidade** dentro de um planejamento de projetos integrado, de forma a garantir a governança, aplicando **SCRUM** e COBIT.
- Desenhar uma solução com arquitetura tecnológica, de dados e aplicação de software alinhadas com estratégias empresariais, aplicando **TOGAF-ADM**
- Utilizar métricas quantitativas de **Pontos de Função** (APF) para estimar projetos de software.

- Gerenciar o processo produtivo de SW com **métricas** e controle estatístico de processos (aplicação de medidas de centro e dispersão e gráficos de controle, análise de causa e efeito e Pareto).
- Aplicar métricas de avaliação estrutural do modelo de SW que apontem a complexidade ciclomática da lógica da aplicação, acoplamentos, tamanho de código, **documentação interna de SW**.
- Planejar, projetar e aplicar **testes de produto**, cumprindo níveis, tipos e técnicas específicas que atendem os princípios de BDD (Behavior Driven Development) e TDD (Test Driven Development).

- Criar controles e diagnósticos de TI alinhados com as **expectativas de negócio**;
- Desenvolver SW dentro de uma arquitetura que otimize o esforço e entregue soluções alinhadas com objetivos e **metas de negócio**.
- Aumentar o **nível de maturidade** e capacidade em desenvolvimento de software com qualidade;
- Participar de **programas de certificação** da qualidade em software;
- Criar **controles** para gerenciamento de requisitos;
- Entender o processo de gerenciamento de requisitos;
- Entender e identificar as etapas do processo de medição funcional e estrutural do software;
- Calcular tamanho e complexidade de software de acordo com seus **requisitos funcionais**, possibilitando estimativas de resultado de esforço, custo e prazo;

- Diferenciar **níveis de teste**, técnicas e tipos de teste;
- Utilizar técnicas para planejar, aplicar e registrar **resultados de testes**;
- Criar e executar casos, roteiros e **planos de teste**;
- Usar testes para guiar a produção do software;
- Fazer uso de ferramentas de planejamento, controle e automação teste baseadas em software e ferramentas CASE (auxílio computacional a engenharia de software);
- Aplicar de forma integrada as práticas de **gerenciamento de projetos** com as práticas da qualidade.

METODOLOGIA

- ☐ Leitura e Compreensão da Apostila Central.
- ☐ Aulas dinâmicas com desenvolvimento permanente de um projeto de disciplina.
- ☐ Utilização de Vídeos complementares ao conteúdo.
- ☐ Utilização de artigos para estudos de caso.
- ☐ Prática de projetos da disciplina e multidisciplinar.

I AVALIAÇÃO

Checkpoints periódicos com atividades práticas; projetos de desafio multidisciplinar (**CHALLENGE**).



I FALTAS

O aluno deve atender a frequência mínima de **75%** (setenta e cinco por cento) às aulas e demais atividades escolares, caso contrário será **REPROVADO**.

IMPORTANTE

- ✓ O CONTROLE SOBRE O NÚMERO DE FALTAS É RESPONSABILIDADE DO ALUNO.
- ✓ SERÁ COMPUTADA PRESENÇA SOMENTE PARA OS ALUNOS PRESENTES NO HORÁRIO DA CHAMADA.

Unidade I – Conceitos de **Qualidade**, **Governança** em projetos de software e os **Guias e Normas** associados.

- Gestão Estratégica, Tática e Operacional na área de TI - visão geral da abordagem FLEKS e BSC no desdobramento de diretrizes corporativas.
- Governança empresarial nos tempos atuais e a contribuição da qualidade de software e **Governança de TI**
- Resolvendo problemas hot spot: automação de testes com JUNIT
- Governança de projetos de software com **COBIT** (visão geral do modelo)
- Visão geral das normas **ISO**, aplicadas ao processo de software (série 9000, 25010/9126, 12207, 14000, 15504-Spice, 17000).
- TQM, **Six sigma** e o controle estatístico de resultados para gestão da qualidade segundo a ISO
- Guias para desenvolvimento de capacidade de maturidade de processos na produção de software - aplicação do **CMMi** e **MPS.br**

1º semestre

Unidade II – Gestão do processo produtivo de software.

- Arquitetura de soluções: o ciclo **TOGAF-ADM**
- Aplicação prática de desenho de arquitetura com TOGAF e **ARCHIMATE**
- Definição do backlog de produto em projeto **SCRUM**, orientado pela arquitetura inicial de solução
- Planejamento de entregas e previsão de corridas de produção em SCRUM: planejamento de release com AZURE Boards, com uso de **Poker Scrum** na estimativa de esforço
- Planejamento das tarefas de produção de software em SCRUM: planejamento de sprint com AZURE Boards
- Instrumentos de controle de projeto SCRUM: Burndown chat e **KANBAN** Board com AZURE Boards

1º semestre

Unidade III – Métricas de estimativa de complexidade de software.

- Introdução aos modelos de estimativa Diretos e Paramétricos e seus impactos para formação de valor, segundo o [FLEKS](#)
- Medição de tamanho e complexidade funcional de projeto de software
- Ponderação da complexidade não funcional de projetos de software
- Estimativa de esforço, custo de mão-de-obra e prazo de projetos com [APF](#) (análise de pontos de função), aplicado a projetos ágeis com Scrum.
- Uso da APF em substituição à estimativa de esforço com Poker Scrum
- Desafios das estimativas em projetos de [IA](#) e [Data Science](#)

Unidade IV – Teste de software

- Estratégia, Níveis, Tipos e Técnicas de teste do **Modelo V**, acompanhados da definição de planos, roteiros e casos de testes.
- Métricas de avaliação da qualidade dos modelos da engenharia de SW (acoplamento, coesão, profundidade de árvore de herança, resposta de classe, número de filhos), com uso do software de medição Source Monitor
- Planejamento, aplicação e registros de **testes unitários** com método de avaliação de complexidade ciclomática, avaliação de enlaces, avaliação de limites, avaliação de condição e equivalência.
- Automação de testes unitários com scripts JUNIT, aplicando os métodos de testes unitários estudados e TDD
- **Testes automatizados** de integração de componentes no modelo BDD (Behaviour Driven Development) com Maven, Mockito, Cucumber
- Elaboração e registro de testes de sistemas funcionais e não funcionais, aplicando ferramentas de apoio ao teste manual para validar Casos de Uso: TestLink aplicado

CONTEÚDO DO CURSO

2º semestre

- Testes funcionais de sistemas com automação na prática: Smoke test usando Xenu, Record & Playback teste usando [Selenium IDE](#), Data driven test usando [Katalon Studio](#), teste de protótipo com ZAPTest.
- Testes não funcionais de portabilidade, desempenho e robustez de sistema, incluindo a prática com ferramentas de automação [JMETER](#), e BlazeMeter
- Criação de casos de testes para validar aplicações de IA baseadas em NLP
- Emprego de IA na geração de casos de testes para a estratégia de teste traçada, com utilização de [IA Generativa](#).
- Impacto do desenvolvimento NO CODE: exemplificação do uso de ferramentas CASE (com prática de geração de código com ASTAH)
- Impacto da qualidade de dados no processamento do software, Data profiling e data cleansing com [Data Cleaner](#)
- Testes exploratórios de aceitação pelo usuário e critérios de liberação de software

I REFERÊNCIAS

BÁSICA

BRAGA, Pedro Henrique C. Teste de Software. Pearson. 2018.

VASQUEZ, Carlos Eduardo - SOMÕES, Guilherme Siqueira - ALBERT, Renato Machado. Pontos de Função/Medição, Estimativas e Gerenciamento de Projetos de Software. Érica. 2013.

R Maldonado, José. Automatização de Teste de Software com Ferramentas de Software Livre. São Paulo: LTC editora, 2018.





- 1- Parducci, Renato J.. Oliveira, Elisamara. TOGAF: Arquitetura de soluções de TI para empresas. São Paulo: Phorte Editora, 2019.
- 2- SELEME, Robson, STADLER, Humberto. Controle da Qualidade - As ferramentas essenciais. 1ª ed. Intersaberes, 2012. *
- 3- MANSUR, R. Governança da nova TI. A Revolução. Rio de Janeiro: Editora Ciência Moderna, 2013.
- 4- Rubin, Kenneth S. Scrum Essencial - Um Guia Prático Para o Mais Popular Processo Ágil. Alta Books. 2018.
- 5- PRESSMAN, R. S.; Maxim, B. R. Engenharia de Software – Uma abordagem Profissional. 7ª ed. Porto Alegre: Editora Bookman, 2011.
- 6- PFLEEGER, Shari Lawrence. Engenharia de software: teoria e prática. 2ª ed. São Paulo: Pearson Education do Brasil, 2004. *
- 7- FOGGETTI, C. (organizador) Gestão Ágil de Projetos. São Paulo: Pearson Education do Brasil, 2014. *
- 8- LÉLIS, Eliacy Cavalcanti. (organizador) Gestão da Qualidade. 1ª ed. São Paulo: Pearson Education do Brasil 2012. *
- 9- ISACA. COBIT 5, USA, 2014 - Disponível em: <http://www.isaca.org/cobit/pages/default.aspx>. *
- 10- SEI, Carnegie Mellon University. CMMi V3. SEI - Software Engineering Institute., USA, 2007. Disponível em: <https://www.sei.cmu.edu/cmmi/> *
- 11- SELEME, Robson, STADLER, Humberto. Controle da Qualidade - As ferramentas essenciais. 1ª ed. Intersaberes, 2012. *

Copyright © 2025 Prof. Paulo Sampaio
profpaulo.sampaio@fiap.com.br
www.linkedin.com/in/profpaulosampaio

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).