ACADEMIC PROGRAM

SOFTWARE VERIFICATION

B.F.A. IN **COMPUTER SCIENCE**

MODALITY: ON CAMPUS

ACADEMIC YEAR: 2022-2023

Name of the course:	Software verification
Degree :	Computer Science
Location:	Centro Universitario de Tecnología y Arte Digital
Area:	Optativity
Year:	4º
Teaching period:	2
Type:	OP
ECTS credits:	3
Teaching modality:	On campus
Language:	English
Lecturer / Email	-
Web page:	http://www.u-tad.com/

SUBJECT DESCRIPTION

Area description

This subject includes some advanced and/or specialized content that a generalist software engineer may require.

Subject description

This course is an approach to software quality. Therefore, it focuses on understanding the basic concepts of quality and its special characteristics when applied to the world of software, with special emphasis on the analysis of use cases, test planning and monitoring and, in general, the basic elements present in any software testing process, regardless of the way in which it is implemented. These techniques and practices are based on industry standards when designing and specifying test cases, as well as their manual or automated execution.

COMPETENCIES AND LEARNING OUTCOMES

Competencies

BASIC AND GENERAL SKILLS

CG1 -Ability to understand, schedule and solve problems trough software development

CG2 - To develop software that are environmental friendly, engaged with society and natural resources and law and ethics compliant

CG3 - Knowledge of the scientific fundamentals applicable to the resolution of computer problems

CG4 - Ability to simplify and optimize computer systems by understanding their complexity

CG6 - Develop collaborative projects showing teamwork skills, versatility, flexibility, creativityand respect for the work of the team members

CG7 - Knowledge of the creative foundations of ideation in software development projects.

CG9 - Ability to learn, modify and develop new software solutions

CG10 - Use of creative techniques to carry out computer projects

CG11 -Ability to search, analyze and manage information for insights capture

BC1: Students should demonstrate knowledge in an area of study that builds upon the foundation of general secondary education and goest beyond at a level that, while supported by advanced textbooks, also encompasses certain aspects derived from the cutting edge of their field of study.

BC2: Students should be able to apply their knowledge to their work or vocation in a professional manner, and they should possess the competencies typically demonstrated through the development and defence of arguments as well as problem-solving within their field of study.

BC3: Students must possess the ability to gather and interpret relevant data (usually within their field of study) in order to make judgments that involve reflection on socially, scientifically, or ethically significant issues.

BC4: Students should be capable of conveying information, ideas, problems, and solutions to both specialized and non-specialized audiences.

BC5: Students should have developed the learning skills necessary to pursue further studies with a high degree of autonomy.

TRANVERSALES SKILLS

CT1 - Knowledge of the definition, scope and implementation of the fundamentals of project management methodologies for technology projects

CT2 - Knowledge of the main sectorial players and the life cycle of a digital content development and commercialization project

CT4 - Ability to update the knowledge acquired in the management of digital tools and technologies according to the current state of affairs of the sector and the technological solution

CT5 - Development of the basic skills for digital entrepreneurship.

SPECIFIC SKILLS

CE10 - Ability to work with a release manager and generate application documentation automatically.

CE15 - Knowledge of fault tolerance, adaptability, load balancing and system predictability for distributed application development

CE17 - Knowledge of the parallelization characteristics of graphics cards and high-performance architectures for application development.

CE20 - Ability to test the operation and functionality of a computer application, develop test plans and use test-oriented design and programming techniques

CE21 -Ability to assess the quality of a computer application by applying software quality measurement metrics, procedures, and standards

Learning outcomes

Upon completion of the degree, the graduate will be able to:

- To understand the software quality assurance cycle

- To design a testing plan

- To know the most common testing frameworks in industry

- To develop an intensive-GPU application

- To be able to profile a distributed application

CONTENTS

Software testing and validation

Software Quality

Software maintenance and reverse engineering

SUBJECT SYLLABUS

Topic 1. Software Quality Fundamentals

1.1. What is Quality?

1.2. Quality in sw

1.3. Basic concepts: QA. QC, testing

1.4. Software verification or testing

1.5. The seven principles of testing

1.6. Testing as a process

Topic 2. QA in the SDLC

2.1. Software quality in predictive development

2.2. Software quality in agile paradigm

2.3. Maintenance and support

2.4. Test Cases from USsTema

3. Review and testing

3.1. Types of tests

3.2. White box testing

3.3. Static code analysis

3.4. Review

3.5. Black box testing

3.6 Experience-based testing

Topic 4. Techniques and tools

4.1. Testing tools

4.2. Test oriented models: TDD, BDD, ATDD

4.3. Test automation

4.4. CI/CD and DevOps

Topic 5. Test management

5.1. Testing roles

5.2. Reporting and control. Traceability

TRAINING ACTIVITIES AND TEACHING METHODOLOGIES

TRAINING ACTIVITIES

LEARNING ACTIVITIES	**Total hours**	**Hours of presence**
Theoretical / Expository classes	16,00	16,00
Practical classes	11,00	11,00
Tutorials	2,00	1,00
Independent study and autonomous work of the student	25,00	0,00
Elaboration of work (group or individual)	18,00	0,00
Evaluation Activities	3,00	3,00
TOTAL	75	31

Teaching methodologies

Expository method or master lesson

Case learning

Learning based on problem solving

Cooperative or collaborative learning

inquiry learning

Flipped classroom methodology

Gamification

Just in time Teaching (JITT) or classroom on time

Expository method or master lesson

Case method

Learning based on problem solving

Cooperative or collaborative learning

inquiry learning

Flipped classroom methodology

Gamification

TEMPORAL DEVELOPMENT

DIDACTIC UNITS / TOPICS TIME PERIOD

Topic 1. Fundamentals of Software Quality 2-3 weeks

Topic 2. QA in the SDLC 3 weeks

Topic 3. Review and Testing 4-5 weeks

Topic 4. Techniques and tools 3 weeks

Topic 5. Test Management 1 week

EVALUATION SYSTEM

ASSESSMENT SYSTEM	**MINIMUM SCORE RESPECT TO THE FINAL ASSESSMENT (%)**	**MAXIMUM SCORE RESPECT TO THE FINAL ASSESSMENT (%)**
Assessment of participation in class, exercises or projects of the course	10	30
Assessment of assignments, projects, reports, memos	30	60

Objective test	30	60

GRADING CRITERIA

ASSESSMENT SYSTEM	ORDINARY EVALUATION	EXTRAORDINARY EVALUATION
Assessment of participation in class, exercises or projects of the course	10	10
Assessment of assignments, projects, reports, memos	60	60
Objective test	30	30

General comments on the evaluations/assessments

- Active participation" will be weighted at 10% of the final grade. "Active participation" does not mean coming to class or "winning" group exercises. The voluntary resolution of exercises and expositions will be valued. Likewise, the exposition of ideas, the participation in debates, the presentation of proposals or additional exercises and, in general, everything that demonstrates an involvement in the subject, and not the mere passive attendance, will be valued as part of this section.

- Throughout the course there will be activities, exercises and assignments that must be submitted before the indicated date through the virtual platform. The set of assignments will account for 60% of the final grade. It is necessary to pass this section with a 5 to pass the course, and a 4 to release it in the extraordinary call.

- All the code and work handed in by the students must be ORIGINAL. This means that they must have been developed by the students throughout the course, without external help. In case of using external code/libraries to the one provided by the teacher, it must be properly documented and justified. It is allowed to consult external documentation to the course, but the code provided by the student must respect the copyright laws and software licenses in force. In any case, the student must be able to explain the code used and delivered during the course.

- Work submitted out of form and time without a justified cause will not be evaluated.

- The final exam will be worth 30%.It is necessary to pass the exam with a 5 to pass the course, and a 4 to release it in extraordinary call.

- In the event that a student has not achieved a 5 in projects and exam, but has at least a 4 in both sections, he/she will be assigned a special additional work to be determined to cover the most deficient aspects of the previous deliveries.- The course can only be passed if the average grade exceeds a grade of 5.0, fulfilling the above requirements.If any of these requirements are not fulfilled, the course will be considered automatically failed regardless of the rest of the grades.

- Those students who fail the assignments in the Ordinary Examination will have the possibility of repeating them in the Extraordinary Examination.- Any written work submitted by the student (problems, exams, program comments, etc.) must be well presented, correctly written (with the commas, periods and semicolons in their proper place) and without spelling mistakes.If this is not the case, the grade for the paper may be lowered by up to 20%, since a university student is expected to be of the highest quality in his or her written expression.

- Exam and project grades are not saved between successive academic years.

- It is not possible to obtain a Matrícula de Honor (MH) in the Extraordinary Exams.

- The use of notes is not allowed in the exams.Cell phones, or calculators of any kind, for AC65which the student must refer to the specific instructions of the professor on this subject.

- The COMPLETE course will be failed if it is discovered that in the exam the student copies another student (both will be failed) or copies from a book or from the Internet. In addition, the university will open disciplinary proceedings against both students, which may even lead to their expulsion.

LIST OF REFERENCES (BOOKS, PUBLICATIONS, WEBSITES):

Basic Bibliography:

• G. Myers, “The Art of software testing”, Wiley John + Sons, ISBN: 978-1118031964

• C. Kaner, J. Falk, H.Q. Nguyen, “Testing Computer Software”, Wiley John +Sons, ISBN: 978-0471358466

• D. Graham, R. Black, E. van Veenendaal, “Foundations of software testing”,Cengage Learning EMEA, ISBN: 978-1473764798

Recommended Bibliography:

• R. Pressman, B. Maxim, “Software Engineering: A Practitioner's Approach”,Ninth edition, McGraw-Hill Education, ISBN: 978-1260548006

• K. Beck, “Test Driven Development. By Example”, Addison-Wesley, ISBN: 978-0321146533

• K. Beck, “Extreme Programming Explained: Embrace Change” , Addison-Wesley, ISBN: 978-0321278654

• ISO/IEC/IEEE 29119-1 “Software and systems engineering — Software testing”

REQUIRED MATERIALS, SOFTWARE AND TOOLS

Type of classroom

Theory classroom

Board and projection system

Materials:

Personal Computer

Software:

Editor de texto: Notepad ++

Selenium Web, Cucumber