Advanced Master's Degree

Software Engineering and Quality

Accreditation/Membership

ASEE American Society for Engineering Education

tech global university

tech global university

Advanced Master's Degree
Software Engineering and Quality

- Modality: **online**
- Duration: **2 years**
- Certificate: **TECH Global University**
- Accreditation: **120 ECTS**
- Schedule: **at your own pace**
- Exams: **online**

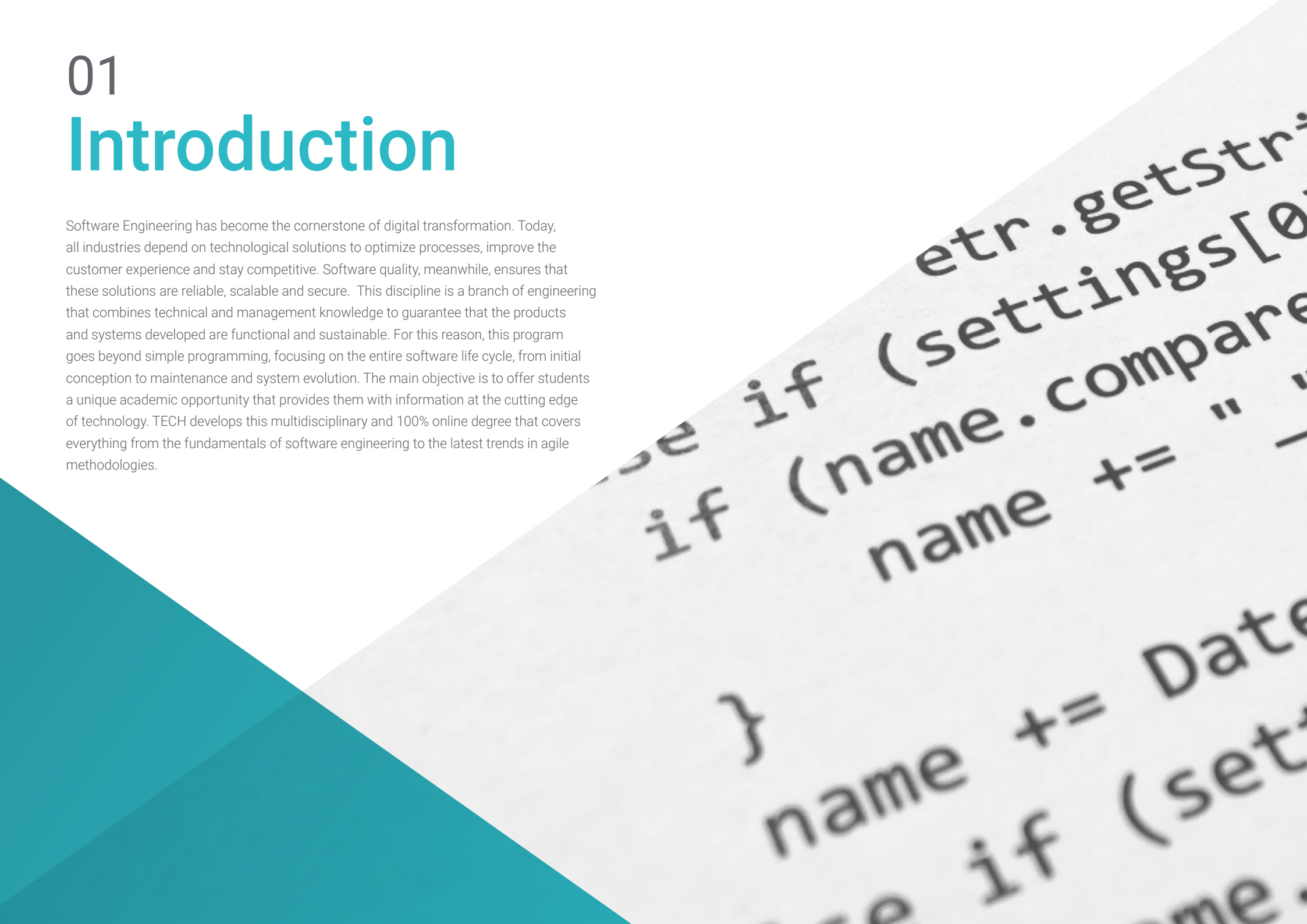
Website: www.techtitute.com/us/information-technology/advanced-master-degree/advanced-master-degree-software-engineering-quality

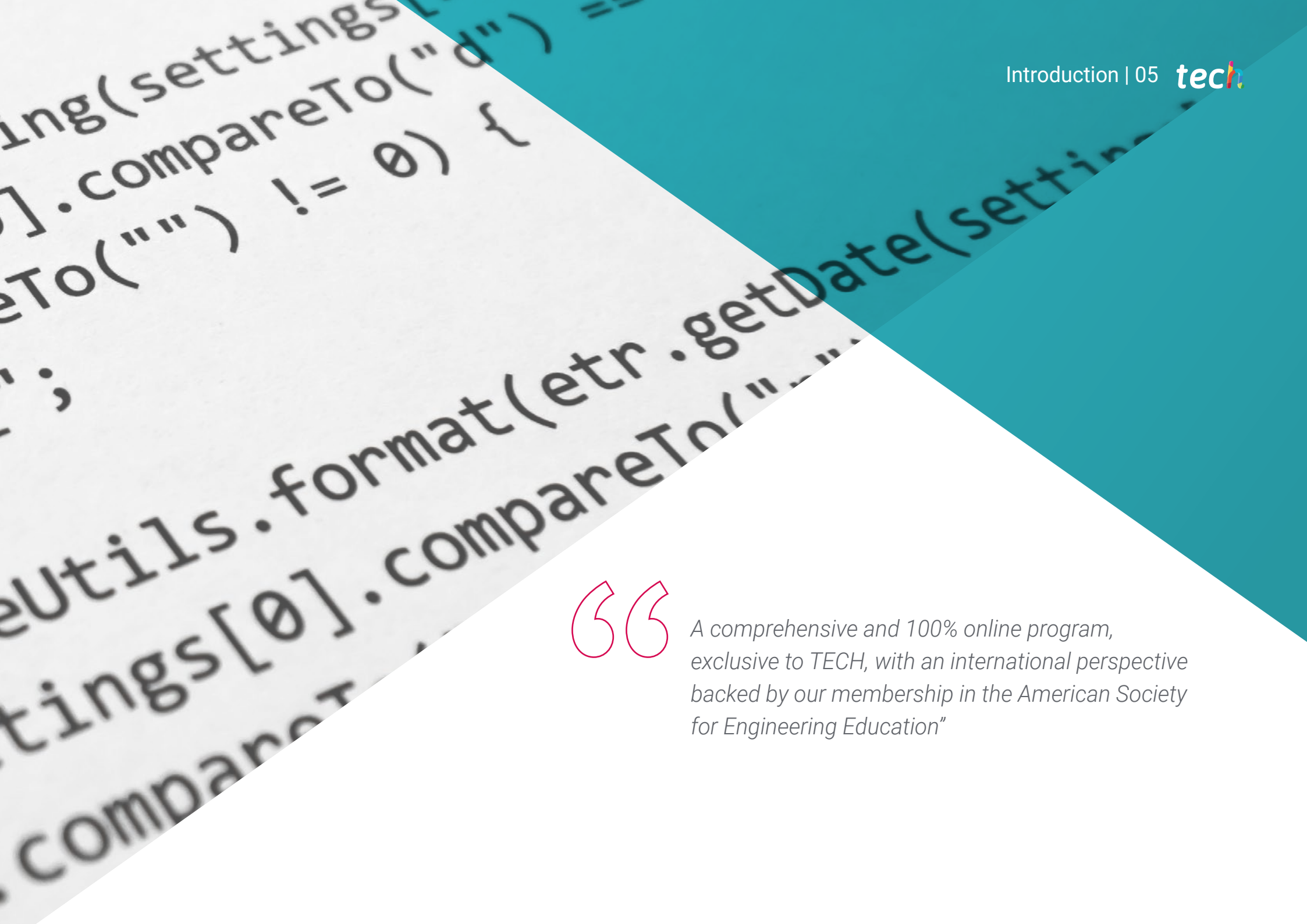
Index

01

Introduction

Software Engineering has become the cornerstone of digital transformation. Today, all industries depend on technological solutions to optimize processes, improve the customer experience and stay competitive. Software quality, meanwhile, ensures that these solutions are reliable, scalable and secure. This discipline is a branch of engineering that combines technical and management knowledge to guarantee that the products and systems developed are functional and sustainable. For this reason, this program goes beyond simple programming, focusing on the entire software life cycle, from initial conception to maintenance and system evolution. The main objective is to offer students a unique academic opportunity that provides them with information at the cutting edge of technology. TECH develops this multidisciplinary and 100% online degree that covers everything from the fundamentals of software engineering to the latest trends in agile methodologies.





A comprehensive and 100% online program, exclusive to TECH, with an international perspective backed by our membership in the American Society for Engineering Education"

Software quality guarantees that systems not only meet functional requirements, but are also intuitive, secure and sustainable in the long term. This is particularly relevant in critical sectors such as finance, healthcare and transportation, where failures can have serious consequences. Furthermore, prioritizing quality ensures that companies can adapt quickly to constant technological advances and respond effectively to growing market demands.

Through the use of methodologies such as agile development, DevOps and the implementation of international quality standards, software engineering guarantees the delivery of products in shorter times. Furthermore, controlled costs and a level of quality that minimizes critical errors have been expanded with the integration of emerging technologies, such as artificial intelligence, cloud computing and cybersecurity. In this context, the program designed by TECH is aimed at training professionals to be highly skilled in the design, development, management and quality assurance of software. In order to acquire the necessary skills, the Advanced Master's Degree syllabus includes the most up-to-date concepts in technology project management and strategic management. This approach represents added value both for engineers who already hold positions of responsibility and wish to update their knowledge, and for those who aspire to lead teams and projects in this field for the first time.

One of the main benefits of this program is that it will be 100% online, eliminating the need for travel and specific schedules. One of the main benefits of this program is that it will be 100% online, so there is no need to travel or adapt to fixed schedules. This flexible approach is very useful, as it allows students to organize their daily obligations, whether professional or family, efficiently, thereby achieving comprehensive development.

Thanks to TECH's membership in the **American Society for Engineering Education (ASEE)**, its students gain free access to annual conferences and regional workshops that enrich their engineering education. Additionally, they enjoy online access to specialized publications such as Prism and the Journal of Engineering Education, enhancing their academic development and expanding their professional network on an international scale.

This **Advanced Master's Degree in Software Engineering and Quality** contains the most complete and up-to-date program on the market. The most important features include:

- Practical cases presented by experts in IT
- The graphic, schematic, and practical contents with which they are created, provide scientific and practical information on the disciplines that are essential for professional practice
- Practical exercises where self-assessment can be used to improve learning
- Special emphasis on innovative methodologies in Software Engineering and Quality
- Theoretical lessons, questions to the expert, debate forums on controversial topics, and individual reflection assignments
- Content that is accessible from any fixed or portable device with an Internet connection

> *With TECH, you will not only learn to develop software, but also to create systems that make a difference in the lives of people and companies"*

Master the most advanced engineering skills and tools with the most innovative teaching methodology on the current educational market"

Raise your professional ambitions by learning 100% online, without compromising your personal and family responsibilities.

Become a professional engineering leader, ready to learn from anywhere in the world.

Its teaching staff includes professionals from the field of information technology, who bring to this program the experience of their work, as well as recognized specialists from leading companies and prestigious universities.

The multimedia content, developed with the latest educational technology, will provide the professional with situated and contextual learning, i.e., a simulated environment that will provide an immersive learning experience designed to prepare for real-life situations.

This program is designed around Problem-Based Learning, whereby the student must try to solve the different professional practice situations that arise throughout the program. For this purpose, the professional will be assisted by an innovative interactive video system created by renowned and experienced experts.

02

Why Study at TECH?

TECH is the world's largest online university. With an impressive catalog of more than 14,000 university programs available in 11 languages, it is positioned as a leader in employability, with a 99% job placement rate. In addition, it relies on an enormous faculty of more than 6,000 professors of the highest international renown.

Study at the world's largest online university and guarantee your professional success. The future starts at TECH"

The world's best online university according to FORBES

The prestigious Forbes magazine, specialized in business and finance, has highlighted TECH as "the world's best online university" This is what they have recently stated in an article in their digital edition in which they echo the success story of this institution, "thanks to the academic offer it provides, the selection of its teaching staff, and an innovative learning method aimed at educating the professionals of the future"

A world-class teaching staff

TECH's teaching staff is made up of more than 6,000 professors with the highest international recognition. Professors, researchers and top executives of multinational companies, including Isaiah Covington, performance coach of the Boston Celtics; Magda Romanska, principal investigator at Harvard MetaLAB; Ignacio Wistumba, chairman of the department of translational molecular pathology at MD Anderson Cancer Center; and D.W. Pine, creative director of TIME magazine, among others.

The world's largest online university

TECH is the world's largest online university. We are the largest educational institution, with the best and widest online educational catalog, one hundred percent online and covering the vast majority of areas of knowledge. We offer a large selection of our own degrees and accredited online undergraduate and postgraduate degrees. In total, more than 14,000 university degrees, in ten different languages, make us the largest educational institution in the world.

The most complete study plans on the university scene

TECH offers the most complete study plans on the university scene, with syllabuses that cover fundamental concepts and, at the same time, the main scientific advances in their specific scientific areas. In addition, these programs are continuously being updated to guarantee students the academic vanguard and the most in-demand professional skills. In this way, the university's qualifications provide its graduates with a significant advantage to propel their careers to success.

A unique learning method

TECH is the first university to use *ReLearning* in all its programs. It is the best online learning methodology, accredited with international teaching quality certifications, provided by prestigious educational agencies. In addition, this disruptive educational model is complemented with the "Case Method", thereby setting up a unique online teaching strategy. Innovative teaching resources are also implemented, including detailed videos, infographics and interactive summaries.

The official online university of the NBA

TECH is the official online university of the NBA. Thanks to our agreement with the biggest league in basketball, we offer our students exclusive university programs, as well as a wide variety of educational resources focused on the business of the league and other areas of the sports industry. Each program is made up of a uniquely designed syllabus and features exceptional guest hosts: professionals with a distinguished sports background who will offer their expertise on the most relevant topics.

Leaders in employability

TECH has managed to become the leading university in employability. 99% of its students obtain jobs in the academic field they have studied, within one year of completing any of the university's programs. A similar number achieve immediate career enhancement. All this thanks to a study methodology that bases its effectiveness on the acquisition of practical skills, which are absolutely necessary for professional development.

Google Premier Partner

The American technology giant has awarded TECH the Google Google Premier Partner badge. This award, which is only available to 3% of the world's companies, highlights the efficient, flexible and tailored experience that this university provides to students. The recognition as a Google Premier Partner not only accredits the maximum rigor, performance and investment in TECH's digital infrastructures, but also places this university as one of the world's leading technology companies.

The top-rated university by its students

Students have positioned TECH as the world's top-rated university on the main review websites, with a highest rating of 4.9 out of 5, obtained from more than 1,000 reviews. These results consolidate TECH as the benchmark university institution at an international level, reflecting the excellence and positive impact of its educational model." Unas cifras que sitúan a TECH como la referencia universitaria absoluta a nivel internacional.

03

Syllabus

The syllabus for the Advanced Master's Degree in Software Engineering and Quality is designed to provide comprehensive and advanced specialization in all key areas of software engineering. The first modules focus on the fundamentals, ranging from software design and requirements management to technology architectures and agile methodologies. As the program progresses, students delve into more specialized areas such as test automation, continuous integration and quality assurance. In addition, subjects on technology project management are incorporated, where participants learn to lead multidisciplinary teams.

This Advanced Master's Degree prepares you to be the expert who makes a difference in the Software Engineering and Quality sector"

Module 1. Software Quality. TRL Development Levels

1.1. Elements that Influence Software Quality (I). The Technical Debt
- 1.1.1. The Technical Debt. Causes and Consequences
- 1.1.2. Software Quality. General Principles
- 1.1.3. Unprincipled and Principled Quality Software
  - 1.1.3.1. Consequences
  - 1.1.3.2. Necessity of Applying Quality Principles in Software
- 1.1.4. Software Quality Typology
- 1.1.5. Quality Software. Specific features

1.2. Elements that Influence Software Quality (II). Associated Costs
- 1.2.1. Software Quality. Influencing Elements
- 1.2.2. Software Quality. Misconceptions
- 1.2.3. Software Quality. Associated Costs

1.3. Software Quality Models (I). Knowledge Management
- 1.3.1. General Quality Models
  - 1.3.1.1. Total Quality Management
  - 1.3.1.2. European Business Excellence Model (EFQM)
  - 1.3.1.3. Six-Sigma Model
- 1.3.2. Knowledge Management Models
  - 1.3.2.1. Dyba Model
  - 1.3.2.2. Seks Model
- 1.3.3. Experience Factory and QIP Paradigm
- 1.3.4. Quality in Use Models (25010)

1.4. Software Quality Models (III). Quality in Data, Processes and SEI Models
- 1.4.1. Data Quality Data Model
- 1.4.2. Software Process Modeling
- 1.4.3. Software & Systems Process Engineering Metamodel Specification (SPEM)
- 1.4.4. SEI Models
  - 1.4.4.1. CMMI
  - 1.4.4.2. SCAMPI
  - 1.4.4.3. IDEAL

1.5. ISO Software Quality Standards (I). Analysis of the Standards
- 1.5.1. ISO 9000 Standards
  - 1.5.1.1. ISO 9000 Standards
  - 1.5.1.2. ISO Family of Quality Standards (9000)
- 1.5.2. Other ISO Standards Related to Quality
- 1.5.3. Quality Modeling Standards (ISO 2501)
- 1.5.4. Quality Measurement Standards (ISO 2502n)

1.6. ISO Software Quality Standards (II). Requirements and Assessment
- 1.6.1. Standards on Quality Requirements (2503n)
- 1.6.2. Standards on Quality Assessment (2504n)
- 1.6.3. ISO/IEC 24744:2007

1.7. TRL Development Levels (I). Levels 1 to 4
- 1.7.1. TRL Levels
- 1.7.2. Level 1: Basic Principles
- 1.7.3. Level 2: Concept and/or Application
- 1.7.4. Level 3: Critical Analytical Function
- 1.7.5. Level 4: Component Validation in Laboratory Environment

1.8. TRL Development Levels (II). Levels 5 to 9
- 1.8.1. Level 5: Component Validation in Relevant Environment
- 1.8.2. Level 6: System/Subsystem Model
- 1.8.3. Level 7: Demonstration in Real Environment
- 1.8.4. Level 8: Complete and Certified System
- 1.8.5. Level 9: Success in Real Environment

1.9. TRL Development Levels. Uses
- 1.9.1. Example of Company with Laboratory Environment
- 1.9.2. Example of an R&D&I Company
- 1.9.3. Example of an Industrial R&D&I Company
- 1.9.4. Example of a Laboratory-Engineering Joint Venture Company

1.10. Software Quality. Key Details
- 1.10.1. Methodological Details
- 1.10.2. Technical Details
- 1.10.3. Software Project Management Details
  - 1.10.3.1. Computer System Quality
  - 1.10.3.2. Software Product Quality
  - 1.10.3.3. Software Process Quality

Module 2. Software Project Development. Functional and Technical Documentation

2.1. Project Management
- 2.1.1. Project Management in Software Quality
- 2.1.2. Project Management. Advantages
- 2.1.3. Project Management Typology

2.2. Methodology in Project Management
- 2.2.1. Methodology in Project Management
- 2.2.2. Project Methodologies. Typology
- 2.2.3. Methodologies in Project Management. Application

2.3. Requirements Identification Phase
- 2.3.1. Identification of Project Requirements
- 2.3.2. Management of Project Meetings
- 2.3.3. Documentation to Be Provided

2.4. Models
- 2.4.1. Initial Phase
- 2.4.2. Analysis Phase
- 2.4.3. Construction Phase
- 2.4.4. Testing Phase
- 2.4.5. Delivery

2.5. Data Model to Be Used
- 2.5.1. Determination of the New Data Model
- 2.5.2. Identification of the Data Migration Plan
- 2.5.3. Data Set

2.6. Impact on Other Projects
- 2.6.1. Impact of a Project. Examples

2.7. MUST of the Project
- 2.7.1. MUST of the Project
- 2.7.2. Identification of the MUST of the Project
- 2.7.3. Identification of the Execution Points for Project Delivery

2.8. The Project Construction Team
- 2.8.1. Roles to be Involved According to the Project
- 2.8.2. Contact with HR for Recruitment
- 2.8.3. Project Deliverables and Schedule

2.9. Technical Aspects of a Software Project
- 2.9.1. Project Architect. Technical Aspects
- 2.9.2. Technical Leaders
- 2.9.3. Construction of the Project Software
- 2.9.4. Code Quality Assessment Sonar

2.10. Project Deliverables
- 2.10.1. Functional Analysis
- 2.10.2. Data Model
- 2.10.3. State Diagram
- 2.10.4. Technical Documentation

Module 3. SoftwareTesting. Test Automation

3.1. Software Quality Models
- 3.1.1. Product Quality
- 3.1.2. Process Quality
- 3.1.3. Quality of Use

3.2. Process Quality
- 3.2.1. Process Quality
- 3.2.2. Maturity Models
- 3.2.3. ISO 15504 Standards
  - 3.2.3.1. Purposes
  - 3.2.3.2. Context
  - 3.2.3.3. Stages

3.3. ISO/IEC 15504 Standard
- 3.3.1. Process Categories
- 3.3.2. Development Process. Example
- 3.3.3. Profile Fragment
- 3.3.4. Stages

3.4. CMMI (Capability Maturity Model Integration)
- 3.4.1. CMMI Capability Maturity Model Integration
- 3.4.2. Models and Areas. Typology
- 3.4.3. Process Areas
- 3.4.4. Capacity Levels
- 3.4.5. Process Management
- 3.4.6. Project Management

3.5. Change and Repository Management
- 3.5.1. Software Change Management
  - 3.5.1.1. Configuration Item. Continuous Integration
  - 3.5.1.2. Lines
  - 3.5.1.3. Flowcharts
  - 3.5.1.4. Branches
- 3.5.2. Repository
  - 3.5.2.1. Version Control
  - 3.5.2.2. Work Team and Use of the Repository
  - 3.5.2.3. Continuous Integration in the Repository

3.6. Team Foundation Server (TFS)
- 3.6.1. Installation and Configuration
- 3.6.2. Creation of a Team Project.
- 3.6.3. Adding Content to Source Code Control
- 3.6.4. TFS on Cloud

3.7. Testing
- 3.7.1. Motivation for Testing
- 3.7.2. Verification Testing
- 3.7.3. Beta Testing
- 3.7.4. Implementation and Maintenance

3.8. Load Testing
- 3.8.1. Load Testing
- 3.8.2. LoadView Testing
- 3.8.3. K6 Cloud Testing
- 3.8.4. Loader Testing

3.9. Unit Stress and Endurance Tests
- 3.9.1. Motivation of Unit Tests
- 3.9.2. Unit Testing Tools
- 3.9.3. Motivation for Stress Testing
- 3.9.4. Testing UsingStressTesting
- 3.9.5. Motivation for stress Resistance
- 3.9.6. Tests Using LoadRunner

3.10. Scalability. Scalable Software Design
- 3.10.1. Scalability and Software Architecture
- 3.10.2. Independence Between Layers
- 3.10.3. Coupling Between Layers. Architecture Patterns

Module 4. Software Project Management Methodologies. Waterfall Methodology vs Agile Methodology

4.1. Waterfall Methodology
- 4.1.1. Waterfall Methodology
- 4.1.2. Waterfall Methodology. Influence on Software Quality
- 4.1.3. Waterfall Methodology Examples

4.2. Agile Methodology
- 4.2.1. Agile Methodology
- 4.2.2. Agile Methodology. Influence on Software Quality
- 4.2.3. Agile Methodology. Examples

4.3. SCRUM Methodology
- 4.3.1. SCRUM Methodology
- 4.3.2. SCRUM Manifesto
- 4.3.3. SCRUM Application

4.4. Panel Kanban
- 4.4.1. Kanban Method
- 4.4.2. Panel Kanban
- 4.4.3. Panel Kanban. Application Examples

4.5. Waterfall Project Management
- 4.5.1. Project Phases
- 4.5.2. Vision in a Waterfall Project
- 4.5.3. Deliverables to Consider

4.6. Project Management in SCRUM
- 4.6.1. Phases in a SCRUM Project
- 4.6.2. Vision in a SCRUM Project
- 4.6.3. Deliverables to Consider

4.7. Waterfall vs. Scrum Comparison
- 4.7.1. Pilot Project Approach
- 4.7.2. Project Applying Waterfall. Example
- 4.7.3. Project Applying Waterfall. Example

4.8. Customer Vision
- 4.8.1. Documents in a Waterfall
- 4.8.2. Documents in a SCRUM
- 4.8.3. Comparison

4.9. Kanban Structure
- 4.9.1. User Stories
- 4.9.2. Backlog
- 4.9.3. Kanban Analysis

4.10. Hybrid Projects
- 4.10.1. Project Construction
- 4.10.2. Project Management
- 4.10.3. Deliverables to Consider

Module 5 (*Test-Driven Development*). Test-Driven Software Design

5.1. TDD. Test Driven Development
- 5.1.1. TDD. Test Driven Development
- 5.1.2. TDD. Influence of TDD on Quality
- 5.1.3. Test-Driven Design and Development. Examples

5.2. TDD Cycle
- 5.2.1. Choice of a Requirement
- 5.2.2. Performing Tests. Typology
  - 5.2.2.1. Unit Tests
  - 5.2.2.2. Integration Tests
  - 5.2.2.3. End To EndTests
- 5.2.3. Test Verification. Errors
- 5.2.4. Creation of the Implementation
- 5.2.5. Automated Test Execution
- 5.2.6. Elimination of Duplication
- 5.2.7. Requirements Lists Update
- 5.2.8. Repeating the TDD Cycle
- 5.2.9. TDD Cycle. Theoretical and Practical Example

5.3. TDD Implementation Strategies
- 5.3.1. Mock Implementation
- 5.3.2. Triangular Implementation
- 5.3.3. Obvious Implementation

5.4. TDD. Use. Advantages and Inconveniences
- 5.4.1. Advantages of Use
- 5.4.2. Limitations of Use
- 5.4.3. Quality Balance in the Implementation

5.5. TDD. Good Practices
- 5.5.1. TDD Rules
- 5.5.2. Rule 1: Have a Previous Test that Fails Before Coding in Production
- 5.5.3. Rule 2: Not to Write More than One Unit Test
- 5.5.4. Rule 3: Not to Write More Code than Necessary
- 5.5.5. Errors and Anti-Patterns to Avoid in TDD

5.6. Simulation of a Real Project to use TDD (I)
- 5.6.1. Project Overview (Company A)
- 5.6.2. Application of TDD
- 5.6.3. Proposed Exercises
- 5.6.4. Exercises. Feedback

5.7. Simulation of a Real Project to use TDD (II)
- 5.7.1. Project Overview (Company B)
- 5.7.2. Application of TDD
- 5.7.3. Proposed Exercises
- 5.7.4. Exercises. Feedback

5.8. Simulation of a Real Project to use TDD (III)
- 5.8.1. General Description of the Project (Company C)
- 5.8.2. Application of TDD
- 5.8.3. Proposed Exercises
- 5.8.4. Exercises. Feedback

5.9. Alternatives to TDD. Test Driven Development
- 5.9.1. TCR (Test Commit Revert)
- 5.9.2. BDD (Behavior Driven Development)
- 5.9.3. ATDD (Acceptance Test Driven Development)
- 5.9.4. TDD. Theoretical Comparison

5.10. TDD TCR, BDD and ATDD. Practical Comparison
- 5.10.1. Defining the Problem
- 5.10.2. Resolution with TCR
- 5.10.3. Resolution with BDD
- 5.10.4. Resolution with ATDD

Module 6. DevOps. Software Quality Management

6.1. DevOps. Software Quality Management
- 6.1.1. DevOps.
- 6.1.2. DevOps and Software Quality
- 6.1.3. DevOps. Benefits of DevOps Culture

6.2. DevOps. Relation to Agile
- 6.2.1. Accelerated Delivery
- 6.2.2. Quality
- 6.2.3. Cost Reduction

6.3. DevOps Implementation
- 6.3.1. Problem Identification
- 6.3.2. Implementation in a Company
- 6.3.3. Implementation Metrics

6.4. Software Delivery Cycle
- 6.4.1. Design Methods
- 6.4.2. Agreements
- 6.4.3. Roadmap

6.5. Error-Free Code Development
- 6.5.1. Maintainable Code
- 6.5.2. Development Patterns
- 6.5.3. Code Testing
- 6.5.4. Software Development at Code Level. Good Practices

6.6. Automation
- 6.6.1. Automation. Types of Tests
- 6.6.2. Cost of Automation and Maintenance
- 6.6.3. Automation. Mitigating Errors

6.7. Deployment
- 6.7.1. Target Assessment
- 6.7.2. Design of an Automatic and Adapted Process
- 6.7.3. Feedback and Responsiveness

6.8. Information Security
- 6.8.1. Incident Management
- 6.8.2. Incident Analysis and Resolution
- 6.8.3. How to Avoid Future Mistakes

6.9. Deployment Automation
- 6.9.1. Preparing for Automated Deployments
- 6.9.2. Assessment of the Health of the Automated Process
- 6.9.3. Metrics and Rollback Capability

6.10. Good Practices. Evolution of DevOps
- 6.10.1. Guide of Good Practices applying DevOps
- 6.10.2. DevOps. Methodology for the Team
- 6.10.3. Avoiding Niches

Module 7. DevOps and Continuous Integration. Advanced Practical Solutions in Software Development

7.1. Software Delivery Flow
- 7.1.1. Identification of Actors and Artifacts
- 7.1.2. Software Delivery Flow Design
- 7.1.3. Software Delivery Flow. Inter-Stage Requirements

7.2. Process Automation
- 7.2.1. Continuous Integration
- 7.2.2. Continuous Deployment
- 7.2.3. Environment Configuration and Secret Management

7.3. Declarative Pipelines
- 7.3.1. Differences Between Traditional, Code-Like and Declarative Pipelines
- 7.3.2. Declarative Pipelines
- 7.3.3. Declarative Pipelines in Jenkins
- 7.3.4. Comparison of Continuous Integration Providers

7.4. Quality Gates and Enriched Feedback
- 7.4.1. Quality Gates
- 7.4.2. Quality Standards with Quality Gates. Maintenance
- 7.4.3. Business Requirements in Integration Requests

7.5. Artifact Management
- 7.5.1. Artifacts and Life Cycle
- 7.5.2. Artifact Storage and Management Systems
- 7.5.3. Security in Artifact Management

7.6. Continuous Deployment
- 7.6.1. Continuous Deployment as Containers
- 7.6.2. Continuous Deployment with PaaS

7.7. Improving Pipeline Runtime: Static Analysis and Git Hooks
- 7.7.1. Static Analysis
- 7.7.2. Code Style Rules
- 7.7.3. Git Hooks and Unit Tests
- 7.7.4. The Impact of Infrastructure

7.8. Vulnerabilities in Containers
- 7.8.1. Vulnerabilities in Containers
- 7.8.2. Image Scanning
- 7.8.3. Periodic Reports and Alerts

Module 8. Database (DB) Design. Standardization and performance. Software Quality

8.1. Database Design
- 8.1.1. Databases. Typology
- 8.1.2. Databases Currently Used
  - 8.1.2.1. Relational
  - 8.1.2.2. Key-Value
  - 8.1.2.3. Based on Graphs
- 8.1.3. Data Quality

8.2. Entity-Relationship Model Design (I)
- 8.2.1. Entity-Relationship Model. Quality and Documentation
- 8.2.2. Entities
  - 8.2.2.1. Strong Entity
  - 8.2.2.2. Weak Entity
- 8.2.3. Attributes
- 8.2.4. Set of Relations
  - 8.2.4.1. 1 to 1
  - 8.2.4.2. 1 to Many
  - 8.2.4.3. Many to 1
  - 8.2.4.4. Many to Many
- 8.2.5. Keys
  - 8.2.5.1. Primary Key
  - 8.2.5.2. Foreign Key
  - 8.2.5.3. Weak Entity Primary Key
- 8.2.6. Restrictions
- 8.2.7. Cardinality
- 8.2.8. Heritage
- 8.2.9. Aggregation

8.3. Entity-Relationship Model (II). Tools
- 8.3.1. Entity-Relationship Model. Tools
- 8.3.2. Entity-Relationship Model. Practical Example
- 8.3.3. Entity-Relationship Model feasible
  - 8.3.3.1. Visual Sample
  - 8.3.3.2. Sample in Table Representation

8.4. Database (DB) Standardization (I). Software Quality Considerations
- 8.4.1. DB Standardization and Quality
- 8.4.2. Dependency
  - 8.4.2.1. Functional Dependence
  - 8.4.2.2. Properties of Functional Dependence
  - 8.4.2.3. Deduced Properties
- 8.4.3. Keys

8.5. Database (DB) Normalization (II). Normal Forms and Codd's Rules
- 8.5.1. Normal Forms
  - 8.5.1.1. First Normal Form (1FN)
  - 8.5.1.2. Second Normal Form (2FN)
  - 8.5.1.3. Third Normal Form (3FN)
  - 8.5.1.4. Boyce-Codd Normal Form (BCNF)
  - 8.5.1.5. Fourth Normal Form (4FN)
  - 8.5.1.6. Fifth Normal Form (5FN)
- 8.5.2. Codd's Rules
  - 8.5.2.1. Rule 1: Information
  - 8.5.2.2. Rule 2: Guaranteed Access
  - 8.5.2.3. Rule 3: Systematic Treatment of Null Values
  - 8.5.2.4. Rule 4: Description of the Database
  - 8.5.2.5. Rule 5: Integral Sub-Language
  - 8.5.2.6. Rule 6: View Update
  - 8.5.2.7. Rule 7: Insert and Update
  - 8.5.2.8. Rule 8: Physical Independence
  - 8.5.2.9. Rule 9: Logical Independence
  - 8.5.2.10. Rule 10: Integrity Independence
    - 8.5.2.10.1. Integrity Rules
  - 8.5.2.11. Rule 11: Distribution
  - 8.5.2.12. Rule 12: Non-Subversion
- 8.5.3. Practical Example

8.6. Data Warehouse/OLAP System
- 8.6.1. Data Warehouse
- 8.6.2. Fact Table
- 8.6.3. Dimension Table
- 8.6.4. Creation of the OLAP System. Tools

8.7. Database (DB) Performance
- 8.7.1. Index Optimization
- 8.7.2. Query Optimization
- 8.7.3. Table Partitioning

8.8. Simulation of Real Project for DB Design (I)
- 8.8.1. Project Overview (Company A)
- 8.8.2. Application of Database Design
- 8.8.3. Proposed Exercises
- 8.8.4. Proposed Exercises *Feedback*

8.9. Simulation of Real Project for BD Design (II)
- 8.9.1. Project Overview (Company B)
- 8.9.2. Application of Database Design
- 8.9.3. Proposed Exercises
- 8.9.4. Proposed Exercises *Feedback*

8.10. Relevance of DB Optimization to Software Quality
- 8.10.1. Design Optimization
- 8.10.2. Query Code Optimization
- 8.10.3. Stored Procedure Code Optimization
- 8.10.4. Influence of *Triggers* on Software Quality. Recommendations for Use.

Module 9. Design of Scalable Architectures. The Architecture in the Software Life Cycle

9.1. Design of Scalable Architectures (I)
- 9.1.1. Scalable Architectures
- 9.1.2. Principles of a Scalable Architecture
  - 9.1.2.1. Reliable
  - 9.1.2.2. Scalable
  - 9.1.2.3. Maintainable
- 9.1.3. Types of Scalability
  - 9.1.3.1. Vertical
  - 9.1.3.2. Horizontal
  - 9.1.3.3. Combined

9.2. Architecture DDD (*Domain-Driven Design*)
- 9.2.1. The DDD Model. Domain Orientation
- 9.2.2. Layers, Distribution of Responsibility and Design Patterns
- 9.2.3. Decoupling as a Basis for Quality

9.3. Design of Scalable Architectures (II). Benefits, Limitations and Design Strategies
- 9.3.1. Scalable Architecture. Benefits
- 9.3.2. Scalable Architecture. Limitations
- 9.3.3. Strategies for the Development of Scalable Architectures (Descriptive Table)

9.4. Software Life Cycle (I). Stages
- 9.4.1. Software Life Cycle
  - 9.4.1.1. Planning Stage
  - 9.4.1.2. Analysis Stage
  - 9.4.1.3. Design Stage
  - 9.4.1.4. Implementation Stage
  - 9.4.1.5. Testing Stage
  - 9.4.1.6. Installation/Deployment Stage
  - 9.4.1.7. Use and Maintenance Stage

9.5. Software Life Cycle Models
- 9.5.1. Waterfall Model
- 9.5.2. Repetitive Model
- 9.5.3. Spiral Model
- 9.5.4. Big Bang Model

9.6. Software Life Cycle (II). Automation
- 9.6.1. Software Development Life Cycle. Solutions
  - 9.6.1.1. Continuous Integration and Development (CI/CD)
  - 9.6.1.2. Agile Methodologies
  - 9.6.1.3. DevOps / Production Operations
- 9.6.2. Future Trends
- 9.6.3. Practical Examples

9.7. Software Architecture in the Software Life Cycle
- 9.7.1. Benefits
- 9.7.2. Limitations
- 9.7.3. Tools

9.8. Real Project Simulation for Software Architecture Design (I)
- 9.8.1. General Description of the Project (Company A)
- 9.8.2. Software Architecture Design Application
- 9.8.3. Proposed Exercises
- 9.8.4. Proposed Exercises *Feedback*

9.9. Simulation of a Real Project for Software Architecture Design (II)
- 9.9.1. Project Overview (Company B)
- 9.9.2. Software Architecture Design Application
- 9.9.3. Proposed Exercises
- 9.9.4. Proposed Exercises. Feedback

9.10. Simulation of a Real Project for Software Architecture Design (III)
- 9.10.1. General Description of the Project (Company C)
- 9.10.2. Software Architecture Design Application
- 9.10.3. Proposed Exercises
- 9.10.4. Proposed Exercises. *Feedback*

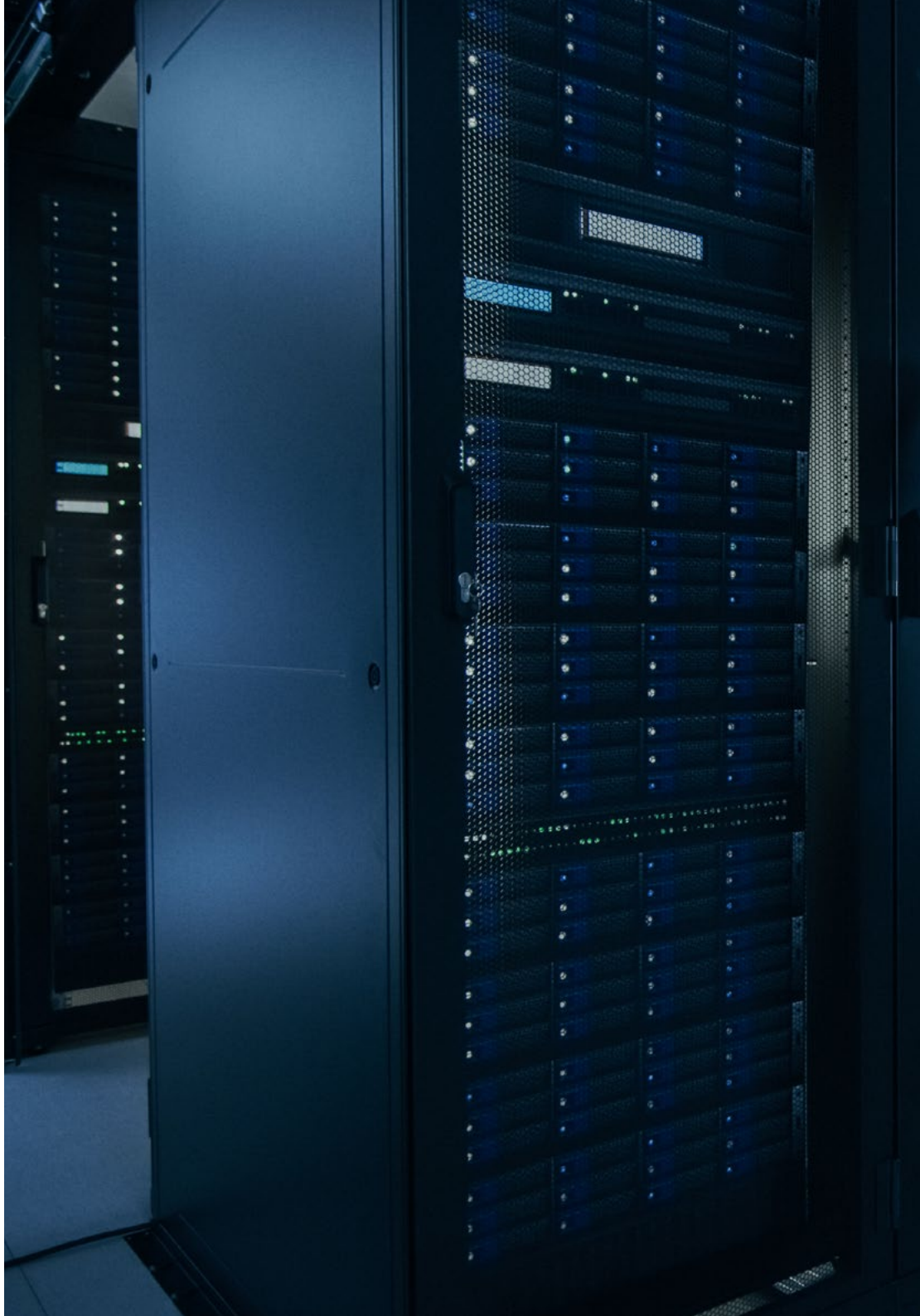
Module 10. ISO, IEC 9126 Quality Criteria. Software Quality Metrics

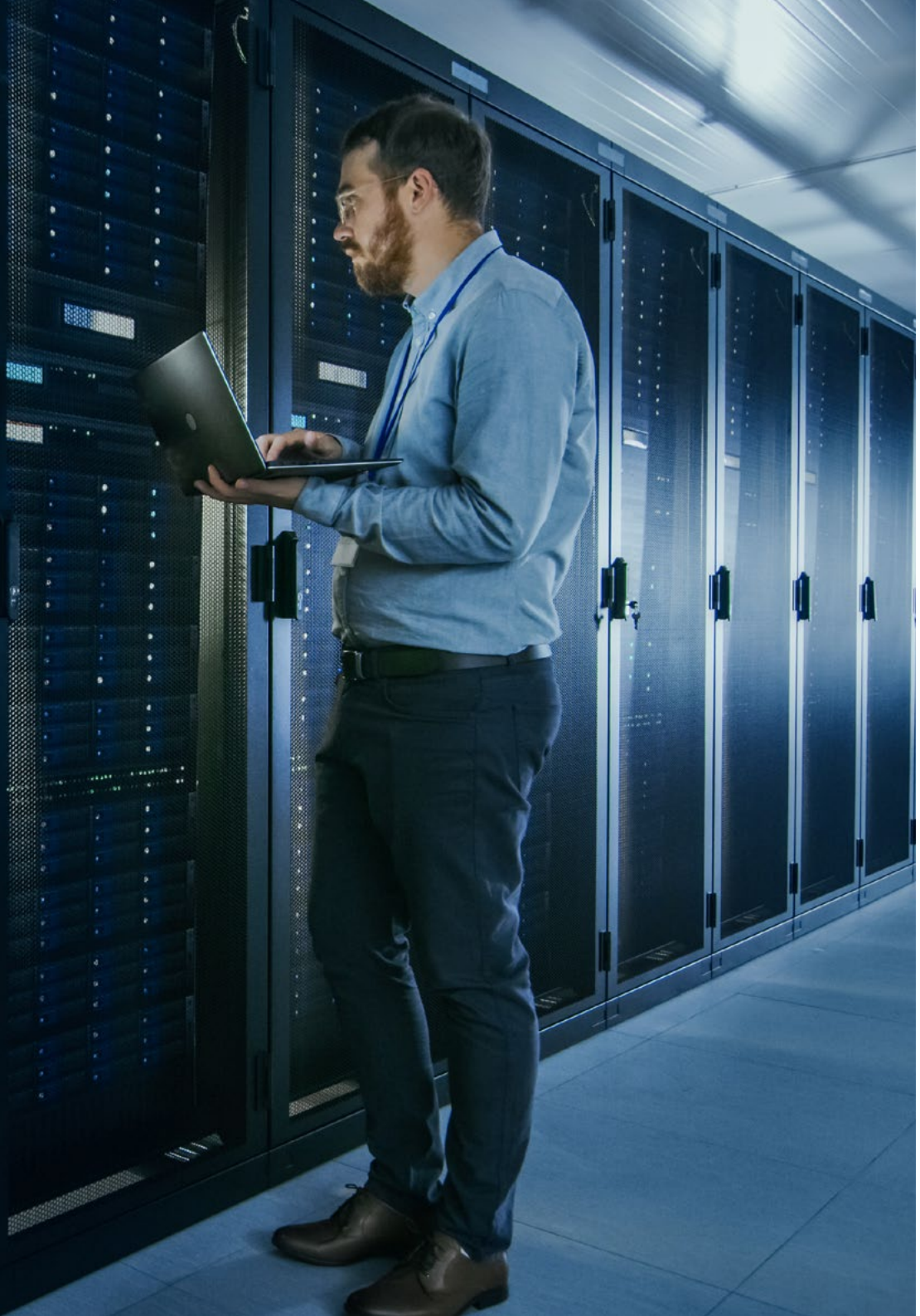
10.1. Quality Criteria. ISO, IEC 9126 Standard
- 10.1.1. Quality Criteria.
- 10.1.2. Software Quality Justification. ISO, IEC 9126 Standard
- 10.1.3. Software Quality Measurement as a Key Indicator

10.2. Software Quality Criteria. Features
- 10.2.1. Reliability
- 10.2.2. Functionality
- 10.2.3. Efficiency
- 10.2.4. Usability
- 10.2.5. Maintainability
- 10.2.6. Portability

10.3. ISO Standard, IEC 9126 (I). Introduction
- 10.3.1. Description of ISO, IEC 9126 Standard
- 10.3.2. Functionality
- 10.3.3. Reliability
- 10.3.4. Usability
- 10.3.5. Maintainability

10.3.6. Portability
10.3.7. Quality in Use
10.3.8. Software Quality Metrics
10.3.9. ISO 9126 Quality Metrics
10.4. ISO Standard, IEC 9126 (II). McCall and Boehm Models
10.4.1. McCall Model: Quality Factors
10.4.2. Boehm Model
10.4.3. Intermediate Level. Features
10.5. Software Quality Metrics (I). Components
10.5.1. Measurement
10.5.2. Metrics
10.5.3. Indicator
10.5.3.1. Types of Indicators
10.5.4. Measurements and Models
10.5.5. Scope of Software Metrics
10.5.6. Classification of Software Metrics
10.6. Software Quality Measurement (II). Measurement Practice
10.6.1. Metric Data Collection
10.6.2. Measurement of Internal Product Attributes
10.6.3. Measurement of External Product Attributes
10.6.4. Measurement of Resources
10.6.5. Metrics for Object-Oriented Systems
10.7. Design of a Single Software Quality Indicator
10.7.1. Single Indicator as a Global Qualifier
10.7.2. Indicator Development, Justification and Application
10.7.3. Example of Application. Need to Know the Detail
10.8. Simulation of Real Project for Quality Measurement (I)
10.8.1. General Description of the Project (Company A)
10.8.2. Application of Quality Measurement
10.8.3. Proposed Exercises
10.8.4. Proposed Exercises. *Feedback*





10.9. Real Project Simulation for Quality Measurement (II)
- 10.9.1. General Description of the Project (Company B)
- 10.9.2. Application of Quality Measurement
- 10.9.3. Proposed Exercises
- 10.9.4. Proposed Exercises *Feedback*

10.10. Real Project Simulation for Quality Measurement (III)
- 10.10.1. General Description of the Project (Company C)
- 10.10.2. Application of Quality Measurement
- 10.10.3. Proposed Exercises
- 10.10.4. Proposed Exercises. *Feedback*

Module 11. Methodologies, Development and Quality in Software Engineering

11.1. Model-Based Software Development
- 11.1.1. The Need for
- 11.1.3. Object Modeling
- 11.1.4. UML
- 11.1.5. CASE Tools

11.2. Application Modeling and Design Patterns with UML
- 11.2.1. Advanced Requirements Modeling
- 11.2.2. Advanced Static Modeling
- 11.2.3. Advanced Dynamic Modeling
- 11.2.4. Component Modeling
- 11.2.5. Introduction to Design Patterns with UML
- 11.2.6. Adapter
- 11.2.7. Factory
- 11.2.8. Singleton
- 11.2.9. Strategy
- 11.2.10. Composite
- 11.2.11. Facade
- 11.2.12. Observer

11.3. Model-Driven Engineering
- 11.3.1. Introduction
- 11.3.2. Metamodeling of Systems
- 11.3.3. MDA
- 11.3.4. DSL
- 11.3.5. Model Refinements with OCL
- 11.3.6. Model Transformations

11.4. Ontologies in Software Engineering
- 11.4.1. Introduction
- 11.4.2. Ontology Engineering
- 11.4.3. Application of Ontologies in Software Engineering

Module 12. Software Project Management

12.1. *Stakeholders* and Outreach Management
- 12.1.1. Identify Stakeholders
- 12.1.2. Develop Plan for Stakeholder Management
- 12.1.3. Manage Stakeholder Engagement
- 12.1.4. Control Stakeholder Engagement
- 12.1.5. The Objective of the Project
- 12.1.6. Scope Management and its Plan
- 12.1.7. Gathering Requirements
- 12.1.8. Define the Scope Statement
- 12.1.9. Create the WBS
- 12.1.10. Verify and Control the Scope

12.2. The Development of the Time-Schedule
- 12.2.1. Time Management and its Plan
- 12.2.2. Defining Activities
- 12.2.3. Establishment of the Sequence of Activities
- 12.2.4. Estimated Resources for Activities
- 12.2.5. Estimated Duration of Activities
- 12.2.6. Development of the Time-Schedule and Calculation of the Critical Path
- 12.2.7. Schedule Control

12.3. Budget Development and Risk Response
- 12.3.1. Estimate Costs
- 12.3.2. Develop Budget and S-Curve
- 12.3.3. Cost Control and Earned Value Method
- 12.3.4. Risk Concepts
- 12.3.5. How to Perform a Risk Analysis
- 12.3.6. The Development of the Response Plan

12.4. Communication and Human Resources
- 12.4.1. Planning Communications Management
- 12.4.2. Communications Requirements Analysis
- 12.4.3. Communication Technology
- 12.4.4. Communication Models
- 12.4.5. Communication Methods
- 12.4.6. Communications Management Plan
- 12.4.7. Communications Management
- 12.4.8. Management of Human Resources
- 12.4.9. Main Stakeholders and their Roles in the Projects
- 12.4.10. Types of Organization
- 12.4.11. Project Organization
- 12.4.12. The Work Equipment

12.5. Procurement
- 12.5.1. The Procurement Process
- 12.5.2. Planning
- 12.5.3. Search for Suppliers and Request for Quotations
- 12.5.4. Contract Allocation
- 12.5.5. Contract Administration
- 12.5.6. Contracts
- 12.5.7. Types of Contracts
- 12.5.8. Contract Negotiation

12.6. Execution, Monitoring and Control and Closure
- 12.6.1. Process Groups
- 12.6.2. Project Execution
- 12.6.3. Project Monitoring and Control
- 12.6.4. Project Closure

12.7. Professional Responsibility
- 12.7.1. Professional Responsibility
- 12.7.2. Characteristics of Social and Professional Responsibility
- 12.7.3. Project Leader Code of Ethics
- 12.7.4. Liability vs. PMP®
- 12.7.5. Examples of Liability
- 12.7.6. Benefits of Professionalization

Module 13. Software Development Platforms

13.1. Introduction to Application Development
- 13.1.1. Desktop Applications
- 13.1.2. Programming Language
- 13.1.3. Integrated Development Environments
- 13.1.4. Web Applications
- 13.1.5. Mobile Applications
- 13.1.6. Cloud Applications

13.2. Application Development and Graphical User Interface in Java
- 13.2.1. Integrated Development Environments for Java
- 13.2.2. Main IDE for Java
- 13.2.3. Introduction to the Eclipse Development Platform
- 13.2.4. Introduction to the NetBeans Development Platform
- 13.2.5. Controller View Model for Graphical User Interfaces
- 13.2.6. Design a Graphical Interface in Eclipse
- 13.2.7. Design a Graphical Interface in NetBeans

13.3. Debugging and Testing in Java
- 13.3.1. Testing and Debugging of Java programs
- 13.3.2. Debugging in Eclipse
- 13.3.3. Debugging in NetBeans

13.4. Application Development and Graphical User Interface in. NET
- 13.4.1. Net Framework
- 13.4.2. Components of the .NET Development Platform
- 13.4.3. Visual Studio .NET
- 13.4.4. .NET tools for GUI
- 13.4.5. The GUI with Windows Presentation Foundation
- 13.4.6. Debugging and Compiling a WPF Application

13.5. Programming for .NET Networks
- 13.5.1. Introduction to .NET Network Programming
- 13.5.2. Requests and Responses in .NET
- 13.5.3. Use of Application Protocols in .NET
- 13.5.4. Security in .NET Network Programming

13.6. Mobile Application Development Environments
- 13.6.1. Mobile Applications
- 13.6.2. Android Mobile Applications
- 13.6.3. Steps for Development in Android
- 13.6.4. The IDE Android Studio

13.7. Development of Applications in the Environment Android Studio
- 13.7.1. Install and Start Android Studio
- 13.7.2. Running an Android Application
- 13.7.3. Development of the Graphic Interface in Android Studio
- 13.7.4. Starting Activities in Android Studio

13.8. Debugging and Publishing of Android Applications
- 13.8.1. Debugging an Application in Android Studio
- 13.8.2. Memorizing Applications in Android Studio
- 13.8.3. Publishing an Application on Google Play

13.9. Cloud Application Development
- 13.9.1. *Cloud Computing*
- 13.9.2. *Cloud* Levels: SaaS, PaaS, IaaS
- 13.9.3. Main Development Platforms in the Cloud
- 13.9.4. Bibliographical References

13.10. Introduction to Google Cloud Platform
- 13.10.1. Basic Concepts of Google Cloud Platform
- 13.10.2. Google Cloud Platform Services
- 13.10.3. Tools in Google Cloud Platform

Module 14. Web-Client Computing

14.1. Introduction to HTML
- 14.1.1. Structure of the Document
- 14.1.2. Color
- 14.1.3. Text
- 14.1.4. Hypertext Links
- 14.1.5. Images
- 14.1.6. Lists
- 14.1.7. Tables
- 14.1.8. *Frames*
- 14.1.9. Forms
- 14.1.10. Specific Elements for Mobile Technologies
- 14.1.11. Obsolete Elements

14.2. Cascading Style Sheets (CSS)
- 14.2.1. Elements and Structure of a Cascading Style Sheet
  - 14.2.1.1. Creation of Style Sheets
  - 14.2.1.2. Application of Styles Selectors
  - 14.2.1.3. Style Inheritance and Cascading
  - 14.2.1.4. Page Formatting Using Styles
  - 14.2.1.5. Page Structuring Using Styles. The Box Model
- 14.2.2. Style Design for different Devices
- 14.2.3. Types of Style Sheets: Static and Dynamic. Pseudo-Classes
- 14.2.4. Best Practices in the Use of Style Sheets

14.3. Introduction and History of JavaScript
- 14.3.1. Introduction
- 14.3.2. History of JavaScript
- 14.3.3. Development Environment to be Used

14.4. Basic Notions of Web Programming
- 14.4.1. Basic JavaScript Syntax
- 14.4.2. Primitive Data Types and Operators
- 14.4.3. Variables and Areas
- 14.4.4. Text Strings and Template Literals
- 14.4.5. Numbers and Booleans
- 14.4.6. Comparisons

14.5. Complex JavaScript Structures
- 14.5.1. Vectors or Arrays and Objects
- 14.5.2. Sets
- 14.5.3. Maps
- 14.5.4. Disjunctive
- 14.5.5. Loops

14.6. Functions and Objects
- 14.6.1. Function Definition and Invocation
- 14.6.2. Arguments
- 14.6.3. Arrow Functions
- 14.6.4. Callback Functions
- 14.6.5. Higher Order Functions
- 14.6.6. Literal Objects
- 14.6.7. The *This* Object
- 14.6.8. Objects as Namespaces: theMaths and Date Objects

14.7. The Document Object Model (DOM)
- 14.7.1. What is the DOM?
- 14.7.2. A Bit of History
- 14.7.3. Navigation and Element Retrieval
- 14.7.4. A Virtual DOM with JSDOM
- 14.7.5. Query Selectors
- 14.7.6. Navigation using Properties
- 14.7.7. Assigning Attributes to Elements
- 14.7.8. Creation and Modification of Nodes
- 14.7.9. Updated Styling of the DOM Elements

14.8. Modern Web Development
- 14.8.1. Event-Driven Flow and Listeners
- 14.8.2. Modern Web Toolkits and Alignment Systems
- 14.8.3. Strict JavaScript Mode
- 14.8.4. More about Functions
- 14.8.5. Asynchronous Promises and Functions
- 14.8.6. *Closures*
- 14.8.7. Functional Programming
- 14.8.8. POO in JavaScript

14.9. Web Usability
- 14.9.1. Introduction to Usability
- 14.9.2. Definition of Usability
- 14.9.3. Importance of User-Centered Web Design
- 14.9.4. Differences Between Accessibility and Usability
- 14.9.5. Advantages and Problems in Combining Accessibility and Usability
- 14.9.6. Advantages and Difficulties in the Implementation of Usable Websites
- 14.9.7. Usability Methods
- 14.9.8. User Requirements Analysis
- 14.9.9. Conceptual Design Principles. User-Oriented Prototyping
- 14.9.10. Guidelines for the Creation of Usable Web Sites
  - 14.9.10.1. Usability Guidelines of Jakob Nielsen
  - 14.9.10.2. Usability Guidelines of Bruce Tognazzini
- 14.9.11. Usability Evaluation

14.10. Web Accessibility
- 14.10.1. Introduction
- 14.10.2. Definition of Web-Accessibility
- 14.10.3. Types of Disabilities
  - 14.10.3.1. Temporary or Permanent Disabilities
  - 14.10.3.2. Visual Impairment
  - 14.10.3.3. Hearing Impairment
  - 14.10.3.4. Motor Impairment
  - 14.10.3.5. Neurological or Cognitive Disabilities
  - 14.10.3.6. Difficulties Arising from Aging
  - 14.10.3.7. Limitations Arising from the Environment
  - 14.10.3.8. Barriers Preventing Access to the Web
- 14.10.4. Technical Aids and Support Products to Overcome Barriers
  - 14.10.4.1. Aids for the Blind
  - 14.10.4.2. Aids for Persons with Low Vision
  - 14.10.4.3. Aids for People with Color Blindness
  - 14.10.4.4. Aids for the Hearing Impaired
  - 14.10.4.5. Aids for the Motor Impaired
  - 14.10.4.6. Aids for the and Neurological Impaired
- 14.10.5. Advantages and Difficulties in the Implementation of Web Accessibility
- 14.10.6. Web Accessibility Regulations and Standards
- 14.10.7. Web Accessibility Regulatory Bodies
- 14.10.8. Comparison of Standards and Regulations
- 14.10.9. Guidelines for Compliance with Regulations and Standards
  - 14.10.9.1. Description of the Main Guidelines (Images, links, videos, etc.)
  - 14.10.9.2. Guidelines for Accessible Navigation
    - 14.10.9.2.1. Perceptibility
    - 14.10.9.2.2. Operability
    - 14.10.9.2.3. Comprehensibility
    - 14.10.9.2.4. Robustness
- 14.10.10. Description of the Web Accessibility Compliance Process
- 14.10.11. Compliance Levels
- 14.10.12. Compliance Criteria
- 14.10.13. Compliance Requirements
- 14.10.13. Web Site Accessibility Evaluation Methodology

Module 15. Web Server Computing

15.1. Introduction to Server-Side Programming: PHP
- 15.1.1. Server-Side Programming Basics
- 15.1.2. Basic PHP Syntax
- 15.1.3. HTML Content Generation with PHP
- 15.1.4. Development and Testing Environments: XAMPP

15.2. Advanced PHP
- 15.2.1. Control Structures with PHP
- 15.2.2. PHP Functions
- 15.2.3. *Array* Handling in PHP
- 15.2.4. String Handling with PHP
- 15.2.5. Object Orientation in PHP

15.3. Data Models
- 15.3.1. Concept of Data. Life Cycle of Data
- 15.3.2. Types of Data
  - 15.3.2.1. Basic
  - 15.3.2.2. Records
  - 15.3.2.3. Dynamics

15.4. Relational Model
- 15.4.1. Description
- 15.4.2. Entities and Types of Entities
- 15.4.3. Data Elements. Attributes
- 15.4.4. Relationships: Types, Subtypes, Cardinality
- 15.4.5. Keys Types of Keys
- 15.4.6. Normalization. Normal Forms

15.5. Construction of the Logical Data Model
- 15.5.1. Specification of Tables
- 15.5.2. Definition of Columns
- 15.5.3. Key Specification
- 15.5.4. Conversion to Normal Forms. Dependency

15.6. The Physical Data Model. Data Files
- 15.6.1. Description of Data Files
- 15.6.2. Types of Files
- 15.6.3. Access Modes
- 15.6.4. File Organization

15.7. Database Access from PHP
- 15.7.1. Introduction to MariaDB
- 15.7.2. Working with a MariaDB Database: the SQL Language
- 15.7.3. Accessing the MariaDB Database from PHP
- 15.7.4. Introduction to MySql
- 15.7.5. Working with a MySql Database: The SQL language
- 15.7.6. Accessing MySql Database from PHP

15.8. Client Interaction from PHP
- 15.8.1. PHP Forms
- 15.8.2. Cookies
- 15.8.3. Session Management

15.9. Web Application Architecture
- 15.9.1. The Controller View Model Pattern
- 15.9.2. Controller
- 15.9.3. Models
- 15.9.4. View

15.10. Introduction to Web Services
- 15.10.1. Introduction to XML
- 15.10.2. Service-Oriented Architecture (SOA): Web services
- 15.10.3. Creation of SOAP and REST Web Services
- 15.10.4. The SOAP Protocol
- 15.10.5. The REST Protocol

Module 16. Safety Management

16.1. Information Security
- 16.1.1. Introduction
- 16.1.2. Information Security Involves Confidentiality, Integrity and Availability
- 16.1.3. Safety is an Economic Issue
- 16.1.4. Safety is a Process
- 16.1.5. Classification of Information
- 16.1.6. Information Security Involves Risk Management
- 16.1.7. Security is Articulated with Security Controls
- 16.1.8. Security is both Physical and Logical
- 16.1.9. Safety Involves People

16.2. The Information Security Professional
- 16.2.1. Introduction
- 16.2.2. Information Security as a Profession
- 16.2.3. Certifications (ISC)2
- 16.2.4. The ISO 27001 Standard
- 16.2.5. Best Security Practices in IT Service Management
- 16.2.6. Information Security Maturity Models
- 16.2.7. Other Certifications, Standards and Professional Resources

16.3. Access Control
- 16.3.1. Introduction
- 16.3.2. Access Control Requirements
- 16.3.3. Authentication Mechanisms
- 16.3.4. Authorization Methods
- 16.3.5. Access Accounting and Auditing
- 16.3.6. Triple A Technologies

16.4. Information Security Programs, Processes and Policies
- 16.4.1. Introduction
- 16.4.2. Security Management Programs
- 16.4.3. Risk Management
- 16.4.4. Design of Security Policies

16.5. Business Continuity Plans
- 16.5.1. Introduction to BCPs
- 16.5.2. Phase I and II
- 16.5.3. Phase III and IV
- 16.5.4. Maintenance of the BCP

16.6. Procedures for the Correct Protection of the Company
- 16.6.1. DMZ Networks
- 16.6.2. Intrusion Detection Systems
- 16.6.3. Access Control Lists
- 16.6.4. Learning from the Attacker: Honeypot

16.7. Security Architecture. Prevention
- 16.7.1. Overview. Activities and Layer Model
- 16.7.2. Perimeter Defence (Firewalls, WAFs, WAFs, IPS, etc.)
- 16.7.3. Endpoint Defence (Equipment, Servers and Services)

16.8. Security Architecture Detection
- 16.8.1. Overview Detection and Monitoring
- 16.8.2. Logs, Encrypted Traffic Breaking, Recording and Siems
- 16.8.3. Alerts and Intelligence

16.9. Security Architecture Reaction
- 16.9.1. Reaction Products, Services and Resources
- 16.9.2. Information Security
- 16.9.3. CERTS and CSIRTs

16.10. Security Architecture Recovery
- 16.10.1. Resilience, Concepts, Business Requirements and Regulations
- 16.10.2. IT Resilience Solutions
- 16.10.3. Crisis Management and Governance

Module 17. Software Security

17.1. Software Security Problems
- 17.1.1. Introduction to the Problem of Software Security
- 17.1.2. Vulnerabilities and their Classification
- 17.1.3. Secure Software Properties
- 17.1.4. References

17.2. Software Safety Design Principles
- 17.2.1. Introduction
- 17.2.2. Software Safety Design Principles
- 17.2.3. Types of S-SDLC
- 17.2.4. Software Safety in S-SDLC Phases
- 17.2.5. Methodologies and Standards
- 17.2.6. References

17.3. Software Lifecycle Safety in the Requirements and Design Phases
- 17.3.1. Introduction
- 17.3.2. Attack Modeling
- 17.3.3. Cases of Abuse
- 17.3.4. Safety Requirements Engineering
- 17.3.5. Risk Analysis Architectural
- 17.3.6. Design Patterns
- 17.3.7. References

17.4. Software Lifecycle Safety in the Coding, Testing and Operation Phases
- 17.4.1. Introduction
- 17.4.2. Risk-Based Safety Testing
- 17.4.3. Code Review
- 17.4.4. Penetration Test
- 17.4.5. Security Operations
- 17.4.6. External Review
- 17.4.7. References

17.5. Secure Coding Applications I
- 17.5.1. Introduction
- 17.5.2. Secure Coding Practices
- 17.5.3. Manipulation and Validation of Inputs
- 17.5.4. Memory Overflow
- 17.5.5. References

17.6. Secure Coding Applications II
- 17.6.1. Introduction
- 17.6.2. Integers Overflows, Truncation Errors and Problems with Type Conversions between Integers
- 17.6.3. Errors and Exceptions
- 17.6.4. Privacy and Confidentiality
- 17.6.5. Privileged Programs
- 17.6.6. References

17.7. Development and Cloud Security
- 17.7.1. Safety in Development; Methodology and Practice
- 17.7.2. PaaS, IaaS, CaaS and SaaS Models
- 17.7.3. Security in the Cloud and for Cloud Services

17.8. Encryption
- 17.8.1. Fundamentals of Cryptology
- 17.8.2. Symmetric and Asymmetric Encryption
- 17.8.3. Encryption at Rest and in Transit

17.9. Security Automation and Orchestration (SOAR)
- 17.9.1. Complexity of Manual Processing; Need to Automate Tasks
- 17.9.2. Products and Services
- 17.9.3. SOAR Architecture

17.10. Telework Safety
- 17.10.1. Need and Scenarios
- 17.10.2. Products and Services
- 17.10.3. Telework Safety

Module 18. Web Server Administration

18.1. Introduction to Web Servers
- 18.1.1. What is a Web Server?
- 18.1.2. Architecture and Operation of a Web Server
- 18.1.3. Resources and Contents on a Web Server
- 18.1.4. Application Servers
- 18.1.5. Proxy Servers
- 18.1.6. Main Web Servers on the Market
- 18.1.7. Web Server Usage Statistics
- 18.1.8. Web Server Security
- 18.1.9. Load Balancing on Web Servers
- 18.1.10. References

18.2. HTTP Protocol Handling
- 18.2.1. Operation and Structure
- 18.2.2. Request Methods
- 18.2.3. Status Codes
- 18.2.4. Headers
- 18.2.5. Content Coding. Code Pages
- 18.2.6. Performing HTTP Requests on the Internet using a Proxy, Livehttpheadersor Similar Method, Analyzing the Protocol Used

18.3. Description of Distributed Multi-Server Architectures
- 18.3.1. 3-Layer Model
- 18.3.2. Fault Tolerance
- 18.3.3. Load Sharing
- 18.3.4. Session State Stores
- 18.3.5. Cache Stores

18.4. Internet Information Services (IIS)
- 18.4.1. What is IIS?
- 18.4.2. History and Evolution of IIS
- 18.4.3. Main Advantages and Features of IIS7 and Later Versions
- 18.4.4. IIS7 Architecture and Later Versions

18.5. IIS Installation, Administration and Configuration
- 18.5.1. Preamble
- 18.5.2. Internet Information Services (IIS) Installation
- 18.5.3. IIS Administration Tools
- 18.5.4. Web Site Creation, Configuration and Administration
- 18.5.5. Installation and Management of IIS Extensions

18.6. Advanced Security in IIS
- 18.6.1. Preamble
- 18.6.2. Authentication, Authorization, and Access Control in IIS
- 18.6.3. Configuring a Secure Website on IIS with SSL
- 18.6.4. Security Policies Implemented in IIS 8.x

18.7. Introduction to Apache
- 18.7.1. What is Apache?
- 18.7.2. Main Advantages of Apache
- 18.7.3. Main Features of Apache
- 18.7.4. Architecture

18.8. Apache Installation and Configuration
- 18.8.1. Initial Installation of Apache
- 18.8.2. Apache Configuration

18.9. Installation and Configuration of the Different Apache Modules
- 18.9.1. Apache Module Installation
- 18.9.2. Types of Modules
- 18.9.3. Secure Apache Configuration

18.10. Advanced Security
- 18.10.1. Authentication, Authorization and Access Control
- 18.10.2. Authentication Methods
- 18.10.3. Secure Apache Configuration with SSL

Module 19. Security Audit

19.1. Introduction to Information Systems in the Company
- 19.1.1. Introduction to Information Systems in the Company and the Role of IT Auditing
- 19.1.2. Definitions of "IT Audit" and "IT Internal Control"
- 19.1.3. Functions and Objectives of IT Auditing
- 19.1.4. Differences between Internal Control and IT Auditing

19.2. Internal Controls of Information Systems
- 19.2.1. Functional Flowchart of a Data Processing Center
- 19.2.2. Classification of Information Systems Controls
- 19.2.3. The Golden Rule

19.3. The Process and Phases of the Information Systems Audit
- 19.3.1. Risk Assessment and Other IT Auditing Methodologies
- 19.3.2. Execution of an Information Systems Audit. Phases of the Audit
- 19.3.3. Fundamental Skills of the Auditor of an IT System

19.4. Technical Audit of Security in Systems and Networks
- 19.4.1. Technical Security Audits. Intrusion Test. Previous Concepts
- 19.4.2. Security Audits in Systems. Support Tools
- 19.4.3. Security Audits in Networks. Support Tools

19.5. Technical Audit of Security on the Internet and in Mobile Devices
- 19.5.1. Internet Security Audit. Support Tools
- 19.5.2. Mobile Devices Security Audit. Support Tools
- 19.5.3. Annex 1. Structure of an Executive Report and Technical Report
- 19.5.4. Annex 2. Tools Inventory
- 19.5.5. Annex 3. Methods

19.6. Information Security Management System
- 19.6.1. Security of IS: Properties and Influential Factors
- 19.6.2. Business Risks and Risk Management: Implementing Controls
- 19.6.3. Information Security Management System (ISMS): Concept and Critical Success Factors
- 19.6.4. ISMS-PDCA Model
- 19.6.5. ISMS ISO-IEC 27001: Organizational Context
- 19.6.6. Context of the Organization
- 19.6.7. Leadership
- 19.6.8. Planning
- 19.6.9. Support
- 19.6.10. Operation
- 19.6.11. Performance Evaluation
- 19.6.12. Improvement
- 19.6.13. Annex to ISO 27001/ISO-IEC 27002: Objectives and Controls
- 19.6.14. ISMS Audit

19.7. Carrying Out the Audit
- 19.7.1. Procedures
- 19.7.2. Techniques

19.8. Traceability
- 19.8.1. Methods
- 19.8.2. Analysis

19.9. Copyright
- 19.9.1. Techniques
- 19.9.2. Results

19.10. Reports and Presenting Proof
- 19.10.1. Types of Reports
- 19.10.2. Analysis of Data
- 19.10.3. Presenting Proof

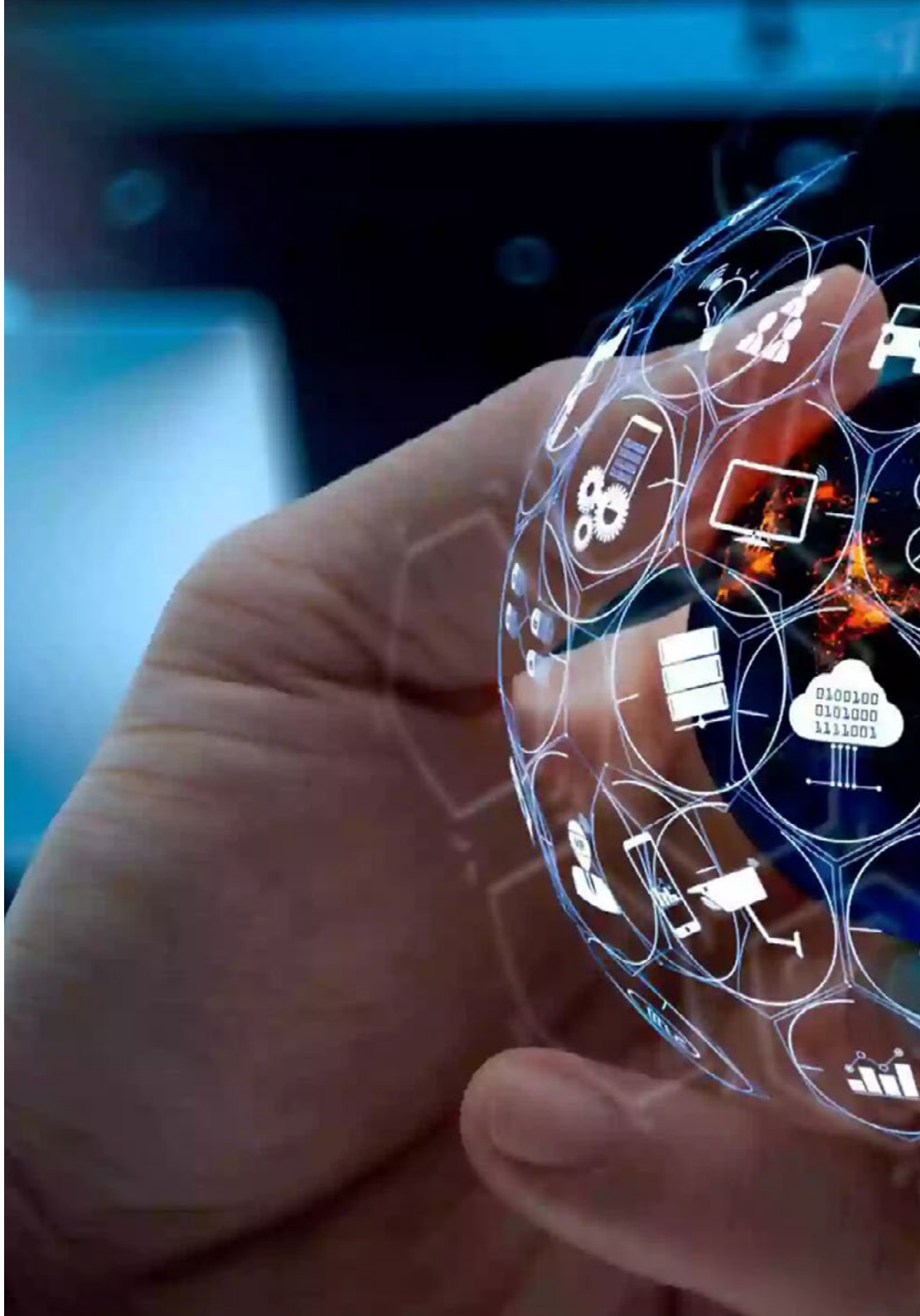
Module 20. Online Applications Security

20.1. Vulnerabilities and Security Issues in Online Applications
- 20.1.1. Introduction to Online Application Security
- 20.1.2. Security Vulnerabilities in the Design of Web Applications
- 20.1.3. Security Vulnerabilities in the Implementation of Web Applications
- 20.1.4. Security Vulnerabilities in the Deployment of Web Applications
- 20.1.5. Official Lists of Security Vulnerabilities

20.2. Policies and Standards for Online Application Security
- 20.2.1. Pillars for the Security of Online Applications
- 20.2.2. Security Policy
- 20.2.3. Information Security Management System
- 20.2.4. Secure Software Development Life Cycle
- 20.2.5. Standards for Application Security

20.3. Security in the Design of Web Applications
- 20.3.1. Introduction to Web Application Security
- 20.3.2. Security in the Design of Web Applications

20.4. Testing the Online Safety and Security of Web Applications
- 20.4.1. Web Application Security Testing and Analysis
- 20.4.2. Web Application Deployment and Production Security



20.5. Web Services Security
- 20.5.1. Introduction to Web Services Security
- 20.5.2. Web Services Security Functions and Technologies

20.6. Testing the Online Safety and Security of Web Services
- 20.6.1. Evaluation of Web Services Security
- 20.6.2. Online Protection. Firewalls and XML Gateways

20.7. Ethical Hacking, Malware and Forensics
- 20.7.1. Ethical Hacking
- 20.7.2. Malware Analysis
- 20.7.3. Forensic Analysis

20.8. Best Practices to ensure Application Security
- 20.8.1. Handbook of Best Practices in the Development of Online Applications
- 20.8.2. Handbook of Good Practices in the Implementation of Online Applications

20.9. Common Errors that Undermine Application Security
- 20.9.1. Common Errors in Development
- 20.9.2. Common Errors in Hosting
- 20.9.3. Common Production Errors

A comprehensive syllabus that will enable you to master the field of Big Data and become an architect of successful business strategies"

04

Teaching Objectives

The Advanced Master's Degree in Software Engineering and Quality aims to train professionals to be highly skilled in the design, development and management of high-quality software systems. With a particular focus on quality and the strategic management of technological projects, the program also promotes the ability to adapt to rapid advances in the industry. This ensures that students are not only prepared to face the challenges of the future, but are also capable of leading innovation in the field of software.

Turn ideas into effective technological solutions with the art of software engineering"

General Objectives

- Develop advanced skills in the design, development and maintenance of complex and scalable software systems, applying best practices and software engineering methodologies
- Train students in software quality assurance, providing them with tools and techniques to guarantee the reliability, security and performance of technological solutions
- Foster leadership in the management of technological projects, developing skills in the management of multidisciplinary teams, strategic planning and decision making in dynamic environments
- Promote the ability to adapt to rapid technological advances, through specialization in new tools, techniques and trends that allow one to stay at the forefront of software engineering
- Develop skills in quality management throughout the software life cycle, from initial planning to maintenance and continuous improvement of systems
- Strengthen communication and teamwork skills, essential for collaborating effectively with different stakeholders, managing expectations and ensuring the success of technology projects

Improve your skills and become a leader in the creation of cutting-edge technology solutions"

Specific Objectives

Module 1. Software Quality. TRL Development Levels

- Understand the different levels of technological maturity and their relationship with software quality
- Evaluate software development at each stage of the TRL and how it impacts on the final quality of the product

Module 2. Software Project Development. Functional and technical documentation

- Develop skills to create clear and detailed functional and technical documentation in software projects
- Analyze the importance of accurate documentation for project management and software quality

Module 3. Software*Testing*. Test automation

- Develop skills to design and execute automated tests in software applications
- Implement efficient testing solutions using test automation tools

Module 4. Software Project Management Methodologies. Waterfall Methodology vs Agile Methodology

- Analyze the differences between Waterfall and Agile methodologies in the management of software projects
- Evaluate the benefits and limitations of each methodology according to the type of project

Module 7 (*Test-Driven Development*). Test-Driven Software Design

- Develop skills for writing unit tests before writing the production code
- Improve software quality by implementing TDD in the development process

Module 6. DevOps. Software Quality Management

- Explore the concept of DevOps and its impact on the continuous improvement of software quality
- Learn to integrate development and operations practices to achieve a more agile and efficient software life cycle

Module 7. DevOps and Continuous Integration. Advanced Practical Solutions in Software Development

- Delve into advanced continuous integration techniques within the DevOps framework
- Implement practical continuous integration solutions to automate the software development and deployment process

Module 8. Database (DB) Design. Standardization and performance. Software Quality

- Analyze database design principles, including normalization and performance optimization
- Understand how proper database design contributes to software quality

Module 9. Design of Scalable Architectures. The Architecture in the Software Life Cycle

- Delve into the design principles of scalable architectures and their impact on software quality and performance
- Evaluate different architecture patterns for scalable software applications

Module 10. ISO, IEC 9126 Quality Criteria. Software Quality Metrics

- Understand software quality criteria according to these standards and how to apply them
- Implement quality metrics to evaluate and continuously improve software applications

Module 11. Methodologies, Development and Quality in Software Engineering

- Gaina deeper understanding of the most commonly used methodologies in Software Engineering and their relationship with Quality
- Develop a comprehensive approach that combines development, testing and quality in Software projects

Module 12. Software Project Management

- Develop skills in software project management, from planning to execution
- Manage the resources, times and risks associated with software development projects

Module 13. Software Development Platforms

- Explore the different software development platforms and their characteristics
- Evaluate development platforms based on their capabilities, flexibility and compatibility with different projects

Module 14. Web-Client Computing

- Analyze how client-side computing is used in the development of web applications
- Develop applications that take advantage of client-side computing to improve interaction and performance

Module 15. Web Server Computing

- Explore the technologies and techniques used for computing on the web server
- Understand data handling, business logic and user management on the server

Module 16. Safety Management

- Evaluate the security risks in applications and apply preventive measures
- Implement security controls in all stages of the software life cycle

Module 17. Software Security

- Explore the best security practices in Software development
- Analyze the most common software vulnerabilities and learn how to mitigate them

Module 18. Web Server Administration

- Understand the role of web servers in the development and deployment of applications
- Develop skills in the administration and maintenance of web servers

Module 19. Security Audit

- Evaluate the security of systems by means of audits and penetration tests
- Implement continuous audit processes to improve software security

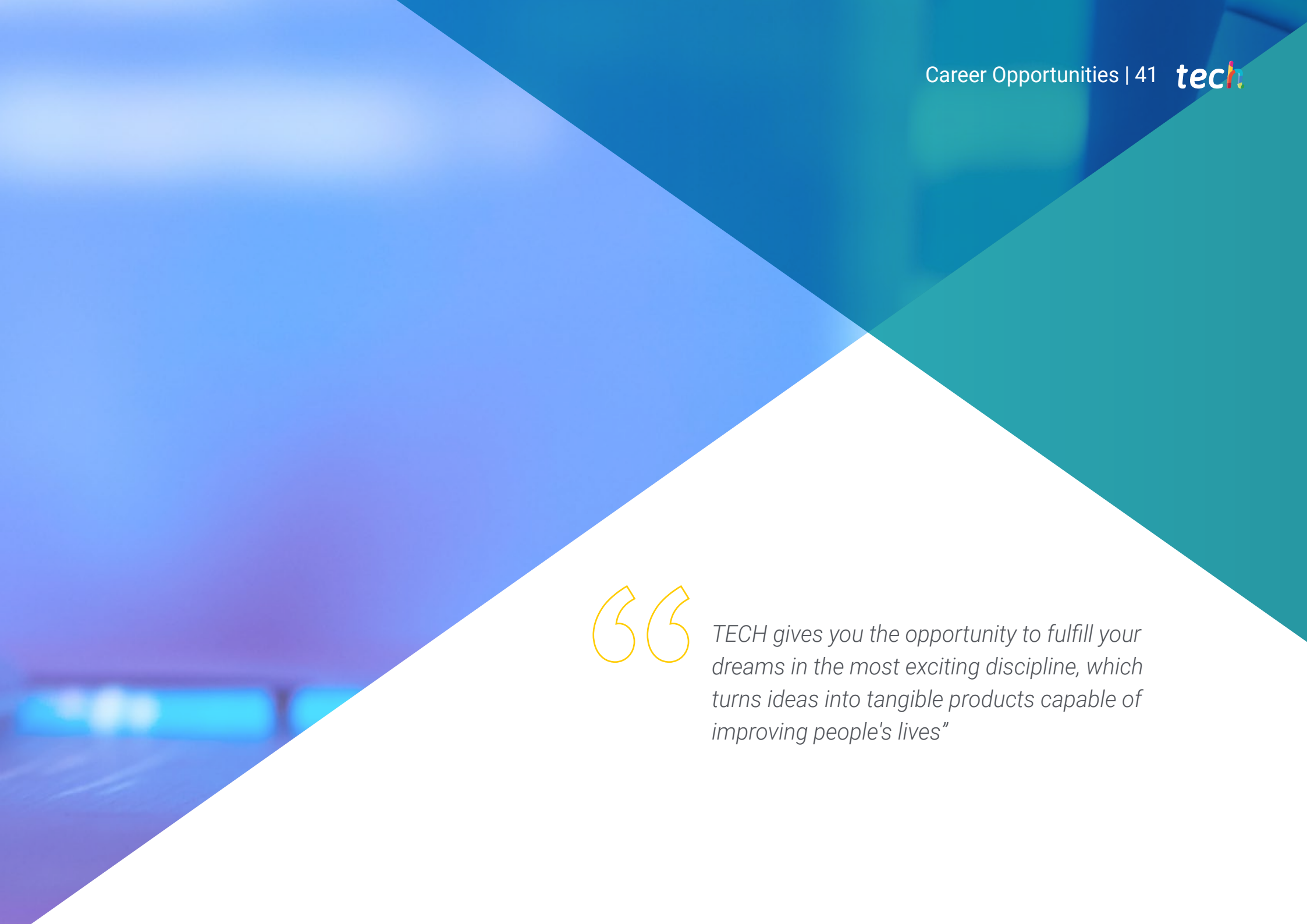
Module 20. Online Applications Security

- Implement solutions to protect online applications against external and internal threats
- Establish security and auditing policies to guarantee the integrity of online applications

05

Career Opportunities

Graduates will be highly qualified to take on leadership roles in the development, implementation and management of high-quality software. Thanks to their advanced specialization in key areas such as software architecture, online application security, technology project management and agile methodologies, they will be able to lead development teams. In addition, their preparation will enable them to occupy management positions in software projects, while their ability to innovate and lead multidisciplinary teams will enable them to face the challenges of the digital environment and contribute significantly to the success of any organization.



TECH gives you the opportunity to fulfill your dreams in the most exciting discipline, which turns ideas into tangible products capable of improving people's lives"

Graduate Profile

The profile of the graduate of the Advanced Master's Degree in Software Engineering and Quality is aimed at training highly qualified professionals, capable of leading and managing high-impact technology projects. Ensuring quality, security and efficiency in all phases of software development, they will master both agile and traditional methodologies. In addition, they will be qualified to design and develop scalable, efficient and secure software systems, applying international quality standards and advanced methodologies such as DevOps and continuous integration.

Become an expert who guarantees the success of companies, at the world's largest online university.

- **Software and Systems Security:** Competence in the implementation of advanced security practices, including data protection and vulnerability management in online applications
- **Software Quality Assurance:** Ability to apply international standards (ISO, IEC 9126) and automated testing tools to ensure software reliability and performance
- **Development of Scalable Architectures:** Ability to design and build software systems that can grow and adapt to market demands without compromising quality or security
- **Continuous Integration and DevOps:** Ability to implement and manage continuous integration processes, ensuring efficient and uninterrupted delivery of new software functionality.

After completing the Advanced Master's Degree, you will be able to apply your knowledge and skills in the following roles:

1. **Chief Technology Officer (CTO):** Responsible for the strategic management of technology in a company, leading development teams and overseeing the implementation of innovative technological solutions.
2. **Software Quality Manager:** Responsible for overseeing and ensuring that software processes and products meet established quality standards, leading continuous improvement initiatives and software testing.
3. **Software Architect:** Principal designer of the structure and architecture of complex software systems, ensuring that they are scalable, secure and efficient.
4. **Software Project Leader:** Responsible for the planning, execution and delivery of software projects, managing multidisciplinary teams and ensuring that projects are completed on time, on budget and to the appropriate quality standards.
5. **Computer Security Specialist:** Responsible for protecting applications, infrastructures and data from cyber threats, implementing effective security strategies and policies
6. **Software Security Auditor:** Carries out exhaustive audits to identify vulnerabilities in applications and systems, proposing improvements and solutions to guarantee software security

"If you want to make a difference in the digital world, choose this path that will specialize you as an expert in the creation of quality software"

06

Study Methodology

TECH is the world's first university to combine the **case study** methodology with **Relearning**, a 100% online learning system based on guided repetition.

This disruptive pedagogical strategy has been conceived to offer professionals the opportunity to update their knowledge and develop their skills in an intensive and rigorous way. A learning model that places students at the center of the educational process giving them the leading role, adapting to their needs and leaving aside more conventional methodologies.

TECH will prepare you to face new challenges in uncertain environments and achieve success in your career"

The student: the priority of all TECH programs

In TECH's study methodology, the student is the main protagonist.
The teaching tools of each program have been selected taking into account the demands of time, availability and academic rigor that, today, not only students demand but also the most competitive positions in the market.

With TECH's asynchronous educational model, it is students who choose the time they dedicate to study, how they decide to establish their routines, and all this from the comfort of the electronic device of their choice. The student will not have to participate in live classes, which in many cases they will not be able to attend. The learning activities will be done when it is convenient for them. They can always decide when and from where they want to study.

"At TECH you will NOT have live classes (which you might not be able to attend)"

The most comprehensive study plans at the international level

TECH is distinguished by offering the most complete academic itineraries on the university scene. This comprehensiveness is achieved through the creation of syllabi that not only cover the essential knowledge, but also the most recent innovations in each area.

By being constantly up to date, these programs allow students to keep up with market changes and acquire the skills most valued by employers. In this way, those who complete their studies at TECH receive a comprehensive education that provides them with a notable competitive advantage to further their careers.

And what's more, they will be able to do so from any device, pc, tablet or smartphone.

> *TECH's model is asynchronous, so it allows you to study with your pc, tablet or your smartphone wherever you want, whenever you want and for as long as you want"*

Case Studies and Case Method

The case method has been the learning system most used by the world's best business schools. Developed in 1912 so that law students would not only learn the law based on theoretical content, its function was also to present them with real complex situations. In this way, they could make informed decisions and value judgments about how to resolve them. In 1924, Harvard adopted it as a standard teaching method.

With this teaching model, it is students themselves who build their professional competence through strategies such as Learning by Doing or Design Thinking, used by other renowned institutions such as Yale or Stanford.

This action-oriented method will be applied throughout the entire academic itinerary that the student undertakes with TECH. Students will be confronted with multiple real-life situations and will have to integrate knowledge, research, discuss and defend their ideas and decisions. All this with the premise of answering the question of how they would act when facing specific events of complexity in their daily work.

Relearning Methodology

At TECH, case studies are enhanced with the best 100% online teaching method: Relearning.

This method breaks with traditional teaching techniques to put the student at the center of the equation, providing the best content in different formats. In this way, it manages to review and reiterate the key concepts of each subject and learn to apply them in a real context.

In the same line, and according to multiple scientific researches, reiteration is the best way to learn. For this reason, TECH offers between 8 and 16 repetitions of each key concept within the same lesson, presented in a different way, with the objective of ensuring that the knowledge is completely consolidated during the study process.

Relearning will allow you to learn with less effort and better performance, involving you more in your specialization, developing a critical mindset, defending arguments, and contrasting opinions: a direct equation to success.

A 100% online Virtual Campus with the best teaching resources

In order to apply its methodology effectively, TECH focuses on providing graduates with teaching materials in different formats: texts, interactive videos, illustrations and knowledge maps, among others. All of them are designed by qualified teachers who focus their work on combining real cases with the resolution of complex situations through simulation, the study of contexts applied to each professional career and learning based on repetition, through audios, presentations, animations, images, etc.

The latest scientific evidence in the field of Neuroscience points to the importance of taking into account the place and context where the content is accessed before starting a new learning process. Being able to adjust these variables in a personalized way helps people to remember and store knowledge in the hippocampus to retain it in the long term. This is a model called Neurocognitive context-dependent e-learning that is consciously applied in this university qualification.

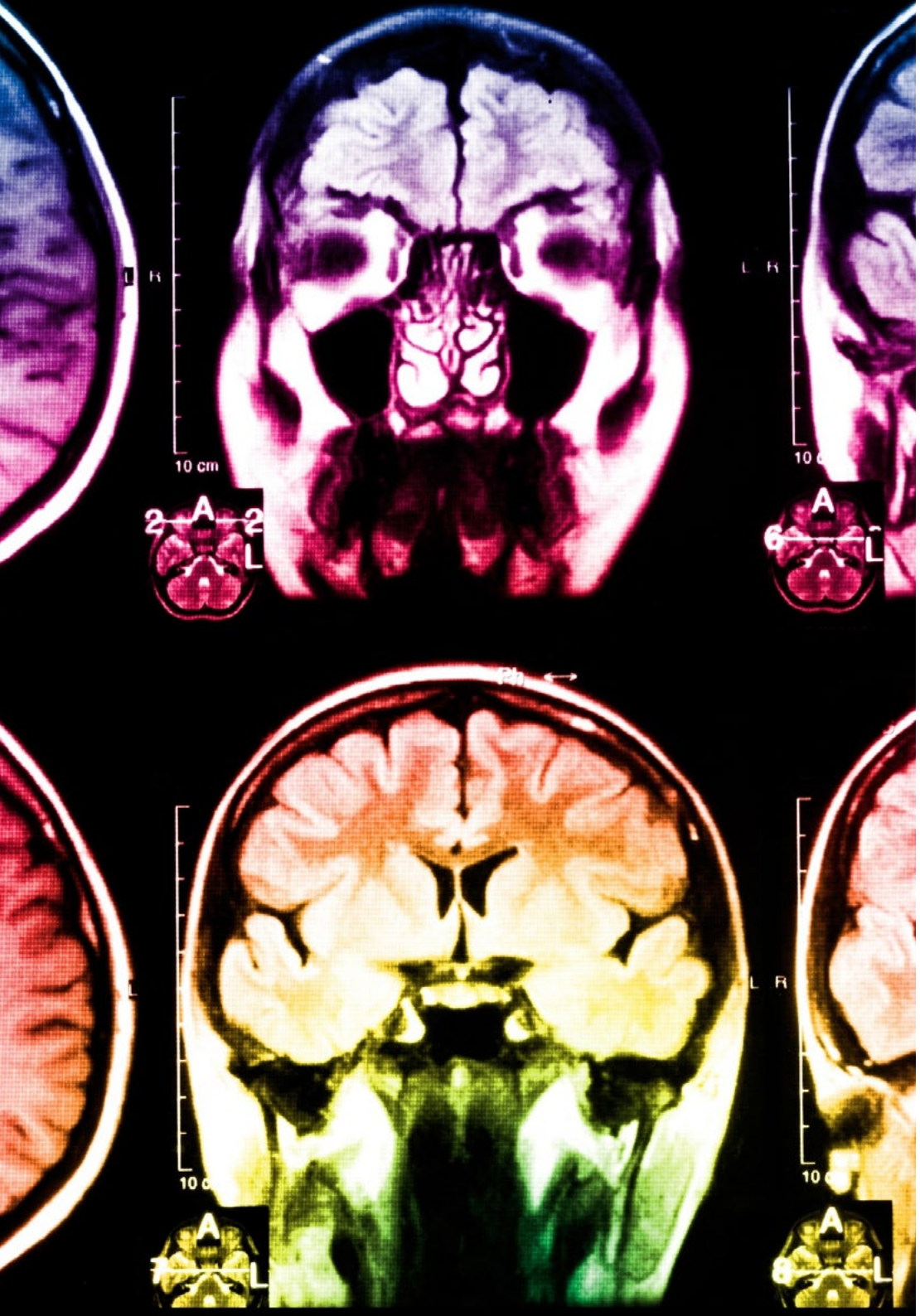
In order to facilitate tutor-student contact as much as possible, you will have a wide range of communication possibilities, both in real time and delayed (internal messaging, telephone answering service, email contact with the technical secretary, chat and videoconferences).

Likewise, this very complete Virtual Campus will allow TECH students to organize their study schedules according to their personal availability or work obligations. In this way, they will have global control of the academic content and teaching tools, based on their fast-paced professional update.

The online study mode of this program will allow you to organize your time and learning pace, adapting it to your schedule"

The effectiveness of the method is justified by four fundamental achievements:

1. Students who follow this method not only achieve the assimilation of concepts, but also a development of their mental capacity, through exercises that assess real situations and the application of knowledge.
2. Learning is solidly translated into practical skills that allow the student to better integrate into the real world.
3. Ideas and concepts are understood more efficiently, given that the example situations are based on real-life.
4. Students like to feel that the effort they put into their studies is worthwhile. This then translates into a greater interest in learning and more time dedicated to working on the course.



The university methodology top-rated by its students

The results of this innovative teaching model can be seen in the overall satisfaction levels of TECH graduates.

The students' assessment of the quality of teaching, quality of materials, course structure and objectives is excellent. Not surprisingly, the institution became the best rated university by its students on the Global Score review platform, obtaining a 4.9 out of 5.

Access the study contents from any device with an Internet connection (computer, tablet, smartphone) thanks to the fact that TECH is at the forefront of technology and teaching.

You will be able to learn with the advantages that come with having access to simulated learning environments and the learning by observation approach, that is, Learning from an expert.

As such, the best educational materials, thoroughly prepared, will be available in this program:

Study Material

All teaching material is produced by the specialists who teach the course, specifically for the course, so that the teaching content is highly specific and precise.

This content is then adapted in an audiovisual format that will create our way of working online, with the latest techniques that allow us to offer you high quality in all of the material that we provide you with.

Practicing Skills and Abilities

You will carry out activities to develop specific competencies and skills in each thematic field. Exercises and activities to acquire and develop the skills and abilities that a specialist needs to develop within the framework of the globalization we live in.

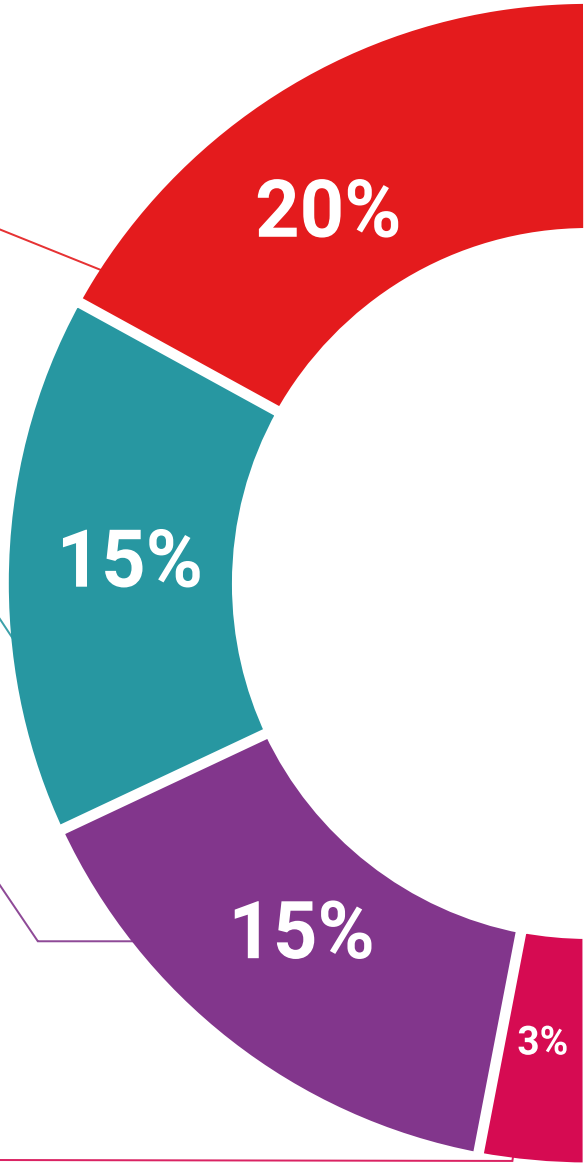
Interactive Summaries

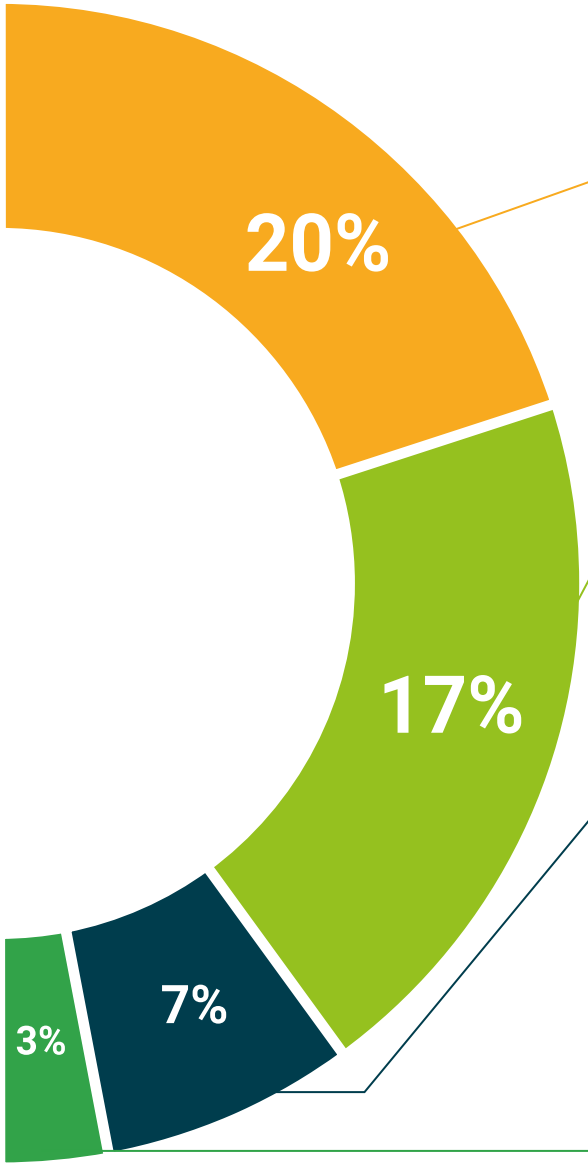
We present the contents attractively and dynamically in multimedia lessons that include audio, videos, images, diagrams, and concept maps in order to reinforce knowledge.

This exclusive educational system for presenting multimedia content was awarded by Microsoft as a "European Success Story".

Additional Reading

Recent articles, consensus documents, international guides... In our virtual library you will have access to everything you need to complete your education.





Case Studies

Students will complete a selection of the best case studies in the field. Cases that are presented, analyzed, and supervised by the best specialists in the world.

Testing & Retesting

We periodically assess and re-assess your knowledge throughout the program. We do this on 3 of the 4 levels of Miller's Pyramid.

Classes

There is scientific evidence suggesting that observing third-party experts can be useful.

Learning from an expert strengthens knowledge and memory, and generates confidence for future difficult decisions.

Quick Action Guides

TECH offers the most relevant contents of the course in the form of worksheets or quick action guides. A synthetic, practical and effective way to help students progress in their learning.

07

Teaching Staff

The management and teaching of this Advanced Master's Degree in Software Engineering and Quality is carried out by a team of engineering experts with many years of experience in the management and development of technical and specialized projects. Their professional background brings to this program a boost in quality that will be reflected in a better contextualization of the content by the graduate, as well as the implementation to the academic experience of real and simulated case studies, but always aimed at offering a 100% online, dynamic and avant-garde program based on the immediate reality of the sector.

Join the most in-demand field for software engineering and quality experts and take this opportunity to become one of them"

International Guest Director

Darren Pulsipher is a highly experienced **software architect**, an innovator with an outstanding international track record in **software and firmware development**. In fact, he possesses highly developed **communication**, **project management** and **business** skills, which have enabled him to lead major global initiatives.

He has also held senior positions of great responsibility throughout his career, such as **Chief Solution Architect for the Public Sector** at **Intel Corporation**, where he has promoted **modern business**, **processes and technologies** for customers, partners and users in the **public sector**. In addition, he founded **Yoly Inc.** where he has also served as **CEO**, working to develop a **social network aggregation and diagnostic tool** based on **Software as a Service (SaaS)**, using ***Big Data*** and **Web 2.0** technologies.

Additionally, he has served in other companies, as **Senior Director of Engineering**, at **Dell Technologies**, where he led the ***Big Data*** in the **Cloud Business Unit**, leading teams in the **United States** and **China** for the management of large projects and the restructuring of business divisions for their successful integration. He has also worked as **Chief Information Officer** at **XanGo**, where he managed projects such as ***Help Desk support***, **production support** and **solution development**.

Among the many specialties in which he is an expert, ***Edge to Cloud*** technology, **cybersecurity**, **Generative Artificial Intelligence**, **software development**, **networking technology**, **cloud-native development** and the **container ecosystem** stand out. Knowledge he has shared through the ***"Embracing Digital Transformation"*** **podcast** and **weekly newsletter**, which he produced and hosted, helping organizations successfully navigate **digital transformation** by leveraging **people**, **processes** and **technology**.

Mr. Pulsipher, Darren

- Chief Solution Architect for Public Sector at Intel, California, United States
- Host and Producer of "Embracing Digital Transformation", California
- Founder and CEO at Yoly Inc., Arkansas
- Senior Director of Engineering at Dell Technologies, Arkansas
- Chief Information Technology Officer, XanGo, Utah
- Senior Architect at Cadence Design Systems, California
- Senior Project Process Manager at Lucent Technologies, California
- Software Engineer at Cemax-Icon, California
- Software Engineer at ISG Technologies, Canada
- MBA in Technology Management from the University of Phoenix
- B.S. in Computer Science and Electrical Engineering from Brigham Young University

Thanks to TECH you will be able to learn with the best professionals in the world"

Management

Mr. Molina Molina, Jerónimo

- Head of Artificial Intelligence at Helphone
- AI Engineer & Software Architect at NASSAT, Internet Satellite in Motion
- Senior Consultant at Hexa Engineer
- Artificial Intelligence Introducer (ML and CV)
- Expert in Artificial Intelligence Based Solutions in the fields of Computer Vision, ML/DL and NLP
- University Expert in Business Creation and Development at Bancaixa and Fundeun
- Computer Engineer by the University of Alicante
- Master's Degree in Artificial Intelligence from the Catholic University of Avila
- Executive MBA at the European Business Campus Forum

Professors

Ms. Rodríguez Míguez, Cándida

- Junior Application Developer at Getronics
- Co-founder and City Leader of the Galicia AI network
- Junior Software Engineer at Indra
- Web Developer at EDISA
- Degree in Computer Engineering from the University of Vigo
- Master's Degree in Computer Engineering from the University of Vigo

Mr. Pi Morell, Oriol

- Functional Analyst at Fihoca
- Hosting and Mail Product Owner at CDmon
- Functional Analyst and Software Engineer at Atmira and Capgemini
- Lecturer at Capgemini, Forms Capgemini and Atmira
- Degree in Technical Engineering in Computer Management from the Autonomous University of Barcelona
- Master's Degree in Artificial Intelligence from the Catholic University of Avila
- MBA in Business Management and Administration from IMF Smart Education
- Master's Degree in Information Systems Management from IMF Smart Education
- Postgraduate degree in Design Patterns from the Open University of Catalonia

Mr. Martínez Calvo, Francisco Javier

- Industrial Technical Engineer specialized in Electricity and Electronics
- Software Technician at HEXA Ingenieros
- Senior .Net Developer / Net Solutions Architect at Everis
- Software Analyst/Architect at LaLiga
- Microsoft On-site Engineer at BBVA
- Freelance Technical-Informatics Consultant
- Trainer in Visual Studio, SqlServer, CCNA (Cisco Routers and Switch), PHP and .Net Web Programming in several centers (Salesianos, Maforem, Dreamsoft)
- Industrial Technical Engineer with Specialization in Electricity, Industrial Electronics
- Cibernos Master's Degree in .NET, MCAD
- Master's Degree Eidos in Advanced Programming, Expert Level
- Master's Degree in Web with Dreamweaver, Fireworks, Flash and ActionScript Certifications, MX Versions

Mr. Tenrero Morán, Marcos

- DevOps Engineer at Allot Communications
- Application Lifecycle Management Manager at Cegid Meta4
- QA Automation Engineer at Cegid Meta4
- Master's Degree in Android Application Development at Galileo University. Guatemala.
- Master's Degree in Cloud Services Development, Node.Js, JavaScript, HTML5 from the Polytechnic University of Madrid
- Web Development with Angular-CLI (4), Ionic and Node.Js, Meta4 from the Rey Juan Carlos Univeristy
- Degree in Computer Engineering from the Rey Juan Carlos Univeristy

Ms. Acebes Tamargo, Patricia

- Consultant specialized in Big Data
- Operations Department, working with Elasticsearch and Kivana at Sirt
- Online Researcher in Human Factor and AI Aplications at CTIC Technology Center
- Online Researcher in Business Unit at CTIC Technology Center
- Digital Health and Active Aging Department at CTIC Technology Center
- Data Science Department at CTIC Technology Center
- PhD in Computer Science in Artificial Intelligence from the Polytechnic University of Valencia
- Degree in Economics from the University of Oviedo
- Master's Degree in Data Analysis UCJC
- Master's Degree in Artificial Intelligence Research UNED
- Master's Degree in Blockchain, Smart Contracts and Cryptocurrencies from the University of Alcalá
- Postgraduate Degree in Blockchain Engineering by EADA
- Master's Degree in Economics, Instruments, Economic Analysis from the University of Oviedo
- Master's Degree in Taxation from the Economists Association

08

Certificate

The Advanced Master's Degree in Software Engineering and Quality guarantees students, in addition to the most rigorous and up-to-date education, access to an Advanced Master's Degree diploma issued by TECH Global University.

Successfully complete this program and receive your university qualification without having to travel or fill out laborious paperwork"

This private qualification will allow you to obtain an **Advanced Master's Degree diploma in Software Engineering and Quality** endorsed by **TECH Global University,** he world's largest online university.

TECH Global University, is an official European University publicly recognized by the Government of Andorra ***(official bulletin)***. Andorra is part of the European Higher Education Area (EHEA) since 2003. The EHEA is an initiative promoted by the European Union that aims to organize the international training framework and harmonize the higher education systems of the member countries of this space. The project promotes common values, the implementation of collaborative tools and strengthening its quality assurance mechanisms to enhance collaboration and mobility among students, researchers and academics.

This **TECH Global University** private qualification, is a European program of continuing education and professional updating that guarantees the acquisition of competencies in its area of knowledge, providing a high curricular value to the student who completes the program.

TECH is a member of the **American Society for Engineering Education (ASEE)**, a society composed of leading international figures in engineering. This distinction strengthens its leadership in academic and technological development in engineering.

Accreditation/Membership ASEE

Title: **Advanced Master's Degree in Software Engineering and Quality**
Modality: **online**
Duration: **2 years**
Accreditation: **120 ECTS**

tech global university

Mr./Ms. _____ with identification document _____
has successfully passed and obtained the title of:

Advanced Master's Degree in Software Engineering and Quality

This is a private qualification of 3,600 hours of duration equivalent to 120 ECTS, with a start date of dd/mm/yyyy and an end date of dd/mm/yyyy.

TECH Global University is a university officially recognized by the Government of Andorra on the 31st of January of 2024, which belongs to the European Higher Education Area (EHEA).

In Andorra la Vella, on the 28th of February of 2024

Dr. Pedro Navarro Illana
Dean

Unique TECH Code: AFWORD23S techtitute.com/certificates

Advanced Master's Degree in Software Engineering and Quality

General Structure of the Syllabus

Year	Subject	ECTS	Type
1º	Software Quality. TRL Development Levels	6	CO
1º	Software Project Development. Functional and Technical Documentation	6	CO
1º	SoftwareTesting. Test Automation	6	CO
1º	Software Project Management Methodologies. Waterfall	6	CO
1º	Methodology vs Agile Methodology	6	CO
1º	(Test-Driven Development). Test-Driven Software Design	6	CO
1º	DevOps. Software Quality Management	6	CO
1º	DevOps and Continuous Integration. Advanced Practical Solutions in Software Development	6	CO
1º	Database (DB) Design. Standardization and performance. Software Quality	6	CO
1º	Design of Scalable Architectures. The Architecture in the Software Life Cycle	6	CO
1º	ISO, IEC 9126 Quality Criteria. Software Quality Metrics	6	CO
2º	Methodologies, Development and Quality in Software Engineering	6	CO
2º	Software Project Management	6	CO
2º	Software Development Platforms	6	CO
2º	Web-Client Computing	6	CO
2º	Web Server Computing	6	CO
2º	Safety Management	6	CO
2º	Software Security	6	CO
2º	Web Server Administration	6	CO
2º	Security Audit	6	CO
2º	Online Applications Security	6	CO
2º			
2º			

Dr. Pedro Navarro Illana
Rector

tech global university

*Apostille Convention. In the event that the student wishes to have their paper diploma issued with an apostille, TECH Global University will make the necessary arrangements to obtain it, at an additional cost.

tech global university

Advanced Master's Degree
Software Engineering and Quality

- Modality: **online**
- Duration: **2 years**
- Certificate: **TECH Global University**
- Accreditation: **120 ECTS**
- Schedule: **at your own pace**
- Exams: **online**

Advanced Master's Degree

Software Engineering and Quality

Accreditation/Membership

ASEE American Society for Engineering Education

tech global university