Advanced Master's Degree
Software Engineering

tech global university

tech global university

Advanced Master's Degree

Software Engineering

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

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

Index

01

Introduction

The demand for software in recent years has skyrocketed. With the emergence of new digital platforms, more sophisticated hardware and ever-increasing virtualization of everyday processes, software engineers face new challenges all the time. The public is becoming more and more accustomed to new technologies and their demands are becoming greater, so software development professionals must adapt to these demands and create products that live up to the market's expectations. This requires a high technical level in multiple fields of computer knowledge.

You will play a key role in the technological future of many companies. Specialize in Software Engineering and start developing the systems that will make a difference"

The technology industry is one of the most relevant industries today, as almost everyone interacts with some kind of digital device on a daily basis. In this context, software engineers are the first line of battle in the whole technological development process, since they are the ones who constantly have to be updating systems, developing new ones and offering intelligent solutions to the problems that arise. Seen in this way, computer engineering professionals must be highly decisive people, with great technical knowledge and an outstanding ability to adapt to all types of development and environments.

With this objective in mind, TECH has designed this Advanced Master's Degree in Software Engineering, offering a complete and high-level training to all developers who want to specialize their career and direct it to the creation of systems. On the one hand, the program deals with the different methodologies to create and manage a software development project, as well as all the aspects to take into account regarding computation, requirements and platforms. On the other hand, the security of the software itself as well as the information systems and work environment used during the process is also emphasized. Upon graduation, the student will have all the necessary knowledge to be an effective and highly competent Software Engineering expert.

In addition, one of the main advantages of this program is that its 100% online. This means that the student does not have to adapt to fixed schedules and is not obliged to attend a specific physical center. Thus, the student has the freedom to manage the study of the subject of his choice, at his own pace and taking into account his obligations, planning his schedule as he sees fit.

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

- The development of case studies presented by experts in software development
- 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
- Its special emphasis on innovative methodologies in the field of Software Engineering
- 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

Can you imagine having participated in the development of Netflix? It's time to stop imagining and focus your career on the best software projects"

Your experience and expertise can make the difference in large projects involving many requirements. Don't miss the opportunity to distinguish yourself in your career and enroll now in this Advanced Master's Degree in Software Engineering"

TECH's goal is to make you a great computer engineer. You are guaranteed access to the best possible material and teaching.

Study it when, where and how you want. The program is 100% online and adapts to your needs, not the other way around.

Its teaching staff includes professionals from the field of Software Engineering, who bring their work experience to this program, as well as renowned 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 during the course. For this purpose, the professional will be assisted by an innovative interactive video system created by renowned and experienced experts.

02

Objectives

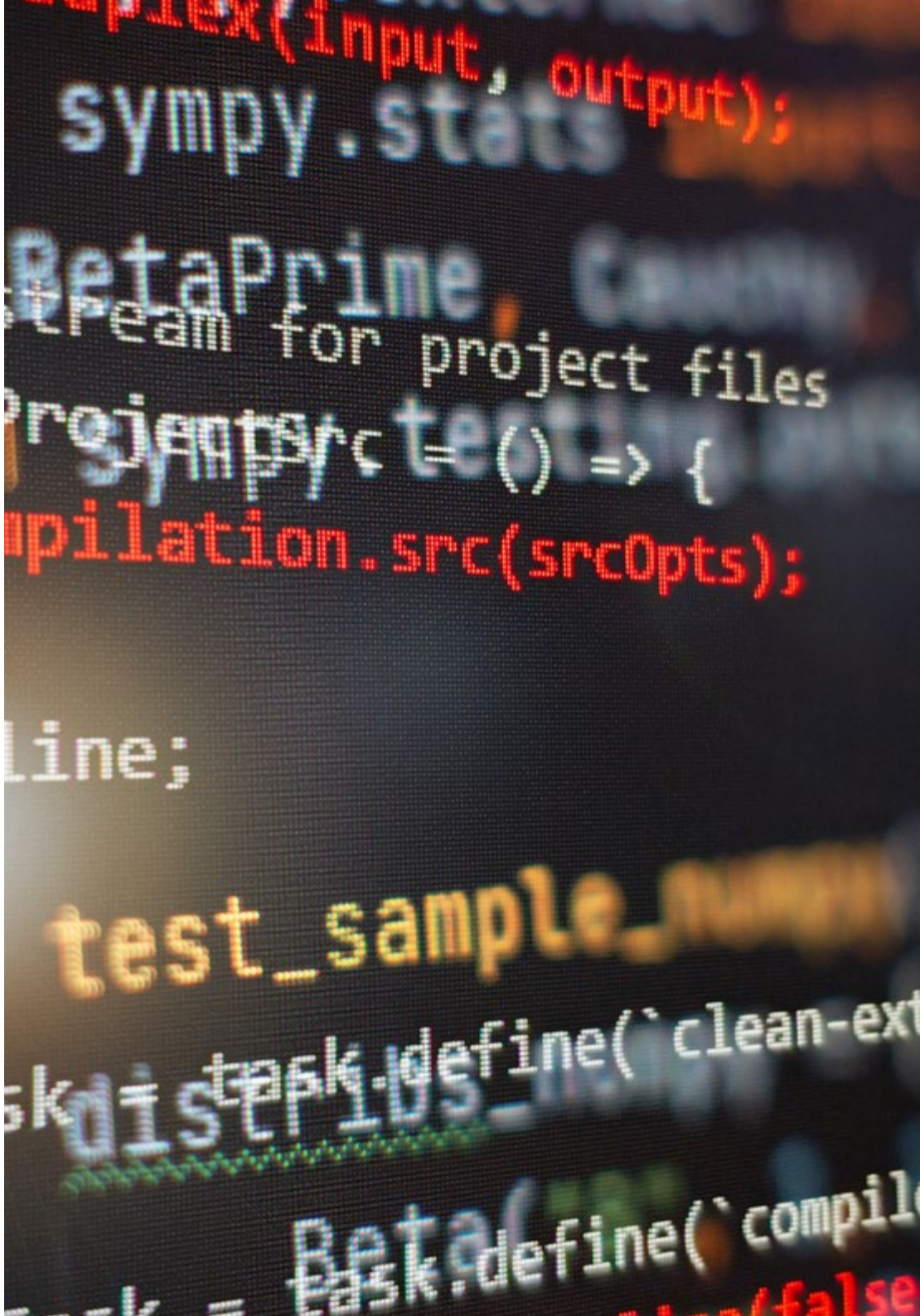
This Advanced Master's Degree in Software Engineering has been developed with the aim of providing all professionals in the IT field with the necessary advanced training to focus their careers on the development of modern software adapted to the new fluctuating realities of the market. With the highly technical knowledge taught throughout the course, the student will greatly increase his or her options for professional advancement and access to jobs in large companies in the sector.

An Advanced Master's Degree that will be the biggest positive boost you can give to your career towards professional success"

General objectives

- Acquire new necessary and demanded competences in terms of new technologies and the latest software developments
- Complement the acquired knowledge with skills in the field of computation and computer structure, including the mathematical, statistical and physical basis essential in engineering
- Expand knowledge in Software Engineering and Computer Systems with the latest developments and most innovative methodology
- Tackle complex software projects and environments, knowing how to provide intelligent solutions to diverse problems

A specialization that will help you master software development with a unique set of skills demanded by every leading company in the industry"



Specific objectives

- Know the basics of Software Engineering, as well as the set of rules or ethical principles and professional responsibility during and after development
- Understand the software development process, under the different programming models and the object oriented programming paradigm
- Understand the different types of application modeling and design patterns in the Unified Modeling Language (UML)
- Know the fundamental concepts of project management and the project management life cycle
- Understand how quality management works in projects, including planning, assurance, control, statistical concepts and available tools
- Acquire the essential knowledge related to the professional responsibility derived from project management
- Understand the different software development platforms
- Acquire the necessary knowledge for the development of applications and graphical interfaces in Java and .NET languages
- Learn Android mobile application development environments and debugging and publishing processes
- Understand cloud-based application development and determine the correct procedures for its implementation
- Understand the procedures and techniques to improve the appearance of a document written in HTML
- Acquire the necessary knowledge for the development of web client-side applications
- Develop applications with complex structures, by using the different procedures, functions and objects that integrate JavaScript
- Learn how to use the DOM programming interface for HTML and XML documents to modify their structure, style and content
- Know the concept of web usability, its advantages, principles, methods and techniques to make a web site usable by the user
- Understand the Model View Controller View (MVC) software architecture that separates an application's data, user interface, and control logic into three distinct components
- Acquire the skills for the use of web services using XML, SOA and REST
- Know the information security process, its implications on confidentiality, integrity, availability and economic costs
- Learn the use of good security practices in the management of information technology services
- Acquire the knowledge for the correct certification of security processes
- Understand authentication mechanisms and methods for access control, as well as the access audit process
- Understand security management programs, risk management and security policy design
- Learn about business continuity plans, their phases and maintenance process
- Know the procedures for the correct protection of the company through DMZ networks, the use of intrusion detection systems and other methodologies
- Understand software security issues, vulnerabilities and how they are classified
- Analyze the different web servers that are trending in today's market
- Understand the process of usage statistics and load balancing on web servers
- Acquire the knowledge required for the correct execution of the audit process and internal computer control

- Understand the concepts and processes of software design, learning also about architecture design and about component-level and pattern-based design
- Understand the different patterns of system architectures and software design, as well as the architecture of cloud applications
- Deepen the improvement of the software development process and software quality using ISO/IEC standards
- Understand the importance of requirements engineering in the software development process
- Have an in-depth knowledge of the requirements sources and requirements elicitation techniques, as they are an essential part of the process
- Understand and apply prototyping as an essential part of the development process
- Lay the foundations for forensic analysis in the world of software and computer audits
- Know the fundamental concepts of project management and the project management life cycle
- Learning schedule development for time management, budget development and risk response
- Understand how quality management works in projects, including planning, assurance, control, statistical concepts and available tools

A complete training that will take you through the knowledge you need to compete among the best."

03

Skills

Software engineers are constantly updating their knowledge as the very tools and realities in which they work are modernized on a daily basis. This requires a regular learning process for which various general skills are necessary, both in pure software development and in other disciplines such as team management. Understanding this circumstance, TECH has elaborated the Advanced Master's Degree in Software Engineering thinking of providing the student with all the possible and necessary competences to make his own continuous learning process lighter and more automatic.

With all the skills you will acquire by studying this Advanced Master's Degree in Software Engineering you will be the best candidate for any position you apply for"

General skills

- Develop a software system taking into account all stages of development, security platforms and security issues
- Correctly and professionally handle all data generated during development
- Apply the best work methodology as appropriate for the project or the people involved in the project
- Know the complete reality of Software Engineering and prevent possible risks or problems quickly and efficiently

You can take the step towards a better working future. Do it and enroll now in this Advanced Master's Degree that will open many doors for your professional career"

Specific skills

- Understand the different types of application modeling and design patterns in the Unified Modeling Language (UML)
- Understand how quality management works in projects, including planning, assurance, control, statistical concepts and available tools
- Use the necessary knowledge for the development of applications and graphical interfaces in Java and .NET languages
- Understand the procedures and techniques to improve the appearance of a document written in HTML
- Master the process of customer interactions, using forms, Cookiesand session management
- Understand authentication mechanisms and methods for access control, as well as the access audit process
- Understand the application of security in the different phases of the software life cycle
- Know the concept, operation, architecture, resources and contents of a web server
- Understand the different support tools, methodologies and subsequent analysis during internet and mobile device security auditing
- Understand the security policies and standards to be applied to online applications
- Be able to write, plan, develop and sign projects in the field of computer engineering aimed at developing or operating computer systems, services and applications
- Direct the activities of IT projects
- Be able to define, evaluate and select hardware and software platforms for the development and implementation of computer systems, services and applications
- Know how to develop using Scrum, extreme programming and reuse-based software development techniques
- Have the ability to design, develop and maintain computer systems, services and applications using software engineering methods as a tool for quality assurance
- Employ the fundamentals of symmetric cryptography and asymmetric cryptography, as well as their main algorithms
- Apply the essential concepts related to information systems in the Company, as well as identify the opportunities and needs of information systems
- Know how to develop a schedule for time, budget and risk response management
- Understand the functioning of ICT governance and management, the ISO/IEC standards that govern it and the best practices to be carried out
- Plan security management and manage the main mechanisms for the protection of information assets

04

Structure and Content

The didactic material of this Advanced Master's Degree in Software Engineering is elaborated to cover all the necessary and complementary teaching in the subject, with the most up-to-date methodologies, tools and knowledge available in the market. An extensive and exhaustive syllabus where the use of programming languages and advanced environments, web server administration and its integration in the development process or the actual management of a project are taught. This ensures the best possible opportunity for the student to specialize in Software Engineering and quickly excel in that field.

Show that your knowledge is on par with your attitude to be the best professional and add great value to your resume with this Advanced Master's Degree in Software Engineering"

Module 1. Methodologies, Development and Quality in Software Engineering

1.1. Introduction to Software Engineering
- 1.1.1. Introduction
- 1.1.2. The Software Crisis
- 1.1.3. Differences between Software Engineering and Computer Science
- 1.1.4. Ethics and Professional Responsibility in Software Engineering
- 1.1.5. Software Factories

1.2. The Software Development Process
- 1.2.1. Definition
- 1.2.2. Software Process Model
- 1.2.3. The Unified Software Development Process

1.3. Object-Oriented Software Development
- 1.3.1. Introduction
- 1.3.2. Principles of Object Orientation
- 1.3.3. Objectives Definition
- 1.3.4. Class Definition
- 1.3.5. Object-Oriented Analysis vs. Object-Oriented Design

1.4. Model-Based Software Development
- 1.4.1. The Need to Model
- 1.4.2. Software Systems Modeling
- 1.4.3. Object Modeling
- 1.4.4. UML
- 1.4.5. CASE Tools

1.5. Application Modeling and Design Patterns with UML
- 1.5.1. Advanced Requirements Modeling
- 1.5.2. Advanced Static Modeling
- 1.5.3. Advanced Dynamic Modeling
- 1.5.4. Component Modeling
- 1.5.5. Introduction to Design Patterns with UML
- 1.5.6. Adapter
- 1.5.7. Factory
- 1.5.8. Singleton
- 1.5.9. Strategy
- 1.5.10. Composite
- 1.5.11. Facade
- 1.5.12. Observer

1.6. Model-Driven Engineering
- 1.6.1. Introduction
- 1.6.2. Metamodeling of Systems
- 1.6.3. MDA
- 1.6.4. DSL
- 1.6.5. Model Refinements with OCL
- 1.6.6. Model Transformations

1.7. Ontologies in Software Engineering
- 1.7.1. Introduction
- 1.7.2. Ontology Engineering
- 1.7.3. Application of Ontologies in Software Engineering

1.8. Agile Methodologies for Software Development, Scrum
- 1.8.1. What is Software Agility?
- 1.8.2. The Agile Manifesto
- 1.8.3. The Roadmap of an Agile Project
- 1.8.4. The Product Owner
- 1.8.5. User Stories
- 1.8.6. Agile Planning and Estimating
- 1.8.7. Measurements in Agile Development
- 1.8.8. Introduction to Scrum
- 1.8.9. The Roles
- 1.8.10. The Product Backlog
- 1.8.11. The Sprint
- 1.8.12. Meetings

1.9. Lean Software Development Methodology
- 1.9.1. Introduction
- 1.9.2. Kanban

1.10. Quality and Software Process Improvement
- 1.10.1. Introduction
- 1.10.2. Software Measurement
- 1.10.3. Software Testing
- 1.10.4. Software Process Quality Model: CMMI

Module 2. Software Project Management

2.1. Fundamental Concepts of Project Management and the Project Management Lifecycle
- 2.1.1. What is a Project?
- 2.1.2. Common Methodology
- 2.1.3. What is Project Management?
- 2.1.4. What is a Project Plan?
- 2.1.5. Benefits
- 2.1.6. Project Life Cycle
- 2.1.7. Process Groups or Project Management Life Cycle
- 2.1.8. The Relationship between Process Groups and Knowledge Areas
- 2.1.9. Relationships between Product and Project Life Cycle

2.2. Start-Up and Planning
- 2.2.1. From the Idea to the Project
- 2.2.2. Development of the Project Record
- 2.2.3. Project Kick-Off Meeting
- 2.2.4. Tasks, Knowledge and Skills in the Startup Process
- 2.2.5. The Project Plan
- 2.2.6. Development of the Basic Plan. Steps
- 2.2.7. Tasks, Knowledge and Skills in the Planning Process

2.3. Stakeholders and Outreach Management
- 2.3.1. Identify Stakeholders
- 2.3.2. Develop Plan for Stakeholder Management
- 2.3.3. Manage Stakeholder Engagement
- 2.3.4. Control Stakeholder Engagement
- 2.3.5. The Objective of the Project
- 2.3.6. Scope Management and its Plan
- 2.3.7. Gathering Requirements
- 2.3.8. Define the Scope Statement
- 2.3.9. Create the WBS
- 2.3.10. Verify and Control the Scope

2.4. The Development of the Time-Schedule
- 2.4.1. Time Management and its Plan
- 2.4.2. Define Activities
- 2.4.3. Establishment of the Sequence of Activities
- 2.4.4. Estimated Resources for Activities
- 2.4.5. Estimated Duration of Activities
- 2.4.6. Development of the Time-Schedule and Calculation of the Critical Path
- 2.4.7. Schedule Control

2.5. Budget Development and Risk Response
- 2.5.1. Estimate Costs
- 2.5.2. Develop Budget and S-Curve
- 2.5.3. Cost Control and Earned Value Method
- 2.5.4. Risk Concepts
- 2.5.5. How to Perform a Risk Analysis?
- 2.5.6. The Development of the Response Plan

2.6. Quality Management
- 2.6.1. Quality Planning
- 2.6.2. Assuring Quality
- 2.6.3. Quality Control
- 2.6.4. Basic Statistical Concepts
- 2.6.5. Quality Management Tools

2.7. Communication and Human Resources
- 2.7.1. Planning Communications Management
- 2.7.2. Communications Requirements Analysis
- 2.7.3. Communication Technology
- 2.7.4. Communication Models
- 2.7.5. Communication Methods
- 2.7.6. Communications Management Plan
- 2.7.7. Manage Communications
- 2.7.8. Management of Human Resources
- 2.7.9. Main Stakeholders and their Roles in the Projects
- 2.7.10. Types of Organization
- 2.7.11. Project Organization
- 2.7.12. The Work Equipment

2.8. Procurement
- 2.8.1. The Procurement Process
- 2.8.2. Planning
- 2.8.3. Search for Suppliers and Request for Quotations
- 2.8.4. Contract Allocation
- 2.8.5. Contract Administration
- 2.8.6. Contracts
- 2.8.7. Types of Contracts
- 2.8.8. Contract Negotiation

2.9. Execution, Monitoring and Control and Closure
- 2.9.1. Process Groups
- 2.9.2. Project Execution
- 2.9.3. Project Monitoring and Control
- 2.9.4. Project Closure

2.10. Professional Responsibility
- 2.10.1. Professional Responsibility
- 2.10.2. Characteristics of Social and Professional Responsibility
- 2.10.3. Project Leader Code of Ethics
- 2.10.4. Liability vs. PMP®
- 2.10.5. Examples of Liability
- 2.10.6. Benefits of Professionalization

Module 3. Software Development Platforms

3.1. Introduction to Application Development
- 3.1.1. Desktop Applications
- 3.1.2. Programming Language
- 3.1.3. Integrated Development Environments
- 3.1.4. Web Applications
- 3.1.5. Mobile Applications
- 3.1.6. Cloud Applications

3.2. Application Development and Graphical User Interface in Java
- 3.2.1. Integrated development environments for Java
- 3.2.2. Main IDE for Java
- 3.2.3. Introduction to the Eclipse Development Platform

3.2.4. Introduction to the NetBeans Development Platform
3.2.5. Controller View Model for Graphical User Interfaces
3.2.6. Design a Graphical Interface in Eclipse
3.2.7. Design a Graphical Interface in NetBeans
3.3. Debugging and Testing in Java
3.3.1. Testing and Debugging of Java programs
3.3.2. Debugging in Eclipse
3.3.3. Debugging in NetBeans
3.4. Application Development and Graphical User Interface in. NET
3.4.1. Net Framework
3.4.2. Components of the .NET Development Platform
3.4.3. Visual Studio .NET
3.4.4. .NET tools for GUI
3.4.5. The GUI with Windows Presentation Foundation
3.4.6. Debugging and Compiling a WPF Application
3.5. Programming for .NET Networks
3.5.1. Introduction to .NET Network Programming
3.5.2. Requests and Responses in .NET
3.5.3. Use of Application Protocols in .NET
3.5.4. Security in .NET Network Programming
3.6. Mobile Application Development Environments
3.6.1. Mobile Applications
3.6.2. Android Mobile Applications
3.6.3. Steps for Development in Android
3.6.4. The IDE Android Studio
3.7. Development of Applications in the Environment Android Studio
3.7.1. Install and Start Android Studio
3.7.2. Run an Android Application
3.7.3. Development of the Graphic Interface in Android Studio
3.7.4. Starting Activities in Android Studio
3.8. Debugging and Publishing of Android Applications
3.8.1. Debugging an Application in Android Studio
3.8.2. Memorizing Applications in Android Studio
3.8.3. Publishing an Application on Google Play
3.9. Cloud Application Development
3.9.1. Cloud Computing
3.9.2. Cloud Levels: SaaS, PaaS, IaaS
3.9.3. Main Development Platforms in the Cloud
3.9.4. Bibliographical References
3.10. Introduction to Google Cloud Platform
3.10.1. Basic Concepts of Google Cloud Platform
3.10.2. Google Cloud Platform Services
3.10.3. Tools in Google Cloud Platform

Module 4. Web-Client Computing

4.1. Introduction to HTML
4.1.1. Structure of the Document
4.1.2. Color
4.1.3. Text:
4.1.4. Hypertext Links
4.1.5. Images
4.1.6. Lists
4.1.7. Tables
4.1.8. Frames
4.1.9. Forms
4.1.10. Specific Elements for Mobile Technologies
4.1.11. Obsolete Elements
4.2. Cascading Style Sheets (CSS)
4.2.1. Elements and Structure of a Cascading Style Sheet
4.2.1.1. Creation of Style Sheets
4.2.1.2. Application of Styles Selectors
4.2.1.3. Style Inheritance and Cascading
4.2.1.4. Page Formatting Using Styles
4.2.1.5. Page Structuring Using Styles. The Box Model
4.2.2. Style Design for different Devices
4.2.3. Types of Style Sheets: Static and Dynamic The Pseudo-Classes
4.2.4. Best Practices in the Use of Style Sheets

4.3. Introduction and history of JavaScript
- 4.3.1. Introduction
- 4.3.2. History of JavaScript
- 4.3.3. Development Environment to be Used

4.4. Basic Notions of Web Programming
- 4.4.1. Basic JavaScript Syntax
- 4.4.2. Primitive Data Types and Operators
- 4.4.3. Variables and Areas
- 4.4.4. Text Strings and Template Literals
- 4.4.5. Numbers and Booleans
- 4.4.6. Comparisons

4.5. Complex JavaScript Structures
- 4.5.1. Vectors or Arrays and Objects
- 4.5.2. Sets
- 4.5.3. Maps
- 4.5.4. Disjunctive
- 4.5.5. Loops

4.6. Functions and Objects
- 4.6.1. Function Definition and Invocation
- 4.6.2. Arguments
- 4.6.3. Arrow Functions
- 4.6.4. Callback Functions
- 4.6.5. Higher Order Functions
- 4.6.6. Literal Objects
- 4.6.7. The This Object
- 4.6.8. Objects as Namespaces: theMaths and Date Objects

4.7. The Document Object Model (DOM)
- 4.7.1. What is DOM?
- 4.7.2. A Bit of History
- 4.7.3. Navigation and Element Retrieval
- 4.7.4. A Virtual DOM with JSDOM
- 4.7.5. Query Selectors
- 4.7.6. Navigation using Properties
- 4.7.7. Assigning Attributes to Elements
- 4.7.8. Creation and Modification of Nodes
- 4.7.9. Updated Styling of the DOM Elements

4.8. Modern Web Development
- 4.8.1. Event-Driven Flow and Listeners
- 4.8.2. Modern Web Toolkits and Alignment Systems
- 4.8.3. Strict JavaScript Mode
- 4.8.4. More about Functions
- 4.8.5. Asynchronous Promises and Functions
- 4.8.6. Closures
- 4.8.7. Functional Programming
- 4.8.8. POO in JavaScript

4.9. Web Usability
- 4.9.1. Introduction to Usability
- 4.9.2. Definition of Usability
- 4.9.3. Importance of User-Centered Web Design
- 4.9.4. Differences Between Accessibility and Usability
- 4.9.5. Advantages and Problems in Combining Accessibility and Usability
- 4.9.6. Advantages and Difficulties in the Implementation of Usable Websites
- 4.9.7. Usability Methods
- 4.9.8. User Requirements Analysis
- 4.9.9. Conceptual Design Principles. User-Oriented Prototyping
- 4.9.10. Guidelines for the Creation of Usable Web Sites
  - 4.9.10.1. Usability Guidelines of Jakob Nielsen
  - 4.9.10.2. Usability Guidelines of Bruce Tognazzini
- 4.9.11. Usability Evaluation

4.10. Web Accessibility
- 4.10.1. Introduction
- 4.10.2. Definition of Web-Accessibility
- 4.10.3. Types of Disabilities
  - 4.10.3.1. Temporary or Permanent Disabilities
  - 4.10.3.2. Visual Impairment
  - 4.10.3.3. Hearing Impairment
  - 4.10.3.4. Motor Impairment
  - 4.10.3.5. Neurological or Cognitive Disabilities
  - 4.10.3.6. Difficulties Arising from Aging
  - 4.10.3.7. Limitations Arising from the Environment
  - 4.10.3.8. Barriers Preventing Access to the Web
- 4.10.4. Technical Aids and Support Products to Overcome Barriers
  - 4.10.4.1. Aids for the Blind
  - 4.10.4.2. Aids for Persons with Low Vision
  - 4.10.4.3. Aids for People with Color Blindness
  - 4.10.4.4. Aids for the Hearing Impaired
  - 4.10.4.5. Aids for the Motor Impaired
  - 4.10.4.6. Aids for the and Neurological Impaired
- 4.10.5. Advantages and Difficulties in the Implementation of Web Accessibility
- 4.10.6. Web Accessibility Regulations and Standards
- 4.10.7. Web Accessibility Regulatory Bodies
- 4.10.8. Comparison of Standards and Regulations
- 4.10.9. Guidelines for Compliance with Regulations and Standards
  - 4.10.9.1. Description of the Main Guidelines (Images, links, videos, etc.)
  - 4.10.9.2. Guidelines for Accessible Navigation
    - 4.10.9.2.1. Perceptibility
    - 4.10.9.2.2. Operability
    - 4.10.9.2.3. Comprehensibility
    - 4.10.9.2.4. Robustness
- 4.10.10. Description of the Web Accessibility Compliance Process
- 4.10.11. Compliance Levels
- 4.10.12. Compliance Criteria
- 4.10.13. Compliance Requirements
- 4.10.14. Web Site Accessibility Evaluation Methodology

Module 5. Web Server Computing

5.1. Introduction to Server-Side Programming: PHP
- 5.1.1. Server-Side Programming Basics
- 5.1.2. Basic PHP Syntax
- 5.1.3. HTML Content Generation with PHP
- 5.1.4. Development and Testing Environments: XAMPP

5.2. Advanced PHP
- 5.2.1. Control Structures with PHP
- 5.2.2. PHP Functions
- 5.2.3. Array Handling in PHP
- 5.2.4. String Handling with PHP
- 5.2.5. Object Orientation in PHP

5.3. Data Models
- 5.3.1. Concept of Data. Life Cycle of Data
- 5.3.2. Types of Data
  - 5.3.2.1. Basic
  - 5.3.2.2. Records
  - 5.3.2.3. Dynamics

5.4. Relational Model
- 5.4.1. Description
- 5.4.2. Entities and Types of Entities
- 5.4.3. Data Elements. Attributes
- 5.4.4. Relationships: Types, Subtypes, Cardinality
- 5.4.5. Keys Types of Keys
- 5.4.6. Normalization. Normal Shapes

5.5. Construction of the Logical Data Model
- 5.5.1. Specification of Tables
- 5.5.2. Definition of Columns
- 5.5.3. Key Specification
- 5.5.4. Conversion to Normal Shapes. Dependency

5.6. The Physical Data Model. Data Files
- 5.6.1. Description of Data Files
- 5.6.2. Types of Files
- 5.6.3. Access Modes
- 5.6.4. File Organization

5.7. Database Access from PHP
- 5.7.1. Introduction to MariaDB
- 5.7.2. Working with a MariaDBdatabase: the SQL language
- 5.7.3. Accessing MariaDB database from PHP
- 5.7.4. Introduction to MySql
- 5.7.5. Working with a MySql Database: The SQL language
- 5.7.6. Accessing MySql Database from PHP

5.8. Client Interaction from PHP
- 5.8.1. PHP Forms
- 5.8.2. Cookies
- 5.8.3. Session Management

5.9. Web Application Architecture
- 5.9.1. The Controller View Model Pattern
- 5.9.2. Controller
- 5.9.3. Models
- 5.9.4. View

5.10. Introduction to Web Services
- 5.10.1. Introduction to XML
- 5.10.2. Service-Oriented Architecture (SOA): Web services
- 5.10.3. Creation of SOAP and REST Web Services
- 5.10.4. The SOAP Protocol
- 5.10.5. The REST Protocol

Module 6. Safety Management

6.1. Information Security
- 6.1.1. Introduction
- 6.1.2. Information Security Involves Confidentiality, Integrity and Availability
- 6.1.3. Safety is an Economic Issue
- 6.1.4. Safety is a Process
- 6.1.5. Classification of Information
- 6.1.6. Information Security Involves Risk Management
- 6.1.7. Security is Articulated with Security Controls
- 6.1.8. Security is both Physical and Logical
- 6.1.9. Safety Involves People

6.2. The Information Security Professional
- 6.2.1. Introduction
- 6.2.2. Information Security as a Profession
- 6.2.3. Certifications (ISC)2
- 6.2.4. The ISO 27001 Standard
- 6.2.5. Best Security Practices in IT Service Management
- 6.2.6. Information Security Maturity Models
- 6.2.7. Other Certifications, Standards and Professional Resources

6.3. Access Control
- 6.3.1. Introduction
- 6.3.2. Access Control Requirements
- 6.3.3. Authentication Mechanisms
- 6.3.4. Authorization Methods
- 6.3.5. Access Accounting and Auditing
- 6.3.6. Triple A" Technologies

6.4. Information Security Programs, Processes and Policies
- 6.4.1. Introduction
- 6.4.2. Security Management Programs
- 6.4.3. Risk Management
- 6.4.4. Design of Security Policies

6.5. Business Continuity Plans
- 6.5.1. Introduction to BCPs
- 6.5.2. Phase I and II
- 6.5.3. Phase III and IV
- 6.5.4. Maintenance of the BCP

6.6. Procedures for the Correct Protection of the Company
- 6.6.1. DMZ Networks
- 6.6.2. Intrusion Detection Systems
- 6.6.3. Access Control Lists
- 6.6.4. Learning from the Attacker: Honeypot

6.7. Security Architecture Prevention
- 6.7.1. Overview. Activities and Layer Model
- 6.7.2. Perimeter Defense (Firewalls, WAFs, WAFs, IPS, etc.)
- 6.7.3. Endpoint Defence (Equipment, Servers and Services)

6.8. Security Architecture Detection
- 6.8.1. Overview Detection and Monitoring
- 6.8.2. Logs, Encrypted Traffic Breaking, Recording and Siems
- 6.8.3. Alerts and Intelligence

6.9. Security Architecture Reaction
- 6.9.1. Reaction Products, Services and Resources
- 6.9.2. Incident Management
- 6.9.3. CERTS and CSIRTs

6.10. Security Architecture Recovery
- 6.10.1. Resilience, Concepts, Business Requirements and Regulations
- 6.10.2. IT Resilience Solutions
- 6.10.3. Crisis Management and Governance

Module 7. Information Systems Security

7.1. A global Perspective on Security, Cryptography and Classical Cryptanalysis
- 7.1.1. Computer Security: Historical Perspective
- 7.1.2. But What Exactly is Meant by Security?
- 7.1.3. History of Cryptography
- 7.1.4. Substitution Ciphers
- 7.1.5. Case Study: The Enigma Machine

7.2. Symmetric Cryptography
- 7.2.1. Introduction and Basic Terminology
- 7.2.2. Symmetric Encryption
- 7.2.3. Modes of Operation
- 7.2.4. DES
- 7.2.5. The New AES Standard
- 7.2.6. Encryption in Flow
- 7.2.7. Cryptanalysis

7.3. Asymmetric Cryptography
- 7.3.1. Origins of Public Key Cryptography
- 7.3.2. Basic Concepts and Operation
- 7.3.3. The RSA Algorithm
- 7.3.4. Digital Certificates
- 7.3.5. Key Storage and Management

7.4. Network Attacks
- 7.4.1. Network Threats and Attacks
- 7.4.2. Enumeration
- 7.4.3. Traffic Interception: Sniffers
- 7.4.4. Denial of Service Attacks
- 7.4.5. ARP Poisoning Attacks

7.5. Security Architectures
- 7.5.1. Traditional Security Architectures
- 7.5.2. Secure Socket Layer: SSL
- 7.5.3. SSH Protocol
- 7.5.4. Virtual Private Networks (VPNs)
- 7.5.5. External Storage Unit Protection Mechanisms
- 7.5.6. Hardware Protection Mechanisms

7.6. System Protection Techniques and Secure Code Development
- 7.6.1. Operational Security
- 7.6.2. Resources and Controls
- 7.6.3. Monitoring
- 7.6.4. Intrusion Detection Systems
- 7.6.5. Host IDS
- 7.6.6. Network IDS
- 7.6.7. Signature-based IDS
- 7.6.8. Lure Systems
- 7.6.9. Basic Security Principles in Code Development
- 7.6.10. Failure Management
- 7.6.11. Public Enemy Number 1: Buffer Overflows
- 7.6.12. Cryptographic Botches

7.7. Botnets and Spam
- 7.7.1. Origin of the Problem
- 7.7.2. Spam Process
- 7.7.3. Sending Spam
- 7.7.4. Refinement of Mailing Lists
- 7.7.5. Protection Techniques
- 7.7.6. Anti-Spam Service offered by Third-Parties
- 7.7.7. Study Cases
- 7.7.8. Exotic Spam

7.8. Web Auditing and Attacks
- 7.8.1. Information Gathering
- 7.8.2. Attack Techniques
- 7.8.3. Data Science

7.9. Malware and Malicious Code
- 7.9.1. What is Malware?
- 7.9.2. Types of Malware
- 7.9.3. Virus
- 7.9.4. Criptovirus
- 7.9.5. Worms
- 7.9.6. Adware
- 7.9.7. Spyware
- 7.9.8. Hoaxes
- 7.9.9. Phishing
- 7.9.10. Trojans
- 7.9.11. The Economy of Malware
- 7.9.12. Possible Solutions

7.10. Forensic Analysis
- 7.10.1. Evidence Collection
- 7.10.2. Evidence Analysis
- 7.10.3. Anti-Forensic Techniques
- 7.10.4. Case Study

Module 8. Software Security

8.1. Problems of the Software Security
- 8.1.1. Introduction to the Problem of Software Safety
- 8.1.2. Vulnerabilities and their Classification
- 8.1.3. Secure Software Properties
- 8.1.4. References

8.2. Software Safety Design Principles
- 8.2.1. Introduction
- 8.2.2. Software Safety Design Principles
- 8.2.3. Types of S-SDLC
- 8.2.4. Software Safety in S-SDLC Phases
- 8.2.5. Methodologies and Standards
- 8.2.6. References

8.3. Software Lifecycle Safety in the Requirements and Design Phases
- 8.3.1. Introduction
- 8.3.2. Attack Modeling
- 8.3.3. Cases of Abuse
- 8.3.4. Safety Requirements Engineering
- 8.3.5. Risk Analysis Architectural
- 8.3.6. Design Patterns
- 8.3.7. References

8.4. Software Lifecycle Safety in the Coding, Testing and Operation Phases
- 8.4.1. Introduction
- 8.4.2. Risk-Based Safety Testing
- 8.4.3. Code Review
- 8.4.4. Penetration Test
- 8.4.5. Security Operations
- 8.4.6. External Review
- 8.4.7. References

8.5. Secure Coding Applications I
- 8.5.1. Introduction
- 8.5.2. Secure Coding Practices
- 8.5.3. Manipulation and Validation of Inputs
- 8.5.4. Memory Overflow
- 8.5.5. References

8.6. Secure Coding Applications II
- 8.6.1. Introduction
- 8.6.2. Integers Overflows, Truncation Errors and Problems with Type Conversions between Integers
- 8.6.3. Errors and Exceptions
- 8.6.4. Privacy and Confidentiality
- 8.6.5. Privileged Programs
- 8.6.6. References

8.7. Development and Cloud Security
- 8.7.1. Safety in Development; Methodology and Practice
- 8.7.2. PaaS, IaaS, CaaS and SaaS Models
- 8.7.3. Security in the Cloud and for Cloud Services

8.8. Security Automation and Orchestration (SOAR)
- 8.8.1. Complexity of Manual Processing; Need to Automate Tasks
- 8.8.2. Products and Services
- 8.8.3. SOAR Architecture

8.9. Telework Safety
- 8.9.1. Need and Scenarios
- 8.9.2. Products and Services
- 8.9.3. Telework Safety

Module 9. Quality and Auditing of Information Systems

9.1. Introduction to Information Security Management Systems
- 9.1.1. Fundamental Principles of ISMS
- 9.1.2. ISMS Golden Rules
- 9.1.3. Role of IT Audit in ISMSs

9.2. Safety Management Planning
- 9.2.1. Concepts Related to Safety Management
- 9.2.2. Classification of Information: Objectives, Concepts and Roles
- 9.2.3. Implementation of Security Policies: Security Policies, Standards and Procedures
- 9.2.4. Risk Management: Information Assets Risk Principles and Analysis

9.3. Main Mechanisms for the Protection of Information Assets (I)
- 9.3.1. Summary of the Main Cryptographic Tools for the Protection of the CID Triad of the CID Triad
- 9.3.2. Consideration of Privacy, Anonymity and Adequate Management of User Traceability Requirements

9.4. Main Mechanisms for the Protection of Information Assets (II)
- 9.4.1. Communications Security: Protocols, Devices and Security Architectures
- 9.4.2. Operating System Security

9.5. ISMS Internal Controls
- 9.5.1. ISMS Controls Taxonomy: Administrative, Logical and Physical Controls
- 9.5.2. Classification of Controls According to How Threats Are Addressed: Controls for Threat Prevention, Detection and Correction
- 9.5.3. Implementation of Internal Control Systems in ISMSs

9.6. Types of Audits
- 9.6.1. Difference between Audit and Internal Control
- 9.6.2. Internal vs. External Audit
- 9.6.3. Audit Classification according to the Objective and Type of Analysis

9.7. Screenwriter and Screenplay: Subject Matter and Object Protected by Intellectual Property
- 9.7.1. Introduction to Penetration Testing and Forensic Analysis
- 9.7.2. Definition and Relevance of Fingerprinting and Footprinting Concepts

9.8. Vulnerability Scanning and Network Traffic Monitoring
- 9.8.1. Tools for Vulnerability Analysis in Systems
- 9.8.2. Main Vulnerabilities in the Context of Web Applications
- 9.8.3. Analysis of Communications Protocols

9.9. The IT Audit Process
- 9.9.1. Life Cycle Concept in Systems Development
- 9.9.2. Activity and Process Monitoring: Collection and Treatment of Evidence
- 9.9.3. IT Audit Methodology
- 9.9.4. IT Audit Process
- 9.9.5. Identification of the Main Crimes and Misdemeanors in the Context of Information Technologies
- 9.9.6. Computer Crime Investigation: Introduction to Forensic Analysis and its relation to Computer Auditing

9.10. Business Continuity and Disaster Recovery Plans
- 9.10.1. Definition of Business Continuity Plan and the Business Interruption Concept
- 9.10.2. NIST Recommendation on Business Continuity Plans
- 9.10.3. Disaster Recovery Plan
- 9.10.4. Disaster Recovery Plan Process

Module 10. Web Server Administration

10.1. Introduction to Web Servers
- 10.1.1. What is a Web Server?
- 10.1.2. Architecture and Operation of a Web Server
- 10.1.3. Resources and Contents on a Web Server
- 10.1.4. Application Servers
- 10.1.5. Proxy Servers
- 10.1.6. Main Web Servers on the Market
- 10.1.7. Web Server Usage Statistics
- 10.1.8. Web Server Security
- 10.1.9. Load Balancing on Web Servers
- 10.1.10. References

10.2. HTTP Protocol Handling
- 10.2.1. Operation and Structure
- 10.2.2. Request Methods
- 10.2.3. Status Codes
- 10.2.4. Headers
- 10.2.5. Content Coding Code Pages
- 10.2.6. Performing HTTP Requests on the Internet Using a Proxy, Livehttpheadersor similar Method, Analyzing the Protocol Used

10.3. Description of Distributed Multi-Server Architectures
- 10.3.1. 3-Layer Model
- 10.3.2. Fault Tolerance
- 10.3.3. Load Sharing
- 10.3.4. Session State Stores
- 10.3.5. Cache Stores

10.4. Internet Information Services (IIS)
- 10.4.1. What is IIS?
- 10.4.2. History and Evolution of IIS
- 10.4.3. Main Advantages and Features of IIS and Later
- 10.4.4. IIS Architecture and Later

10.5. IIS Installation, Administration and Configuration
- 10.5.1. Preamble
- 10.5.2. Internet Information Services (IIS)
- 10.5.3. IIS Administration Tools
- 10.5.4. Web Site Creation, Configuration and Administration
- 10.5.5. Installation and Management of IIS Extensions

10.6. Advanced Security in IIS
- 10.6.1. Preamble
- 10.6.2. Authentication, Authorization, and Access Control in IIS
- 10.6.3. Configuring a Secure Website on IIS with SSL
- 10.6.4. Security Policies Implemented in IIS 10.x

10.7. Introduction to Apache
- 10.7.1. What is Apache?
- 10.7.2. Main Advantages of Apache
- 10.7.3. Main Features of Apache
- 10.7.4. Architecture

10.8. Apache Installation and Configuration
- 10.8.1. Initial Installation of Apache
- 10.8.2. Apache Configuration

10.9. Installation and Configuration of the Different Apache Modules
- 10.9.1. Apache Module Installation
- 10.9.2. Types of Modules
- 10.9.3. Secure Apache Configuration

10.10. Advanced Security
- 10.10.1. Authentication, Authorization and Access Control
- 10.10.2. Authentication Methods
- 10.10.3. Secure Apache Configuration with SSL

Module 11. Online Application Security

11.1. Vulnerabilities and Security Issues in Online Applications
- 11.1.1. Introduction to Online Application Security
- 11.1.2. Security Vulnerabilities in the Design of Web Applications
- 11.1.3. Security Vulnerabilities in the Implementation of Web Applications
- 11.1.4. Security Vulnerabilities in the Deployment of Web Applications
- 11.1.5. Official Lists of Security Vulnerabilities

11.2. Policies and Standards for Online Application Security
- 11.2.1. Pillars for the Security of Online Applications
- 11.2.2. Security Policy
- 11.2.3. Information Security Management System
- 11.2.4. Secure Software Development Life Cycle
- 11.2.5. Standards for Application Security

11.3. Security in the Design of Web Applications
- 11.3.1. Introduction to Web Application Security
- 11.3.2. Security in the Design of Web Applications

11.4. Testing the Online Safety and Security of Web Applications
- 11.4.1. Web Application Security Testing and Analysis
- 11.4.2. Web Application Deployment and Production Security

11.5. Web Services Security
- 11.5.1. Introduction to Web Services Security
- 11.5.2. Web Services Security Functions and Technologies

11.6. Testing the Online Safety and Security of Web Services
- 11.6.1. Evaluation of Web Services Security
- 11.6.2. Online Protection. Firewalls and Gateways XML

11.7. Ethical Hacking, malware and Forensic
- 11.7.1. Ethical Hacking
- 11.7.2. Malware Analysis
- 11.7.3. Forensic Analysis

11.8. Incident Resolution on Web Services
- 11.8.1. Monitoring
- 11.8.2. Performance Measurement Tools
- 11.8.3. Containment Measures
- 11.8.4. Root Cause Analysis
- 11.8.5. Proactive Problem Management

11.9. Best Practices to ensure Application Security
- 11.9.1. Handbook of Best Practices in the Development of Online Applications
- 11.9.2. Handbook of Good Practices in the Implementation of Online Applications

11.10. Common Errors that Undermine Application Security
- 11.10.1. Common Errors in Development
- 11.10.2. Common Errors in Hosting
- 11.10.3. Common Production Errors

Module 12. Software Engineering

12.1. Introduction to Software Engineering and Modeling
- 12.1.1. The Nature of Software
- 12.1.2. The Unique Nature of Webapps
- 12.1.3. Software Engineering
- 12.1.4. The Software Process
- 12.1.5. Software Engineering Practice
- 12.1.6. Software Myths
- 12.1.7. How Does It All Begin?
- 12.1.8. Object-Oriented Concepts
- 12.1.9. Introduction to UML

12.2. The Software Process
- 12.2.1. A General Process Model
- 12.2.2. Prescriptive Process Models
- 12.2.3. Specialized Process Models
- 12.2.4. The Unified Process
- 12.2.5. Personal and Team Process Models
- 12.2.6. What is Agility?
- 12.2.7. What is an Agile Process?
- 12.2.8. Scrum
- 12.2.9. Agile Process Toolkit

12.3. Principles Guiding Software Engineering Practice
- 12.3.1. Principles Guiding the Process
- 12.3.2. Principles Guiding the Practice
- 12.3.3. Principles of Communication
- 12.3.4. Planning Principles
- 12.3.5. Modeling Principles
- 12.3.6. Construction Principles
- 12.3.7. Deployment Principles

12.4. Understanding the Requirements
- 12.4.1. Requirements Engineering
- 12.4.2. Establish the Basis
- 12.4.3. Inquiry of Requirements
- 12.4.4. Development of Cases Studies
- 12.4.5. Elaboration of the Requirements Model
- 12.4.6. Negotiation of Requirements
- 12.4.7. Validation of Requirements

12.5. Requirements Modeling: Scenarios, Information and Analysis Classes
- 12.5.1. Analysis of Requirements
- 12.5.2. Scenario-Based Modeling
- 12.5.3. UML Models that provide the Case Study
- 12.5.4. Data Modeling Concepts
- 12.5.5. Class-Based Modeling
- 12.5.6. Class Diagrams

12.6. Requirements Modeling: Flow, Behavior and Patterns
- 12.6.1. Requirements that Shape Strategies
- 12.6.2. Flow-Oriented Modeling
- 12.6.3. Status Diagrams
- 12.6.4. Creation of a Behavioral Model
- 12.6.5. Sequence Diagrams
- 12.6.6. Communication Diagrams
- 12.6.7. Patterns for Requirements Modeling

12.7. Design Concepts
- 12.7.1. Design in the Software Engineering Context
- 12.7.2. The Design Process

12.7.3. Design Concepts
12.7.4. Object-Oriented Design Concepts
12.7.5. Model of the Design
12.8. Designing the Architecture:
12.8.1. Software Architecture
12.8.2. Architectural Genres
12.8.3. Architectural Styles
12.8.4. Architectural Design
12.8.5. Evolution of Alternative Designs for Architecture
12.8.6. Mapping the Architecture Using the Data Flow
12.9. Component-Level and Pattern-Based Design
12.9.1. What is a Component?
12.9.2. Class-Based Component Design
12.9.3. Realization of the Design at the Component Level
12.9.4. Design of Traditional Components
12.9.5. Component-Based Development
12.9.6. Design Patterns
12.9.7. Pattern-Based Software Design
12.9.8. Architectural Patterns
12.9.9. Design Patterns at the Component Level
12.9.10. User Interface Design Patterns
12.10. Software Quality and Project Management
12.10.1. Quality
12.10.2. Software Quality
12.10.3. The Software Quality Dilemma
12.10.4. Achieving Software Quality
12.10.5. Software Quality Assurance
12.10.6. The Administrative Spectrum
12.10.7. The Staff
12.10.8. The product
12.10.9. The Process
12.10.10. The Project
12.10.11. Principles and Practices

Module 13. Advanced Software Engineering

13.1. Introduction to Agile Methodologies
13.1.1. Process Models and Methodologies
13.1.2. Agility and Agile Processes
13.1.3. Agile Manifesto
13.1.4. Some Agile Methodologies
13.1.5. Agile vs. Traditional
13.2. Scrum
13.2.1. Origins and Philosophy of Scrum
13.2.2. Scrum Values
13.2.3. Scrum Process Flow
13.2.4. Scrum Roles
13.2.5. Scrum Artifacts
13.2.6. Scrum Events
13.2.7. User Stories
13.2.8. Scrum Extensions
13.2.9. Agile Estimates
13.2.10. Scrum Scaling
13.3. Extreme Programming
13.3.1. Justification and Overview of XP
13.3.2. The XP Life Cycle
13.3.3. The Five Core Values
13.3.4. The Twelve Basic Practices in XP
13.3.5. Roles of Participants
13.3.6. XP Industrial
13.3.7. Critical Assessment of XP
13.4. Software Development Based on Reusability
13.4.1. Software Reuse
13.4.2. Code Reuse Levels
13.4.3. Specific Reuse Techniques
13.4.4. Component-Based Development
13.4.5. Benefits and Problems of Reuse
13.4.6. Reuse Planning

13.5. System Architecture and Software Design Patterns
- 13.5.1. Architectural Design
- 13.5.2. General Architectural Patterns
- 13.5.3. Fault Tolerant Architectures
- 13.5.4. Distributed Systems Architectures
- 13.5.5. Design Patterns
- 13.5.6. Gamma Patterns
- 13.5.7. Interaction Design Patterns

13.6. Cloud Application Architecture
- 13.6.1. Cloud Computing Fundamentals
- 13.6.2. Cloud Application Quality
- 13.6.3. Architectural Styles
- 13.6.4. Design Patterns

13.7. Software Testing: TDD, ATDD and BDD
- 13.7.1. Software Verification and Validation
- 13.7.2. Software Testing
- 13.7.3. Test Driven Development (TDD)
- 13.7.4. Acceptance Test Driven Development (ATDD)
- 13.7.5. Test Driven Development (BDD)
- 13.7.6. BDD and Cucumber

13.8. Software Process Improvement
- 13.8.1. Software Process Improvement
- 13.8.2. The Process Improvement Approach
- 13.8.3. Maturity Models
- 13.8.4. The CMMI Model
- 13.8.5. CMMI V13.0
- 13.8.6. CMMI and Agile

13.9. The Quality of the Software Product: SQuaRE
- 13.9.1. Software Quality
- 13.9.2. Software Product Quality Models
- 13.9.3. ISO/IEC 13.000 Family
- 13.9.4. ISO/IEC 13.010: Quality Model and Quality Characteristics
- 13.9.5. ISO/IEC 13.0113. the Quality of the Data
- 13.9.6. ISO/IEC 13.013: Software Quality Measurement
- 13.9.7. ISO/IEC 13.013., 13.013. and 13.013.: Software Quality and Data Quality Metrics
- 13.9.8. ISO/IEC 13.040 Software Assessment
- 13.9.9. Accreditation Process

13.10. Introduction to DevOps
- 13.10.1. DevOps Concept
- 13.10.2. Core Practices

Module 14. Requirements Engineering

14.1. Introduction to Requirements Engineering
- 14.1.1. The Importance of Requirements
- 14.1.2. Concept of Requirement
- 14.1.3. Dimensions of Requirements
- 14.1.4. Levels and Types of Requirements
- 14.1.5. Requirements Characteristics
- 14.1.6. Requirements Engineering
- 14.1.7. The Requirements Engineering Process
- 14.1.8. Frameworks for Requirements Engineering
- 14.1.9. Best Practices in Requirements Engineering
- 14.1.10. The Business Analyst

14.2. Sources of Requirements
- 14.2.1. The Requirements Network
- 14.2.2. The Stakeholders
- 14.2.3. Business Requirements
- 14.2.4. Vision and Scope Document

14.3. Requirements Elicitation Techniques
- 14.3.1. Elicitation of Requirements
- 14.3.2. Problems of Requirements Elicitation
- 14.3.3. Contexts of Discovery
- 14.3.4. Interviews
- 14.3.5. Observation and "Learning
- 14.3.6. Ethnography
- 14.3.7. Workshops
- 14.3.8. Focus groups
- 14.3.9. Questionnaires
- 14.3.10. Brainstorming and Creative Techniques
- 14.3.11. Group Media
- 14.3.12. Analysis of System Interfaces
- 14.3.13. Document Analysis and "Archeology"
- 14.3.14. Case Studies and Scenarios
- 14.3.15. Prototypes
- 14.3.16. Reverse Engineering
- 14.3.17. Reuse of Requirements
- 14.3.18. Good Elicitation Practices

14.4. User Requirements
- 14.4.1. Person
- 14.4.2. Case Studies and User Stories
- 14.4.3. Scenarios
- 14.4.4. Types of Scenarios
- 14.4.5. How to Discover Scenarios?

14.5. Prototyping Techniques
- 14.5.1. Prototyping
- 14.5.2. Prototypes According to their Scope
- 14.5.3. Prototypes According to their Seasonality
- 14.5.4. The Fidelity of a Prototype
- 14.5.5. User Interface Prototypes
- 14.5.6. Evaluation of Prototypes

14.6. Requirements Analysis
- 14.6.1. Requirements Analysis
- 14.6.2. Requirements Analysis Best Practices
- 14.6.3. The Data Dictionary
- 14.6.4. Prioritization of Requirements

14.7. Documentation of Requirements
- 14.7.1. The Requirements Specification Document
- 14.7.2. Structure and Contents of an SRS
- 14.7.3. Natural Language Documentation
- 14.7.4. EARS: Easy Approach to Requirements Syntax
- 14.7.5. Non-Functional Requirements
- 14.7.6. Attributes and Templates in Table Form
- 14.7.7. Good Specifications Practices

14.8. Validation and Negotiation of Requirements
- 14.8.1. Validation of Requirements
- 14.8.2. Requirements Validation Techniques
- 14.8.3. Negotiation of Requirements

14.9. Modeling and Requirements Management
- 14.9.1. Requirements Modeling
- 14.9.2. The User's Perspective
- 14.9.3. The Data Perspective
- 14.9.4. The Functional or Flow-Oriented Perspective
- 14.9.5. The Behavioral Perspective
- 14.9.6. Volatility of Requirements
- 14.9.7. Requirements Management Process
- 14.9.8. Tools for Requirements Management
- 14.9.9. Best Practices in Requirements Management

14.10. Critical Systems and Formal Specification
- 14.10.1. Critical Systems
- 14.10.2. Risk-Driven Specification
- 14.10.3. Formal Specification

Module 15. Software Engineering Processes

15.1. Software Engineering Framework
- 15.1.1. Software Features
- 15.1.2. The Main Processes in Software Engineering
- 15.1.3. Software Development Process Models
- 15.1.4. Standard Reference Framework for the Software Development Process: The ISO/IEC 12207 Standard

15.2. Unified Software Development Process
- 15.2.1. The Unified Process
- 15.2.2. Dimensions of the Unified Process
- 15.2.3. Case Studies Driven Development Process
- 15.2.4. Fundamental Workflows of Unified Processes

15.3. Planning in the Context of Agile Software Development
- 15.3.1. Characteristics of Agile Software Development
- 15.3.2. Different Planning Time Horizons in Agile Development
- 15.3.3. Scrum Agile Development Framework and Planning Time Horizons
- 15.3.4. User Stories as a Planning and Estimating Unit
- 15.3.5. Common Techniques for Deriving an Estimate
- 15.3.6. Scales for Interpreting Estimates
- 15.3.7. Planning Poker
- 15.3.8. Common Scheduling Types: Delivery Scheduling and Iteration Scheduling

15.4. Distributed Software Design Styles and Service-Oriented Software Architectures
- 15.4.1. Communication Models in Distributed Software Systems
- 15.4.2. Middleware
- 15.4.3. Architecture Patterns for Distributed Systems
- 15.4.4. General Software Service Design Process
- 15.4.5. Design Aspects of Software Services
- 15.4.6. Composition of Services
- 15.4.7. Web Services Architecture
- 15.4.8. Infrastructure and SOA Components

15.5. Introduction to Model Driven Software Development
- 15.5.1. The Model Concept
- 15.5.2. Model-Driven Software Development
- 15.5.3. MDA Model-Driven Development Framework
- 15.5.4. Elements of a Transformation Model

15.6. Graphical User Interface Design
- 15.6.1. Principles of User Interface Design
- 15.6.2. Architectural Design Patterns for Interactive Systems: Model View Controller (MVC)
- 15.6.3. UX User Experience
- 15.6.4. User-Centered Design
- 15.6.5. Graphical User Interface Analysis and Design Process
- 15.6.6. Usability of User Interfaces
- 15.6.7. Accessibility in User Interfaces

15.7. Web Application Design
- 15.7.1. Characteristics of Web Applications
- 15.7.2. Web Application User Interface
- 15.7.3. Navigation Design
- 15.7.4 Base Interaction Protocol for Web Applications
- 15.7.5. Architecture Styles for Web Applications

15.8. Software Testing Strategies and Techniques and Software Quality Factors
- 15.8.1. Testing Strategies
- 15.8.2. Test Case Designs
- 15.8.3. Value for Money
- 15.8.4 Quality Models
- 15.8.5. ISO/IEC 25000 Family of Standards (SQuaRE)
- 15.8.6. Product Quality Model (ISO 2501n)
- 15.8.7. Data Quality Models (ISO 2501n)
- 15.8.8. Software Quality Management

15.9. Introduction to Software Engineering Metrics
- 15.9.1. Basic Concepts: Measurements, Metrics and Indicators
- 15.9.2. Types of Metrics in Software Engineering
- 15.9.3. The Measurement Process
- 15.9.4. ISO 250215. External and Quality Metrics in Use
- 15.9.5. Object-Oriented Metrics

15.10. Software Maintenance and Reengineering
- 15.10.1. Maintenance Process
- 15.10.2. Standard Maintenance Process Framework. ISO/EIEC 115.64
- 15.10.3. Software Reengineering Process Model
- 15.10.4. Inverse Engineering

Module 16. Integration Systems

16.1. Introduction to Information Systems in the Company
- 16.1.1. The Role of Information Systems
- 16.1.2. What is an Information System?
- 16.1.3. Dimensions of Information Systems
- 16.1.4. Business Processes and Information Systems
- 16.1.5. The IS/IT Department

16.2. Opportunities and Needs of Information Systems in the Company
- 16.2.1. Organizations and Information Systems
- 16.2.2. Features of Organisations
- 16.2.3. Impact of Information Systems in the Company
- 16.2.4. Information Systems to Achieve a Competitive Advantage
- 16.2.5. Use of Systems in the Administration and Management of the Company

16.3. Basic Concepts of Information Systems and Technologies
- 16.3.1. Data, Information and Knowledge
- 16.3.2. Technology and Information Systems
- 16.3.3. Technology Components
- 16.3.4. Classification and Types of Information Systems
- 16.3.5. Service and Business Process Based Architectures
- 16.3.6. Forms of Systems Integration

16.4. Systems for the Integrated Management of Company Resources
- 16.4.1. Business Needs
- 16.4.2. An integrated Information System for the Company
- 16.4.3. Acquisition vs. Development
- 16.4.4. ERP Implementation
- 16.4.5. Implications for Management
- 16.4.6. Leading ERP Vendors

16.5. Supply Chain and Customer Relationship Management Information Systems
- 16.5.1. Definition of Supply Chain
- 16.5.2. Effective Supply Chain Management
- 16.5.3. The Role of Information Systems
- 16.5.4. Supply Chain Management Solutions
- 16.5.5. Customer Relationship Management
- 16.5.6. The Role of Information Systems
- 16.5.7. Implementation of a CRM System
- 16.5.8. Critical Success Factors in CRM Implementation
- 16.5.9. CRM, e-CRM and Other Trends

16.6. ICT Investment Decision-Making and Information Systems Planning
- 16.6.1. Criteria for ICT Investment Decisions
- 16.6.2. Linking the Project to the Management and Business Plan
- 16.6.3. Management Implications
- 16.6.4. Redesign of Business Processes
- 16.6.5. Management's Decision on Implementation Methodologies
- 16.6.6. Need for Information Systems Planning
- 16.6.7. Objectives, Participants and Moments
- 16.6.8. Structure and Development of the Systems Planning
- 16.6.9. Follow-up and Updating

16.7. Security Considerations in the Use of ICTs
- 16.7.1. Risk Analysis
- 16.7.2. Security in Information Systems
- 16.7.3. Practical Advice

16.8. Feasibility of ICT Project Implementation and Financial Aspects in Information Systems Projects
- 16.8.1. Description and Objectives
- 16.8.2. EVS Participants
- 16.8.3. Techniques and Procedures
- 16.8.4. Cost structure
- 16.8.5. Financial Projection
- 16.8.6. Budgets

16.9. Business Intelligence
- 16.9.1. What is Business Intelligence?
- 16.9.2. BI Implementation Strategy
- 16.9.3. Present and Future in BI

16.10. ISO/IEC 12207
- 16.10.1. What is "ISO/IEC 12207"?
- 16.10.2. Analysis of Information Systems
- 16.10.3. Information System Design
- 16.10.4. Implementation and Acceptance of the Information System

Module 17. Software Reuse

17.1. General Overview of the Software Reuse
- 17.1.1. What is Software Reuse?
- 17.1.2. Advantages and Disadvantages of Software Reuse
- 17.1.3. Main Techniques of Software Reuse

17.2. Introduction to Design Patterns
- 17.2.1. What is a Design Patterns?
- 17.2.2. Catalog of the Main Design Patterns
- 17.2.3. How to Use Patterns to Solve Design Problems
- 17.2.4. How to Select the Best Design Pattern

17.3. Creation Patterns
- 17.3.1. Creation Patterns
- 17.3.2. Abstract Factory Pattern
- 17.3.3. Example of Abstract Factory Pattern implementation
- 17.3.4. Builder Pattern
- 17.3.5. Builder Implementation Example
- 17.3.6. Abstract Factory Pattern vs. Builder

17.4. Creation Patterns (II)
- 17.4.1. Factory Method Pattern
- 17.4.2. Factory Method vs. Abstract Factory
- 17.4.3. Singleton Pattern

17.5. Structural Patterns
- 17.5.1. Structural Patterns
- 17.5.2. Adapter Pattern
- 17.5.3. Bridge Pattern

17.6. Structural Patterns (II)
- 17.6.1. Composite Pattern
- 17.6.2. Decorador Pattern

17.7. Structural Patterns (III)
- 17.7.1. Facade Pattern
- 17.7.2. Proxy Pattern

17.8. Behavioral Patterns
- 17.8.1. Concept of Behavioral Patterns
- 17.8.2. Behavior Pattern: Chain of Responsibility
- 17.8.3. Behavior Pattern Order

17.9. Behavioral Patterns (II)
- 17.9.1. Interpreter Pattern
- 17.9.2. Iterator Pattern
- 17.9.3. Observer Pattern
- 17.9.4. Strategy Pattern

17.10. Frameworks
- 17.10.1. Concept of Framework
- 17.10.2. Development using Frameworks
- 17.10.3. Model View Controller Pattern
- 17.10.4. Framework for Graphical User Interface Design
- 17.10.5. Frameworks for Web Application Development
- 17.10.6. Frameworks for Managing Object Persistence in Databases

Module 18. Information Technology Services

18.1. Digital Transformation (I)
- 18.1.1. Business Innovation
- 18.1.2. Production Management
- 18.1.3. Financial Management

18.2. Digital Transformation (II)
- 18.2.1. Marketing
- 18.2.2. HR Management
- 18.2.3. The Integrated Information System

18.3. Case Study
- 18.3.1. Company Presentation
- 18.3.2. Methodologies to Analyze the Acquisition of IT
- 18.3.3. Determining the Costs, Benefits and Risks
- 18.3.4. Economic Evaluation of Investment

18.4. ICT Governance and Management
- 18.4.1. Definition of IT and Information Systems Governance
- 18.4.2. Difference Between IT Systems Governance and Management
- 18.4.3. Framework for IT Systems Governance and Management
- 18.4.4. Regulations and IT Systems Governance and Management

18.5. ICT Corporate Governance
- 18.5.1. What is Good Corporate Governance?
- 18.5.2. ICT Governance Background
- 18.5.3. The ISO/IEC 318.00:2008 Standard
- 18.5.4. Implementation of Good ICT Governance
- 18.5.5. ICT Governance and Best Practices
- 18.5.6. Corporate Governance. Summary and Trends

18.6. Control Objectives for Information and Related Technologies (COBIT)
- 18.6.1. Application Framework
- 18.6.2. Domain: Planning and Organization
- 18.6.3. Domain: Acquisition and Implementation
- 18.6.4. Domain: Delivery and Support
- 18.6.5. Domain: Supervision and Evaluation
- 18.6.6. Application of the COBIT Guide

18.7. The Information Technology Infrastructure Library (ITIL)
- 18.7.1. Introduction to ITIL
- 18.7.2. Service Strategies
- 18.7.3. Service Design
- 18.7.4. Transition Between Services
- 18.7.5. Service Operation
- 18.7.6. Improving the Service

18.8. The Service Management System
- 18.8.1. Basic Principles of UNE-ISO/IEC 20000-1
- 18.8.2. The Structure of the ISO/IEC 20000 Regulations
- 18.8.3. Service Management System (SMS) Requirements
- 18.8.4. Design and Transition of New or Modified Services
- 18.8.5. Service Provision Processes
- 18.8.6. Groups of Processes

18.9. The Software Asset Management System
- 18.9.1. Justification of Needs
- 18.9.2. Background
- 18.9.3. Presentation of the 19770 Regulation
- 18.9.4. Management Implementation

18.10. Business Continuity Management
- 18.10.1. Business Continuity Plan
- 18.10.2. Implementation of a BCP

A comprehensive program that will be fundamental for your professional development"

05

Methodology

This academic program offers students a different way of learning. Our methodology uses a cyclical learning approach: **Relearning.**

This teaching system is used, for example, in the most prestigious medical schools in the world, and major publications such as the **New England Journal of Medicine** have considered it to be one of the most effective.

Discover Relearning, a system that abandons conventional linear learning, to take you through cyclical teaching systems: a way of learning that has proven to be extremely effective, especially in subjects that require memorization"

Case Study to contextualize all content

Our program offers a revolutionary approach to developing skills and knowledge. Our goal is to strengthen skills in a changing, competitive, and highly demanding environment.

At TECH, you will experience a learning methodology that is shaking the foundations of traditional universities around the world"

You will have access to a learning system based on repetition, with natural and progressive teaching throughout the entire syllabus.

The student will learn to solve complex situations in real business environments through collaborative activities and real cases.

A learning method that is different and innovative

This TECH program is an intensive educational program, created from scratch, which presents the most demanding challenges and decisions in this field, both nationally and internationally. This methodology promotes personal and professional growth, representing a significant step towards success. The case method, a technique that lays the foundation for this content, ensures that the most current economic, social and professional reality is taken into account.

> *Our program prepares you to face new challenges in uncertain environments and achieve success in your career"*

The case method has been the most widely used learning system among the world's leading Information Technology schools for as long as they have existed. The case method was developed in 1912 so that law students would not only learn the law based on theoretical content. It consisted of presenting students with real-life, complex situations for them to make informed decisions and value judgments on how to resolve them. In 1924, Harvard adopted it as a standard teaching method.

What should a professional do in a given situation? This is the question that you are presented with in the case method, an action-oriented learning method. Throughout the course, students will be presented with multiple real cases. They will have to combine all their knowledge and research, and argue and defend their ideas and decisions.

Relearning Methodology

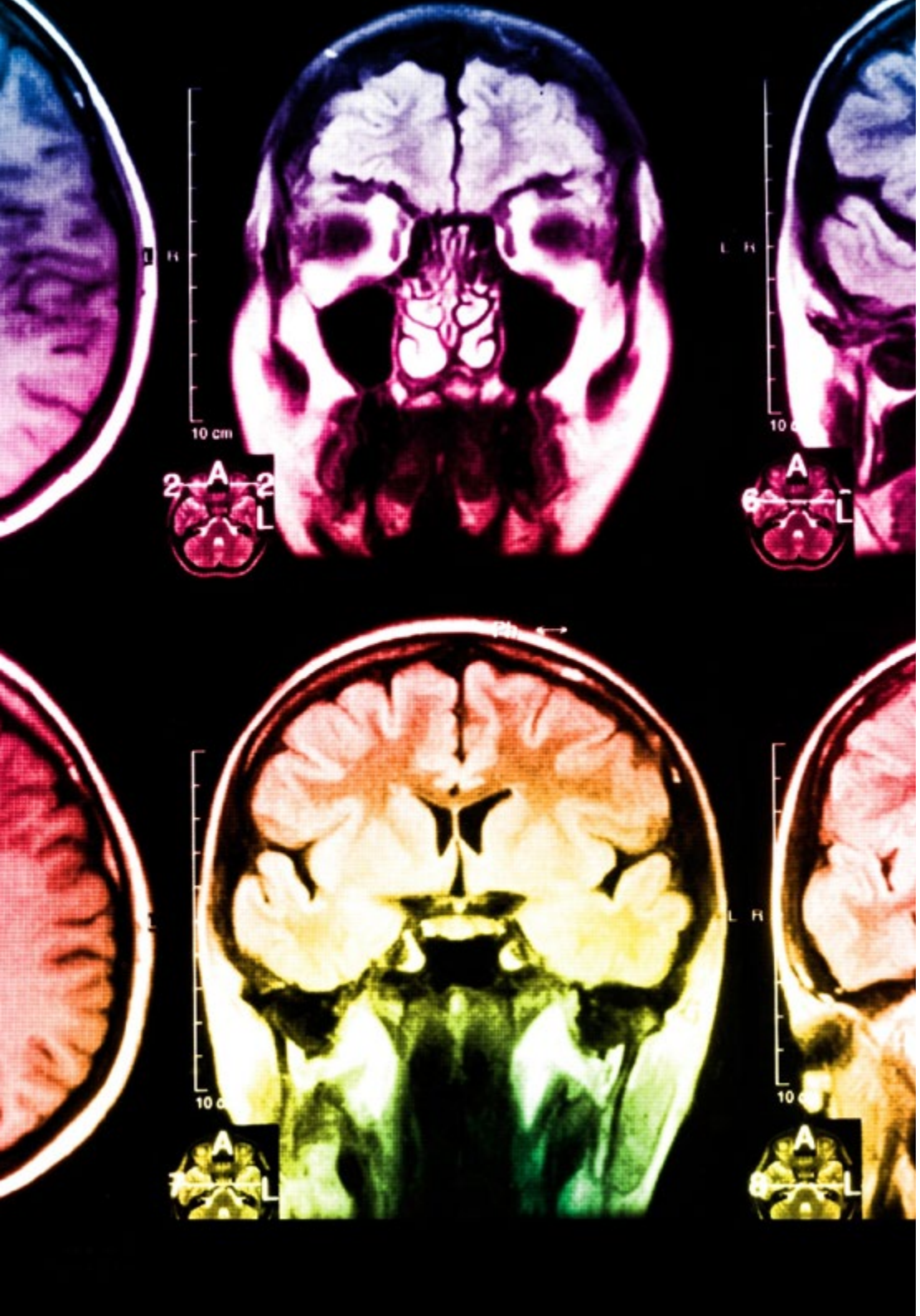
TECH effectively combines the Case Study methodology with a 100% online learning system based on repetition, which combines different teaching elements in each lesson.

We enhance the Case Study with the best 100% online teaching method: Relearning.

In 2019, we obtained the best learning results of all online universities in the world.

At TECH you will learn using a cutting-edge methodology designed to train the executives of the future. This method, at the forefront of international teaching, is called Relearning.

Our university is the only one in the world authorized to employ this successful method. In 2019, we managed to improve our students' overall satisfaction levels (teaching quality, quality of materials, course structure, objectives...) based on the best online university indicators.



In our program, learning is not a linear process, but rather a spiral (learn, unlearn, forget, and re-learn). Therefore, we combine each of these elements concentrically. This methodology has trained more than 650,000 university graduates with unprecedented success in fields as diverse as biochemistry, genetics, surgery, international law, management skills, sports science, philosophy, law, engineering, journalism, history, and financial markets and instruments. All this in a highly demanding environment, where the students have a strong socio-economic profile and an average age of 43.5 years.

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

From the latest scientific evidence in the field of neuroscience, not only do we know how to organize information, ideas, images and memories, but we know that the place and context where we have learned something is fundamental for us to be able to remember it and store it in the hippocampus, to retain it in our long-term memory.

In this way, and in what is called neurocognitive context-dependent e-learning, the different elements in our program are connected to the context where the individual carries out their professional activity.

This program offers the best educational material, prepared with professionals in mind:

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.

These contents are then applied to the audiovisual format, to create the TECH online working method. All this, with the latest techniques that offer high quality pieces in each and every one of the materials that are made available to the student.

Classes

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

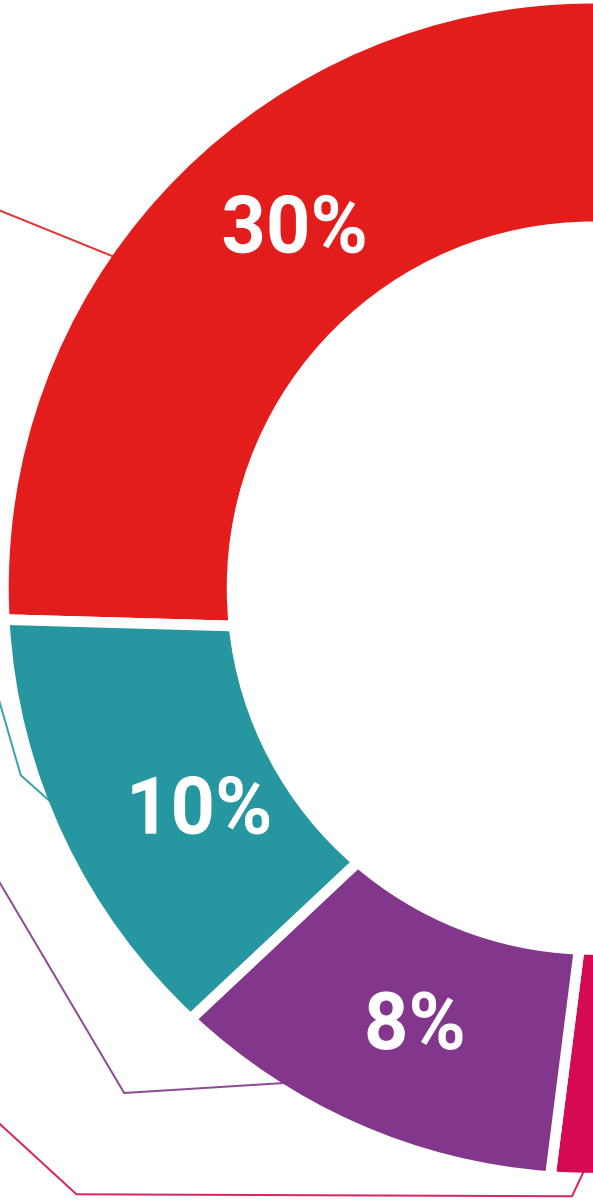
Learning from an Expert strengthens knowledge and memory, and generates confidence in future difficult decisions.

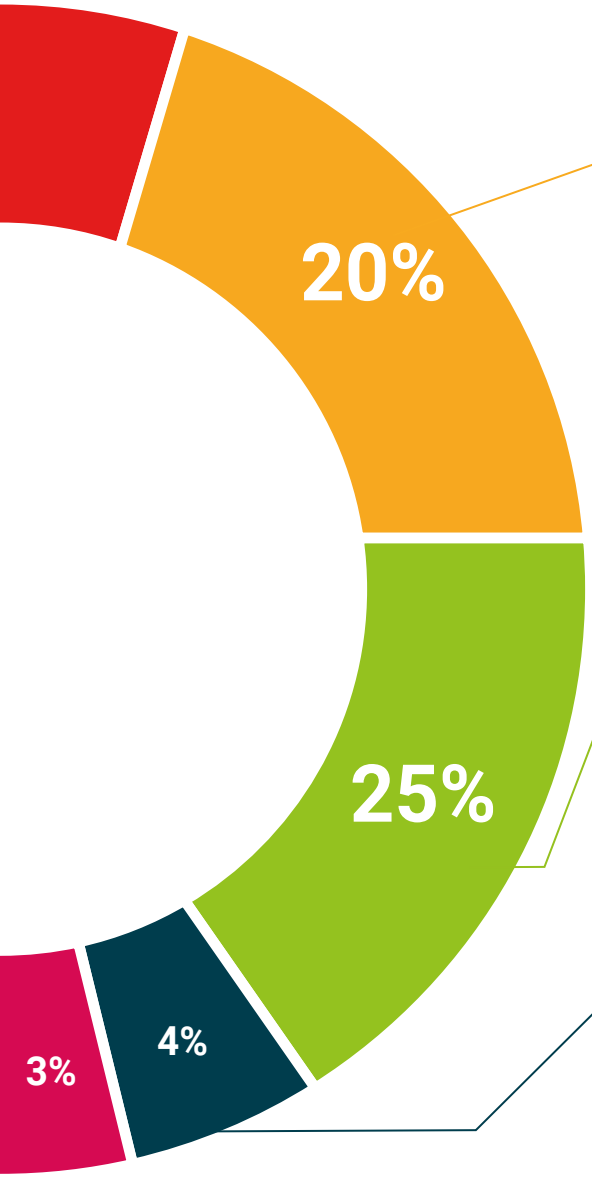
Practising Skills and Abilities

They will carry out activities to develop specific skills and abilities in each subject area. Exercises and activities to acquire and develop the skills and abilities that a specialist needs to develop in the context of the globalization that we are experiencing.

Additional Reading

Recent articles, consensus documents and international guidelines, among others. In TECH's virtual library, students will have access to everything they need to complete their course.





Case Studies

Students will complete a selection of the best case studies chosen specifically for this program. Cases that are presented, analyzed, and supervised by the best specialists in the world.

Interactive Summaries

The TECH team presents 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".

Testing & Retesting

We periodically evaluate and re-evaluate students' knowledge throughout the program, through assessment and self-assessment activities and exercises, so that they can see how they are achieving their goals.

06

Certificate

The Advanced Master's Degree in Software Engineering guarantees you, in addition to the most rigorous and up-to-date training, access to a Advanced Master's Degree issued by TECH Global University.

Successfully complete this program and receive your university degree without travel or laborious paperwork"

This program will allow you to obtain your **Advanced Master's Degree diploma in Software Engineering** endorsed by **TECH Global University**, the 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** title 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.

Title: **Advanced Master's Degree in Software Engineering**

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

This is a program of 3,000 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
Rector

Unique TECH Code: AFWORD23S techtitute.com/certificates


Advanced Master's Degree in Software Engineering

General Structure of the Syllabus

Year	Subject	ECTS	Type
1º	Methodologies, Development and Quality in	7	CO
1º	Software Engineering	7	CO
1º	Software Project Management	7	CO
1º	Software Development Platforms	7	CO
1º	Web-Client Computing	7	CO
1º	Web Server Computing	7	CO
1º	Safety Management	7	CO
1º	Information Systems Security	7	CO
1º	Software Security	7	CO
1º	Web Server Administration	7	CO

Year	Subject	ECTS	Type
2º	Online Application Security	7	CO
2º	Software Engineering	7	CO
2º	Advanced Software Engineering	7	CO
2º	Requirements Engineering	7	CO
2º	Software Engineering Processes	7	CO
2º	Integration Systems	4	CO
2º	Software Reuse	4	CO
2º	Information Technology Services	4	CO

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

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

Advanced Master's Degree
Software Engineering

tech global university