**Course Description:** This independent study course is designed to explore the integration of Artificial Intelligence (AI) techniques into Software Security. The primary objective is to develop and assess a Software Code Analyzer that identifies vulnerabilities in software and recommends fixes using AI-based approaches.

# Assignment 1: Project Plan (Weight: 10%)

## Criteria:

## Clarity and completeness of the project objectives, scope, and potential stakeholders.

## Project Schedule and Timeline as they align to the Software Development Lifecycle (SDLC) Phases

## Inclusion of technical reviews (Requirements Review, Design Review, Development Review, Final Review) with entrance and exit criteria.

## Identification and analysis of risks.

## Assumptions and constraints.

## Expectations:

## A well-structured project plan outlining the entire project lifecycle.

## Clear identification of SDLC phases (e.g., planning, requirements, design, development, testing, deployment).

## Detailed technical review processes with criteria for each phase.

## Risk identification, assessment, and mitigation strategies.

## Explicitly stated assumptions and constraints.

## Methods for Assessment:

* Evaluation of the Project Plan Document
* Project Kick-Off Meeting Approval

# Assignment 2: Requirements Documentation (Weight: 15%)

## Criteria:

## Clarity and completeness of stakeholder requirements.

## Well-defined Software Functional Requirements.

## Explicit Software Non-Functional Requirements.

## Comprehensive Testing / Verification Planning.

## Development of the Requirements Traceability Matrix (RTM) and Requirement Verification Matrix (RVM).

## Expectations:

* Elicitation and documentation of stakeholder requirements.
* Clear and unambiguous Software Functional Requirements.
* Identification and specification of Non-Functional Requirements (e.g., performance, security, usability).
* Detailed testing and verification plans, including test cases.
* Creation of an RTM and RVM to trace requirements to test cases and verification methods.

## Methods for Assessment:

* Evaluation of the Stakeholder Requirements Document
* Evaluation of the Software Requirements Document
* Stakeholder and Software Requirements Review Approval

# Assignment 3: Software Design Documentation (Weight: 15%)

## Criteria:

* Comprehensive Software Design Document.
* Clear Architecture Diagrams, including Structure and Behavior diagrams.
* Adequate Requirements Allocation Matrix.
* Effective use of various diagram types (e.g., Component, Package, Use Case, Sequence, Class).
* Alignment of design with requirements.

## Expectations:

* A detailed Software Design Document outlining the system's architecture and components.
* Well-structured architecture diagrams, including Structure (e.g., Component Diagrams) and Behavior (e.g., Sequence Diagrams) diagrams.
* Explicit Requirements Allocation Matrix linking design elements to requirements.
* Effective use of various diagram types to convey system structure and behavior.
* Demonstrated alignment of design choices with specified requirements.

## Methods for Assessment:

* Evaluation of the Software Design Documentation.
* Software Design Review Approval

# Assignment 4: AI-enabled Software Code Analyzer Prototype Development (Weight: 25%)

Note: You should start this at the beginning of the Course. Do not wait until Assignment 1 – 3 is completed.

## Criteria:

* Effectiveness of the AI-enabled code analyzer prototype.
* Accuracy of vulnerability detection.
* Quality of AI-based recommendations for fixes.
* Documentation and usability of the code analyzer.

## Expectations:

* Development of a functional AI-enabled Software Code Analyzer prototype.
* Effective identification and categorization of software vulnerabilities.
* AI-based recommendations for fixing vulnerabilities.
* Detailed documentation on the operation and usage of the code analyzer.

## Methods for Assessment:

* Evaluation of the functioning code analyzer.
* Assessment of the accuracy of vulnerability detection.
* Review of the quality of AI-based recommendations.
* Review of documentation.
* Software Development Review Approval

# Assignment 5: Project Presentation and Software Demonstration (Weight: 30%)

## Criteria:

* Clarity and organization of the project presentation.
* Comprehensive coverage of the project, including design and development.
* Quality of the software demonstration.
* Ability to answer questions and defend the project.

## Expectations:

* An engaging and informative project presentation.
* Comprehensive coverage of the project, including its design, development, and testing.
* A high-quality software demonstration showcasing the AI-enabled code analyzer.
* Ability to respond to questions and defend the project effectively.

## Methods for Assessment:

* Evaluation of the project presentation and software demonstration.
* Final Project Review Approval.
* 1 – 2 Page Summary Paper on the project and its results.

# Progress Reports and Weekly Touchpoints (Weight: 5%)

## Criteria:

* Regularity and consistency in progress reporting.
* Clarity in reporting achievements, challenges, and next steps.
* Demonstrated responsiveness to feedback.

## Expectations:

* Regular weekly progress reports.
* Clear presentation of achieved milestones and challenges faced.
* Demonstrated adaptation to feedback and evolving project needs.

## Methods for Assessment:

* Evaluation of progress reports at the end of each week.

# Technical Reviews (Grading Part of Assignments):

## Project Kick Off Meeting

* **Description:** The Project Kick Off Meeting is a crucial initial gathering aimed at launching the project and ensuring a shared understanding among all stakeholders. It is leveraged to set the project in motion and manage expectations effectively.
* **Entrance Criteria:**
  + Project Plan and objectives are defined.
  + Potential stakeholders are identified and available for the meeting.
  + Initial timelines and milestones are prepared.
* **Exit Criteria:**
  + Clear project goals, objectives, and scope have been established and documented.
  + Stakeholder approvals are aligned on project expectations, roles, and responsibilities.
  + Shared understanding of the initial project plan.

## Stakeholder and Software Requirements Review (SSRR)

* **Description:** The Stakeholder and Software Requirements Review (SSRR) takes place at the end of the Requirements Phase of the Software Development Lifecycle (SDLC) and upon the completion of Assignment 2. Its purpose is to ensure that all stakeholder requirements have been captured accurately and that software functional and non-functional requirements are well-defined.
* **Entrance Criteria:**
  + The Requirements Phase is completed, including stakeholder interviews and analysis.
  + Stakeholder requirements are documented and validated.
  + Software functional and non-functional requirements are defined and associated with stakeholder needs.
* **Exit Criteria:**
  + All stakeholder requirements are documented, validated, and traceable.
  + Software functional and non-functional requirements are clear and well-documented.
  + A Requirements Traceability Matrix (RTM) and Requirement Verification Matrix (RVM) are created to trace requirements to test cases and verification methods.

## Software Design Review (SDR)

* **Description:** The Software Design Review (SDR) takes place at the end of the Design Phase of the SDLC and upon the completion of Assignment 3. Its purpose is to assess the quality and effectiveness of the software design, ensuring that it aligns with the established requirements.
* **Entrance Criteria:**
  + The Design Phase is completed, including the creation of software design documentation.
  + Architectural diagrams (e.g., Component, Package, Class diagrams) and behavior diagrams (e.g., Use Cases, Sequence Diagrams) are finalized.
  + Requirements Allocation Matrix is prepared, linking design elements to requirements.
* **Exit Criteria:**
  + Software design documentation is comprehensive and aligned with specified requirements.
  + Architectural diagrams effectively convey system structure and behavior.
  + Requirements Allocation Matrix is complete and accurately reflects the design.

## Software Development Review (SDevR)

* **Description:** The Software Development Review (SDevR) takes place at the end of the Development Phase of the SDLC and upon the completion of Assignment 4. While early views of progress are demonstrated during Weekly Touchpoints, this review provides a more comprehensive assessment of the development phase, ensuring that the code analyzer prototype is on track.
* **Entrance Criteria:**
  + The Development Phase, including the creation of the AI-enabled code analyzer prototype, is baselined, and requires minimal updates.
  + Weekly Touchpoints have been conducted to demonstrate ongoing progress.
* **Exit Criteria:**
  + The AI-enabled code analyzer prototype is functional and effective in identifying vulnerabilities.
  + AI-based recommendations for fixing vulnerabilities are of high quality.
  + Documentation on the code analyzer's operation and usage is available and clear.

## Final Project Review:

* **Description:** The Final Project Review takes place at the end of the entire project and upon the completion of Assignment 5. Its purpose is to provide a comprehensive review of the entire effort, including findings, results, lessons learned, and to assess the overall success of the project.
* **Entrance Criteria:**
  + All project phases, including requirements, design, development, and testing, are completed.
  + The software code analyzer prototype is fully functional and documented.
  + Lessons learned from the project have been identified and documented.
* **Exit Criteria:**
  + A comprehensive review of the project effort is conducted, including its goals, achievements, and challenges.
  + Findings and results from the project, including the AI-enabled code analyzer's performance, are presented.
  + Lessons learned are discussed, and recommendations for future projects are provided.
  + The project is assessed for its overall success and contribution to the field of Software Security integrating AI.

These reviews play a critical role in ensuring the successful execution of the independent study course, providing checkpoints for assessment, feedback, and alignment with project goals and requirements at various stages of development.

Overall Assessment:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Final Grade Cutoffs** | | | | | | | | | |
| + | 97.00% | + | 87.00% | + | 77.00% | + | 67.00% |  |  |
| A | 93.00% | B | 83.00% | C | 73.00% | D | 63.00% | F | <60.0% |
| - | 90.00% | - | 80.00% | - | 70.00% | - | 60.00% |  |  |

This updated grading rubric aligns with the course objectives and emphasizes the importance of project planning, requirements documentation, software design, prototype development, and effective communication in the context of AI-enabled software security. It also ensures continuous assessment through weekly progress reports.

A screenshot of a computer

Description automatically generated