Contents

[1. Overview 4](#_Toc195733367)

[1.1. Project summary 4](#_Toc195733368)

[1.2. Project deliverables 6](#_Toc195733369)

[1.4. References 7](#_Toc195733370)

[1.5. Definitions 7](#_Toc195733371)

[2. Project organization 8](#_Toc195733372)

[2.1. Process model 8](#_Toc195733373)

[2.2. Project organizational structure (Internal structure) 8](#_Toc195733374)

[2.3. Organizational boundaries and interfaces (External interfaces) 8](#_Toc195733375)

[2.4. Project roles and responsibilities 9](#_Toc195733376)

[3. Management process 11](#_Toc195733377)

[3.1. Management scope and priorities 11](#_Toc195733378)

[3.2. Supposition, dependencies and restriction 12](#_Toc195733379)

[3.3. Risk management 16](#_Toc195733380)

[3.3.1. Risk Project Management Analysis 18](#_Toc195733381)

[3.4. Monitoring and controlling mechanisms 19](#_Toc195733382)

[3.5. Staffing Plan 20](#_Toc195733383)

[4. Technical process (Supporting process plans) 22](#_Toc195733384)

[4.1. Methods, tools and techniques 22](#_Toc195733385)

[4.2. Software Documentation 24](#_Toc195733386)

[4.3. Project support functions 27](#_Toc195733387)

[4.3.1. Software quality assurance 27](#_Toc195733388)

[**2.REVIEWS AND AUDITS** 38](#_Toc195733389)

[**3.PROBLEM REPORTING AND CORRECTIVE ACTION** 40](#_Toc195733390)

[4.3.2. Software Testing 50](#_Toc195733391)

[5. Work packages, timetable and budget 55](#_Toc195733392)

[5.1. Work packages, and budget 55](#_Toc195733393)

[5.2. Dependencies 56](#_Toc195733394)

[**5.3. Resource requirements** 57](#_Toc195733395)

[5.4. Timetable 59](#_Toc195733396)

[ANNEXES 61](#_Toc195733397)

# 1. Overview

## 1.1. Project summary

*Purpose, scope, and objectives*

**Purpose**

The purpose of the Emotional Pedagogical Agent (EPA) project is to enhance the interaction between educators and learners by developing an EPA that integrates AI features into e-learning environments. This project aims to address the challenges posed by the increasing use of digital solutions and to reduce reliance on traditional learning materials.

**Scope:**

* Design and development of the Emotional Pedagogical Agent (EPA). Additional information in 001.SCD.EPA\_1.0.
* An analysis of existing pedagogical agents.
* Integration of AI functions into the EPA.
* Integration of the EPA into selected e-learning platforms.
* Evaluation of the EPA's impact on learner engagement and knowledge retention.

Additional information in Business Model Canvas.EPA.

**Objectives:**

1. Identify and analyze existing solutions for pedagogical agents within three months.
2. Design and develop an EPA prototype within six months.
3. Integrate the EPA into selected e-learning platforms within nine months.
4. Evaluate the effectiveness of the EPA in enhancing learner engagement and knowledge retention within twelve months.

*Refers to* **3.1. Management scope and priorities**, **4.1. Methods, tools and techniques**

*Assumptions and constraints*

**Assumptions:**

* 1. The project will leverage machine learning, natural language processing (NLP), and affective computing to create a highly responsive and engaging educational experience using AI functions.
  2. The e-learning platform will support the integration of the EPA without requiring significant modifications.
  3. The target audience has access to the necessary equipment and the e-learning platform.
  4. There will be no major changes in educational regulations or standards during the project timeline.

**Constraints:**

* + The project budget is limited.
* The project must be completed within a specified time.

*Refers to* **5.4. Timetable.**

* + The EPA must comply with data privacy regulations (GDPR) and AI usage regulations of the EU.
  + The e-learning platform must support Latvian and English languages to cater to a diverse user base.
  + The system must be compatible with existing infrastructure and technology used by the educational institution, which is *ORTUS (Riga Technical University).*

*Refers to* **3.2. Supposition, dependencies and restriction**

*Schedule and budget summary*

**Schedule of the project:**

1. **Project initiation and planning**: March 24 to April 22, 2025
2. **Requirements analysis**: March 16 to April 30, 2025
3. **Design and prototyping**: May 5 to June 30, 2025
4. **Basic feature development phase**: July 1 to August 31, 2025
5. **Advanced feature development phase**: September 1 to October 31, 2025
6. **Integration and testing**: November 1 to December 30, 2025
7. **User training and documentation**: January 2 to January 31, 2026
8. **Project deployment and go-live**: February 1 to February 28, 2026

**Budget summary**

This project’s total budget is 50’000 euros:

*Refers to* **5.1. Work packages, and budget**.

## 1.2. Project deliverables

The deliverables for the EPA project are:

1. Project charter: 1 electronic copy (PDF), delivery date April 22, 2025 to office in Liepaja.
2. Requirements specification document: 1 electronic copy (PDF), delivery date April 30, 2025, to office in Liepaja.
3. Design documents and prototypes: 1 set of electronic copies (PDF), and interactive prototypes of EPA, delivery date June 30, 2025, to office in Liepaja.
4. Basic feature development phase 1: 1 set of electronic copy (source code), delivery date August 31, 2025, to office in Liepaja.
5. Advanced feature development phase 2: 1 set of electronic copy (source code), delivery date October 31, 2025, to office in Liepaja.
6. Integration and testing report: 1 electronic copy (PDF), delivery date December 30, 2025, to office in Liepaja.
7. User manuals and training materials: 1 set of electronic copies (PDF, video tutorials), delivery date January 31, 2026, to office in Liepaja.
8. Final project report and system deployment: 1 set of printed documents (user manuals), electronic copies (PDF), and system deployment package.

**Delivery Media and Special Instructions:**

* All printed documents should be packed in materials that prevent damage during transit.
* Electronic copies should be delivered via secure file transfer.

It is important that all printed documents should be packed in materials that prevent damage during transit. Electronic copies should be delivered via secure file transfer.

*Refers to* **4.2. Software Documentation**

**1.3.** **Evolution of the SPMP**

Scheduled updates to the SPMP will be produced at regular intervals to ensure the document remains current and reflects the project’s progress and any changes in scope, schedule, or resources. It is the Project Manager's responsibility to update the SPMP in time.

**Update Schedule:**

* **Minor updates:** Quarterly
* **Major updates:** Semi-annually

Minor updates will include routine revisions, such as updates to the project schedule, resource allocations, and minor scope changes. Major updates will include significant revisions, such as changes to the project scope, major milestones, and critical path adjustments.

**Dissemination Methods:**

1. **Email:** All updates will be emailed to the project stakeholders.
2. **Project repository:** Updates will be uploaded to the project’s document repository.
3. **Meeting:** Key updates will be discussed in project meetings and steering committee meetings.

All updates will be maintained by a control log version. The log will include the following information:

* Version Number
* Description of change
* Author
* Date of change

## 1.4. References

IEEE Std 1058-1998: IEEE Standard for Software Project Management Plans.

IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications.

Recent research on affective computing and emotional AI in education. (will be added specific information)

*Refers to:* **4.3.2. Software Testing**

## 1.5. Definitions

* **Emotional Pedagogical Agent (EPA):** AI-powered system that supports learners by recognizing emotions and adapting pedagogical strategies.
* **Affective Computing:** Technology that interprets human emotions to improve human-computer interaction.
* **Natural Language Processing (NLP):** AI technology that enables machines to understand and generate human language.

# 2. Project organization

## 2.1. Process model

The project follows an **Agile development methodology, Scrum framework**, incorporating iterative development, regular stakeholder feedback, and continuous improvement. This life cycle model is to be used for this project in the Agile Scrum model. Main processes are project initiation, project termination, product development, and product release. This project supports processes that are configuration management, quality assurance, risk management, and communication management.

*Refers to* **3.1. Management scope and priorities**, **4.1. Methods, tools and techniques**

## 2.2. Project organizational structure (Internal structure)

Organizational Structure describes the internal management structure of the project, as well as how the project relates to the rest of the organization. This project internal structure organizational chart consists of previously mentioned roles – Project manager, requirements analyst, QA Lead and team, designer and development team.

A diagram of a project manager

AI-generated content may be incorrect.*Refers to* **2.4. Project roles and responsibilities**

Figure of Project Organization Chart

## 2.3. Organizational boundaries and interfaces (External interfaces)

External organizational boundaries of this project:

* Parent organization – Riga Technical University (RTU) external interfaces are that Project Manager reports to the RTU project oversight committee with regular status updates and progress reports.
* Acquiring organization – Ministry of Education and Science of Latvia regularly communicates with the Project Manager and participates in milestones reviews and project evaluations.
* Subcontracted organizations – software development vendor reports to the Lead Developer with regular progress meetings and code reviews. Quality assurance vendor reports to the QA lead by providing test results and defect reports.

Other organizational entities could be external consultants that provide consultations with relevant project team members and deliver reports, recommendations, and vendors who coordinate with the Project Manager for procurement and delivery.

*Refers to* **2.2. Project organizational structure (Internal structure)**

## 2.4. Project roles and responsibilities

In this project there are at least 9 major work activities and 3 supportive processes:

* Project initiation – defining project objectives and scope and conduct initial risk assessment.
* Requirements analysis – gather and document project requirements.
* Design and prototyping – create system architecture and design documents, review final design.
* Basic feature development phase – implementing of core functionalities, testing and refining code.
* Advanced feature development phase – implementing of advanced features and enhancements, testing and refining code.
* Integration and testing – perform system integration, and user acceptance testing, resolve defects and issues.
* User training and documentation – develop user manuals and training materials, and final documentation deployment.
* Project deployment – prepare deployment plan, deploy system and monitor system performance.
* Project termination – conduct project closure meeting, document project experience, release project resources.
* Quality assurance – develop QA plan, conduct testing, perform audits and reviews.
* Risk managements – identify and assess risks, monitor and update risk register.
* Communication management – develop communication plan, conduct regular status meetings.

|  |  |  |
| --- | --- | --- |
| **Role** | **Responsibilities** | **Person** |
| Project Manager | Oversee project initiation and termination. |
| Manage risks and mitigation strategies. |
| Handle communication management among stakeholders. |
| Oversee project deployment and ensure timely delivery. |
| Requirements team | Perform requirements analysis to gather and document functional and non-functional requirements. |
| Ensure requirements align with project objectives. |
| Development team | Design and create prototypes for the EPA. | <name #1>  <name #2> |
| Implement basic features and functionalities. |
| Develop advanced features, including emotional recognition and pedagogical strategies. |
| QA team | Perform integration and system testing of the EPA. |
| Conduct quality assurance to ensure the system meets performance and usability standards. |
| Contribute to risk management by identifying potential quality risks and mitigation strategies. |
| Documentation team | Create user manuals and training materials for educators and students. |
| Document technical details, including system architecture and configurations. |

Table F-n. Project Responsibilities

*Refers to* **2.2. Project organizational structure (Internal structure)**

# 3. Management process

## 3.1. Management scope and priorities

**Scope:** Develop, integrate, and deploy an AI-enabled pedagogical agent.

Additional information in IDEF0.drawio.

1. **Features and Functionalities**:
   * + Emotional recognition.
     + Personalized feedback.
     + Adaptive learning recommendations.
     + Seamless interaction.
   * Include advanced functionalities, such as integration with natural language processing (NLP) and sentiment analysis.
2. **Target Audience**:
   * Primary users: educators, students, and e-learning administrators.
   * Ensure the system adapts to different levels of user expertise.
3. **Supported Platforms**:
   * Compatibility with specific e-learning platforms (e.g. ORTUS) or as a standalone solution.
   * Support for multiple devices: desktop, mobile, and tablet.
4. **Integration Goals**:
   * Ensure seamless integration with existing learning management systems (e.g., video conferencing tools, content libraries).

**Priorities:** Enhance interaction quality, ensure seamless integration with existing e-learning platforms (ORTUS), and deliver user documentation.

1. **User Experience**:
   * Emphasize creating an intuitive and user-friendly interface for both educators and learners.
2. **Scalability and Performance**:
   * Design the solution to handle large numbers of simultaneous users without performance degradation.
   * Optimize for scalability to support future growth and additional features.
3. **Security and Privacy**:
   * Address data protection requirements, ensuring compliance with GDPR or other relevant regulations.
   * Implement encryption and secure data handling practices to protect user data and interactions.
4. **Feedback and Iteration**:
   * Build mechanisms for collecting user feedback during and after deployment, for example a survey.
   * Incorporate iterative updates based on feedback to continuously improve the system’s functionality and user experience.

*Refers to* **1.1. Project summary***,* **2.1. Process model**

## 3.2. Supposition, dependencies and restriction

The following are the key work activities for the development of the emotional pedagogical agent:

* Requirements Gathering and Analysis
  + Conduct stakeholder meetings and workshops.
  + Document functional and non-functional requirements.
  + Create requirements specification document.
* Design and Prototyping
  + Develop architecture design.
  + Create detailed design documents.
  + Develop prototypes for user feedback.
* Implementation:
  + Set up the development environment.
  + Implement the emotional pedagogical agent features.
  + Conduct unit testing for individual components.
* Integration and Testing
  + Integrate all components.
  + Conduct integration testing.
  + Perform system testing and user acceptance testing.
* Deployment
  + Prepare deployment plan.
  + Deploy the software to the production environment.
  + Conduct post-deployment verification.
* Maintenance and Support
  + Provide ongoing maintenance and support.
  + Address any issues or bugs reported by users.
  + Implement updates and improvements as needed.

The project schedule is divided into phases with estimated timelines.

* Requirements Gathering and Analysis:
  + Start Date: 10.03.2025
  + End Date: 31.03.2025
  + Duration: 3 weeks
* Design and Prototyping:
  + Start Date: 01.04.2025
  + End Date: 30.04.2025
  + Duration: 4 weeks
* Implementation:
  + Start Date: 01.05.2025
  + End Date: 31.07.2025
  + Duration: 3 months
* Integration and Testing:
  + Start Date: 01.08.2025
  + End Date: 31.08.2025
  + Duration: 4 weeks
* Deployment:
  + Start Date: 01.09.2025
  + End Date: 15.09.2025
  + Duration: 2 weeks
* Maintenance and Support
  + Start Date: 16.09.2025
  + Ongoing

The estimated budget for the project includes the following cost of required resources.

* Human Resources:[[1]](#footnote-1)
  + Project Manager: 1500 euro in gross/month in 12 months 18 000
  + Software Developers: 1100 euro in gross/month in 6 months 6600
  + QA Engineers: 1100 euro in gross/month in 6 months 6600
  + Designers: 1100 euro in gross/month in 6 months 6600
* Hardware Resources:
  + Development Workstation: 2000 euro each (2 workstations)
  + Testing Server: 1000 euro (2 testing servers)
  + Production Server: 1000 euro
  + Laser Printer: 400 euro
* Software Resources:
  + IDE Licenses: 200 euro/license (2 licenses)
  + Testing Tools: 300 euro/license (2 licenses)
  + Version Control System: Free
  + Project Management Tools: 800 euro/year
* Miscellaneous:
  + Training: 1000 euro
  + Contingency: 2000 euro
* Total Estimated Budget (Budget must be reviewed)
  + Human Resources: 37800 euro
  + Hardware Resources: 7400 euro
  + Software Resources: 1800 euro
  + Miscellaneous: 3000 euro
  + Total: 50000 euro

Measuring Changes:

* Traceability Matrices: Maintain traceability matrices to track changes to requirements and their impact on design, implementation, and testing.
* Change Log: Maintain a change log to document all changes to requirements, including the description, rationale, and stakeholders involved.

Reporting Changes:

* Change Reports: Generate regular change reports to inform stakeholders of any changes to requirements. These reports will include details of the changes, their impact, and the status.
* Status Meetings: Conduct regular status meetings to discuss proposed or approved changes to requirements and their impact on the project.

Controlling Changes:

* Change Control Procedures: Implement formal change control procedures to manage changes to requirements. This includes submitting change requests, performing impact analysis, and obtaining approval from the Project Manager.

Impact on Product Quality

* Impact Analysis: Perform impact analysis to assess the effect of requirements change on the product scope and quality. This includes evaluating how the changes affect the overall functionality, usability, and performance of the emotional pedagogical agent.
* Prototyping and Modeling: Use prototyping and modeling techniques to visualize and assess the impact of changes on the product. This helps stakeholders understand the implications of the changes and make informed decisions.

Impact on Project Schedule, Budget, Resources, and Risk Factors

* Schedule Impact: Analyze the impact of requirements changes on the project schedule. This includes identifying any additional tasks, changes to the timeline, and potential delays.
* Budget Impact: Assess the impact of requirements changes on the project budget. This includes estimating the cost of additional resources, tools, and potential rework.
* Resource Impact: Evaluate the impact of requirements change on project resources. This includes assessing the availability and allocation of team members, tools, and other resources.
* Risk Impact: Identify and assess the impact of requirements changes on project risks. This includes evaluating how the changes may introduce new risks or affect existing risks and updating the risk management plan accordingly.

Change Control Procedures

* Change Request Submission
  + Stakeholders submit change requests through a formal process.
  + Change requests include a description of the change, rationale, and potential impact.
* Impact Analysis
  + Perform impact analysis to assess the effect of the proposed change on product scope, quality, schedule, budget, resources, and risks.
  + Document the results of impact analysis.3
* Implementation of Approved Changes
  + Approved changes are implemented according to the project plan.
  + Update relevant project documents, including requirements specifications, design documents, and test plans.
* Verification and Validation
  + Verify and validate the implemented changes to ensure they meet the new requirements.
  + Conduct reviews and testing to ensure the quality of the changes.

Control Mechanisms

* Measuring Progress
  + Gantt Chart: Use a Gantt chart to visualize the project timeline, including start and end dates for each task. Update the Gantt chart regularly to reflect actual progress.
  + Milestone Reviews: Conduct milestone reviews at the completion of each major and minor milestone. Evaluate the scope and quality of work products completed at each milestone using predefined criteria.
  + Progress Reports: Generate regular progress reports (weekly) to document actual progress against planned progress. Include details of completed tasks, upcoming tasks, and any deviations from the plan.
* Comparing Actual Progress to Planned Progress
  + Schedule Variance: Calculate schedule variance to compare actual progress to planned progress. SV = Earned Value (EV) - Planned Value (PV). A negative SV indicates a delay, while a positive SV indicates ahead of schedule.
  + Earned Value Management: Use EVM to measure project performance and progress. Key metrics include Planned Value (PV), Earned Value (EV), and Actual Cost (AC).
  + Critical Path Analysis: Perform critical path analysis to identify tasks that directly impact the project’s completion date. Monitor the critical path to ensure timely completion of critical tasks.
* Implementing Corrective Actions
  + Deviation Analysis: Analyze deviations from the planned schedule to identify root causes. Document the reasons for deviations and potential corrective actions.
  + Corrective Action Plan: Develop a corrective action plan to address schedule deviations. Define specific actions, responsible personnel, and timelines for implementing corrective actions.
  + Regular Monitoring: Monitor the implementation of corrective actions to ensure their effectiveness. Adjust the corrective action plan as needed based on ongoing evaluation.

Methods and Tools

* Methods
  + Work Breakdown Structure: Use WBS to decompose the project into smaller, manageable tasks. Assign tasks to team members and define start and end dates.
  + Milestone Chart: Use a milestone chart to track the achievement of major and minor milestones. Define objective criteria for assessing the scope and quality of work products at each milestone.
* Tools
  + Project Management Software: Use project management software to plan, track, and manage project schedules. Update the project schedule regularly to reflect actual progress.
  + Progress Tracking Tools: Use progress tracking tools to monitor the status of individual tasks and milestones. Provide real-time updates on task completion and any delays.

Objective Criteria for Milestone Achievements

* Scope of Work Products
  + Define the scope of work products to be completed at each milestone.
  + Ensure that the work products meet the defined requirements and specifications.
* Quality of Work Products
  + Assess the quality of work products using predefined quality criteria.
  + Conduct reviews and testing to ensure that work products meet quality standards.
* Documentation
  + Ensure that all required documentation is completed and reviewed at each milestone.
  + Include design documents, test plans, user manuals, and other relevant documentation.

*Refers to* **1.1. Project summary**

## **3.3. Risk management**

The risk management plan outlines the approach for identifying, analyzing, and prioritizing project risk factors. It describes the procedures for contingency planning, methods for tracking risk factors, evaluating changes in risk levels, and responding to those changes. The plan ensures ongoing identification, assessment, and mitigation of risk factors throughout the project lifecycle.

**Purpose**

To identify, assess, prioritize, and mitigate risks that could affect the successful completion of the AI-enabled pedagogical agent project.

**Risk Project Management Table**

| **#** | **Title** | **Category** | **Prob.** | **Impact** | **Reaction** | **Owner** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Budget Overruns | Budget Risks | Med | Med | Regular budget reviews | Project Manager |
| 2 | Skill Gaps | Personnel Risks | High | High | Arrange training sessions | Risk Manager |
| 3 | Timeline Delays | Schedule Risks | Med | High | Frequent milestone tracking | Project Manager |
| 4 | Stakeholder Misalign. | Acquirer Risks | Med | High | Regular updates to stakeholders | Project Manager |
| 5 | Resource Shortages | Resource Risks | High | Med | Secure resources early | Project Manager |
| 6 | Regulatory Compliance | Compliance Risks | Med | High | Regular compliance audits | Compliance Lead |
| 7 | Communication Issues | Personnel Risks | Med | Med | Weekly team syncs | Project Manager |
| 8 | Tool Delays | Schedule Risks | High | Med | Early procurement planning | Procurement Lead |
| 9 | Change Management | Management Risks | Med | High | Strict change control processes | Risk Manager |
| 10 | Risk Awareness | Management Risks | High | Med | Conduct risk awareness sessions | Risk Manager |

### 3.3.1. Risk Project Management Analysis

1. **Key Risks**:
   * **Budget Overruns** and **Timeline Delays** are medium-to-high risks due to a fixed budget of €50,000 and a strict deadline (February 28, 2026). These factors require careful resource and time management.
   * **Skill Gaps in Team** is a high-probability and high-impact risk that may significantly affect project milestones. Training and mentoring programs must be prioritized.
   * **Stakeholder Misalignment** highlights the need for a strong communication plan to avoid misaligned goals or unclear requirements.
2. **Primary Focus**:
   * Ensure adequate training and prepare the team early.
   * Implement regular project milestone tracking to avoid timeline slippage.
   * Maintain frequent communication with stakeholders (e.g., sponsors, educators).
3. **Mitigation Strategies**:
   * Use preventive measures like budget monitoring and resource allocation.
   * Ensure compliance with GDPR and legal standards to avoid complications.
4. **Overall Assessment**: The project management risks are manageable with proper planning, but high-priority risks like skill gaps and misalignment must be addressed immediately.

*Refers to* **4.1. Methods, tools and techniques**

## 3.4. Monitoring and controlling mechanisms

**Budget Control**

* Conduct **bi-weekly financial reviews** to ensure the project stays within the fixed budget of **€50,000**.
* Track expenses related to **AI tools, training, API integration, and testing** separately to identify cost-intensive areas early.
* Use a **budget tracking tool**  *Monday.com* to monitor spending and allocate contingency funds effectively.
* Maintain a dynamic budget tracking sheet to document and analyze real-time expenses.

*Refers to* **5.1. Work packages, and budget**.

**Schedule Control**

* Perform **weekly progress checks** to ensure milestones are met according to the project timeline (completion by **February 28, 2026**).
* Use **Gantt charts** or **project management software** to track task completion and dependencies.
* Address delays proactively by reallocating resources or adjusting non-critical tasks.
* Include **a buffer of 10%** in the project schedule to accommodate unforeseen delays.

*Refers to* **1.1. Project summary**, **5.4. Timetable**

**Quality Control**

* Conduct **peer reviews** for all major deliverables, including code, AI models, and system architecture.
* Implement **automated testing** for the AI agent's functionality, including integration with e-learning platforms.

*Refers to* **4.3.3. Software Testing**

* Collect **user feedback** in *ORTUS* (from educators and students) at each milestone to ensure the pedagogical agent meets user expectations.
* Ensure compliance with **GDPR[[2]](#footnote-2)** or data protection and accessibility standards.

**Reporting**

* Prepare **monthly progress reports** summarizing:
  + Budget usage and remaining funds.
  + Task completion status and upcoming milestones.
  + Identified risks and mitigation efforts.
  + Feedback received from users and stakeholders.
* Share reports with **project stakeholders**, including sponsors, educators, and technical leads, to maintain transparency.
* Schedule **stakeholder meetings** every two months to review progress and realign priorities if needed.

**Tools and Techniques**

* Use **project management software** Jira for tracking tasks and risks.
* Employ **version control systems** GitHub to monitor code quality and changes.
* Utilize **user feedback tools** surveys in *ORTUS* to gather input from students and educators

## 3.5. Staffing Plan

**1. Required Skills and Expertise**

**Technical Skills:**

* **AI Development**: Proficiency in machine learning frameworks (e.g., TensorFlow, PyTorch) and natural language processing (NLP).
* **API Integration**: Experience with RESTful APIs and integration with Learning Management Systems (LMS), such as ORTUS.
* **Software Testing**: Expertise in automated testing tools and frameworks.
* **UI/UX Design**: Skills in creating accessible and user-friendly interfaces, using WCAG 2.2 Standart[[3]](#footnote-3) (Web Content Accessibility Guidelines).

**Functional Skills:**

* **Project Management**: Experience with Agile methodologies and risk management.
* **Compliance Expertise**: Familiarity with GDPR and data privacy standards.
* **Educational Expertise**: Understanding of pedagogical methods and student learning processes.

**Team Composition:**

| **Role** | **Number of Personnel** | **Experience (Years)** | **Responsibilities** |
| --- | --- | --- | --- |
| Senior Developer | 1 | 5+ | Lead AI development and system architecture. |
| Junior Developer | 1 | 1-2 | Assist with coding, debugging, and feature implementation. |
| QA Engineer | 1 | 3+ | Conduct automated testing and ensure quality. |
| UI/UX Designer | 1 | 3+ | Design user-friendly and accessible interfaces. |
| Project Manager | 1 | 5+ | Oversee project execution, timelines, and budget. Ensure GDPR compliance and manage data privacy. Provide insights into pedagogical methods and usability. |

**2. Personnel Sources**

**Internal Transfers:**

* Identify existing team members with relevant skills (e.g., developers familiar with LMS systems) for reassignment.

**Recruitment:**

* **Roles**: Junior Developer, QA Engineer.
* **Methods**: Job postings on university career portals, referrals, and partnerships with educational institutions.

**Outsourcing:**

* Engage external contractors for specialized tasks:
  + Accessibility audits for WCAG compliance.
  + Scalability testing of the AI agent under user load.

**3. Training Plan**

* **Workshops**: Conduct sessions on AI tools (TensorFlow, PyTorch) and compliance (GDPR, WCAG).
* **Training Provider**: Use internal experts for AI and external trainers for GDPR and accessibility.
* **Schedule**: Complete training within the first 2 months of the project.

**4. Staffing Timeline**

* **Initial Phase**:
  + Senior Developer, Junior Developers.
  + Begin training sessions.
* **Gradual Hiring**:
  + Add QA Engineer, UI/UX Designer during the second phase.
* **Onboarding Schedule**:
  + Allocate 2 weeks for orientation and familiarization with project goals.

**5. Roles and Responsibilities**

* **Senior Developer**: Lead technical implementation and mentor junior developers.
* **Junior Developers**: Assist with coding, debugging, and implementing features.
* **QA Engineer**: Test all deliverables and maintain quality standards.
* **UI/UX Designer**: Ensure the interface is user-friendly and accessible.
* **Project Manager**: Manage timelines, budget, and stakeholder communication. Monitor GDPR compliance and data privacy measures. Provide feedback on usability and student needs.

**6. Staffing Contingency Plan**

* **Cross-Training**: Ensure team members are trained in multiple roles to mitigate shortages.
* **Backup Resources**: Maintain a pool of freelancers for urgent needs.

**7. Performance Monitoring**

* **Metrics**:
  + Task completion rate.
  + Adherence to deadlines.
  + Code quality and testing success rates.
* **Feedback Mechanism**: Regular performance reviews and team meetings to address issues.

*Refers to***2.4. Project roles and responsibilities**, **5.1. Work packages, and budget**

# 4**. Technical process (**Supporting process plans)

## 4.1. Methods, tools and techniques

**Development Methodologies:**

* **Agile Development**: The project will use Agile methodologies to ensure iterative progress and continuous feedback.
* **Scrum Framework**: Scrum will be employed to manage tasks and sprints effectively, ensuring timely delivery of features.

**Programming Languages and Notations:**

* **Python**: Used for AI development and natural language processing (NLP).
* **JavaScript**: Used for front-end development and interactive features.
* **HTML/CSS**: For UI/UX design.

**Tools and Techniques:**

* **TensorFlow/PyTorch**: For AI model development and training.
* **Postman**: For API development and testing.
* **GitHub**: For version control and collaboration.
* **Visual Studio Code (VS Code)**: IDE for coding.

**Testing:**

* **Selenium**: For UI/UX testing.
* **PyTest**: For unit and integration testing.
* **JMeter**: For performance and load testing.

**Documentation:**

* **Microsoft Word/ One Drive**: For creating deliverable documentation.

**Standards and Policies:**

* **GDPR Compliance**: All features must adhere to data protection regulations.
* **WCAG 2.2 Standards**: User interface must be accessible to individuals with disabilities.
* **IEEE 830**: For requirements specification.
* **Code Review Policy**: Mandatory peer reviews for all commits.

**Techniques:**

* **Risk Management**: Identify and mitigate risks during each sprint or milestone.
* **Continuous Integration/Continuous Deployment (CI/CD):** Automate builds, testing, and deployment using GitHub Actions.
* **Model Validation:** Regular testing and validation of AI models to ensure accuracy.

*Refers to* **1.4. References**,3.3. Risk management *,***4.3.2. Software Testing**

## 4.2. Software Documentation

**Nondeliverable Work Products**

| **Document** | **Purpose** | **Responsible Entity** | **Reviewer** | **Template/Standard** | **Distribution List** |
| --- | --- | --- | --- | --- | --- |
| Design Documentation | Document system architecture and AI model design. | Senior Developer | Project Manager | UML Diagrams | Development Team |
| Traceability Matrices | Map requirements to design and testing to ensure coverage. | Project Manager | QA Engineer | Custom Template | Team, QA Team |
| Test Plans | Define objectives, scope, and methods for testing. | QA Engineer | Developers | IEEE 829 | QA Team |
| Meeting Minutes and Reports | Record decisions and discussions during project meetings. | Project Manager | Project Manager | Custom Template | Team, Stakeholders |

**Deliverable Work Products**

| **Document** | **Purpose** | **Responsible Entity** | **Reviewer** | **Template/Standard** | **Distribution List** |
| --- | --- | --- | --- | --- | --- |
| Requirements Specification | Define functional and non-functional requirements of the project. | Project Manager | Stakeholders | IEEE 830 | Team, Stakeholders |
| Source Code | Provide the system’s implementation. | Developers | QA Engineer | Code Style Guide | Development Team |
| Object Code | Compiled and executable code for deployment. | Developers | QA Engineer | Build Process Guide | QA Team, Deployment Team |
| User’s Manual | Guide for end users to interact with the pedagogical agent. | Project Manager | QA Engineer | IEEE 1063 | Users, Educators |
| Online Help System | Integrated help and troubleshooting for end users. | Project Manager | UI/UX Designer | HTML-Based Template | Users, Educators |
| Regression Test Suite | Enable automated testing for future updates. | QA Engineer | Developers | PyTest Framework | Development and QA Teams |
| Configuration Library | Store all project artifacts (e.g., code, datasets, models). | Project Manager | Developers | GitHub Repository | Development Team |
| Maintenance Guide | Provide instructions for maintaining the system post-deployment. | Developers | Project Manager | Custom Template | Operations Team |

## 4.3. Project support functions

### 4.3.1. Software quality assurance

**Quality Assurances Procedures**

Additional information in QFD.EPA.

**Analysis**

* Conduct thorough analysis of requirements, design, and implementation to ensure compliance with standards and specifications.

**Inspections**

* Perform systematic inspections of code, design documents, and other project artifacts to identify defects and areas for improvement.

**Reviews**

* Conduct regular reviews of project deliverables, including requirements specifications, design documentation, and test plans.
  + **Software Requirements Review**: Ensures the adequacy of the requirements.
  + **System Architecture Review**: Ensures compliance with the State Technical Architecture.
  + **Technical Design Review**: Evaluates the technical adequacy of the design.
  + **Software Verification and Validation Plan Review**: Assesses the completeness of verification and validation methods.
  + **Software Configuration Management Plan Review**: Evaluates configuration management methods.
  + **Implementation Plan Review**: Verifies consistency between software and documentation.
  + **Post-implementation Maintenance Plan Review**: Assesses development activities and provides recommendations.

**Audits**

* Conduct periodic audits to ensure adherence to project management practices, quality standards, and regulatory requirements.

**Assessments**

* Perform assessments to evaluate the effectiveness of the quality assurance processes and identify areas for improvement.

**Relationships Among Processes**

* **Quality Assurance**: Ensures that the project meets the defined quality standards.
* **Verification and Validation**: Confirms that the product meets the specified requirements and performs as expected.
* **Reviews**: Provides opportunities for stakeholders to assess the progress and quality of the project deliverables.
* **Audits**: Ensures compliance with project management practices, quality standards, and regulatory requirements.
* **Configuration Management**: Manages changes to the software and documentation to maintain consistency and traceability.
* **System Engineering**: Ensures that the software integrates seamlessly with other system components.
* **Assessment Processes**: Evaluates the effectiveness of the quality assurance processes and identifies areas for improvement.

**Process Improvement Procedures**

**Periodic Assessments**

* Conduct regular assessments of the project to identify areas for improvement.
  + **Frequency**: Quarterly
  + **Responsible Entity**: Project Manager, Quality Assurance Team

**Root Cause Analysis**

* Perform root cause analysis of recurring problems to identify underlying issues and potential process improvements.
  + **Responsible Entity**: Quality Assurance Team

**Implementation of Improvement Plans**

* Develop and implement improvement plans based on the findings from periodic assessments and root cause analysis.
  + **Responsible Entity**: Project Manager, Quality Assurance Team
  + **Review and Approval**: Senior Management

**Monitoring and Evaluation**

* Monitor the implementation of improvement plans and evaluate their effectiveness.
  + **Responsible Entity**: Quality Assurance Team
  + **Frequency**: Monthly

**Points of Contact**

* **Project Manager**: Responsible for overseeing the project and ensuring adherence to the quality assurance and process improvement plans.
* **Quality Assurance Team**: Responsible for conducting reviews, audits, assessments, and implementing improvement plans.

**Relationship with Other Processes**

* **Problem Resolution Plan**: The process improvement plan is closely related to the problem resolution plan, as root cause analysis of recurring problems can lead to process improvements.
* **Project Management**: Ensures that the quality assurance and process improvement plans are integrated into the overall project management plan.
* **Organizational Process Improvement Initiatives**: Identifies processes that can be improved at the organizational level for long-term benefits.

**4.3.1.1. Documentation**

**Non-Deliverable Work Products**

* **Requirements Specifications**: Detailed descriptions of the software requirements.
* **Design Documentation**: Comprehensive design documents including diagrams and descriptions.
* **Traceability Matrices**: Documents tracing requirements to design and implementation.
* **Test Plans**: Detailed plans for testing the software.
* **Meeting Minutes**: Records of project meetings.
* **Review Reports**: Documents summarizing the findings of various reviews.

**Deliverable Work Products**

* **Source Code**: The actual code written for the project.
* **Object Code**: Compiled code ready for execution.
* **User’s Manual**: Documentation for end-users on how to use the software.
* **On-line Help System**: Integrated help system within the software.
* **Regression Test Suite**: Collection of test cases for regression testing.
* **Configuration Library and Tool**: Tools and libraries for configuration management.
* **Principles of Operation**: Documentation detailing the operation principles of the software.
* **Maintenance Guide**: Guide for maintaining the software post-deployment.

**Documentation Plan Details (Make separate table?)**

| **Document** | **Template/Standard** | **Prepared By** | **Reviewed By** | **Due Date (Review Copy)** | **Due Date (Baseline Version)** | **Distribution List** |
| --- | --- | --- | --- | --- | --- | --- |
| Requirements Specifications | IEEE 830 | Requirements Analyst | Project Manager, QA Team | 2025-03-15 | 2025-03-22 | Project Team, Stakeholders |
| Design Documentation | IEEE 1016 | Software Architect | Senior Developer, QA Team | 2025-04-01 | 2025-04-08 | Project Team, Stakeholders |
| Traceability Matrices | Custom | QA Team | Project Manager | 2025-04-15 | 2025-04-22 | Project Team |
| Test Plans | IEEE 829 | Test Engineer | QA Team | 2025-05-01 | 2025-05-08 | Project Team |
| Meeting Minutes | Custom | Project Manager | QA Team | Ongoing | N/A | Project Team |
| Review Reports | Custom | QA Team | Project Manager | Ongoing | N/A | Project Team, Stakeholders |
| Source Code | Coding Standards Document | Development Team | Peer Reviews, QA Team | Ongoing | N/A | Development Team, QA Team |
| Object Code | N/A | Development Team | QA Team | Ongoing | N/A | Development Team, QA Team |
| User’s Manual | IEEE 1063 | Technical Writer | QA Team, Project Manager | 2025-06-01 | 2025-06-08 | End-users, Stakeholders |
| On-line Help System | Custom | Technical Writer | QA Team | 2025-06-15 | 2025-06-22 | End-users |
| Regression Test Suite | Custom | Test Engineer | QA Team | Ongoing | N/A | Development Team, QA Team |
| Configuration Library and Tool | IEEE 828 | Configuration Manager | QA Team | 2025-05-15 | 2025-05-22 | Development Team, QA Team |
| Principles of Operation | Custom | Technical Writer | QA Team | 2025-06-01 | 2025-06-08 | Development Team, Stakeholders |
| Maintenance Guide | IEEE 1219 | Maintenance Engineer | QA Team, Project Manager | 2025-06-15 | 2025-06-22 |  |

**4.3.1.2. Standards**

**Coding Standards**

* **Purpose**: Define the coding conventions, styles, and best practices to be followed by all developers.
* **Reference**: IEEE 730, Standard for Software Quality Assurance Plans
* **Details**: Ensure consistency, readability, and maintainability of the codebase.

**Data Naming Standards**

* **Purpose**: Establish a consistent naming convention for database tables, columns, variables, and other data elements.
* **Reference**: IEEE 830, Recommended Practice for Software Requirement Specification
* **Details**: Enhance clarity and reduce ambiguity in data references.

**User Interface Standards**

* **Purpose**: Provide guidelines for designing user interfaces that are user-friendly, accessible, and consistent across the application.
* **Reference**: IEEE 1063, Standard for Software User Documentation
* **Details**: Ensure adherence to usability principles and industry standards.

**Static Report Format Standards**

* **Purpose**: Define the format, structure, and presentation of static reports generated by the system.
* **Reference**: IEEE 829, Standard for Software Test Documentation
* **Details**: Ensure reports are clear, concise, and informative.

**Development Tool Standards**

* **Purpose**: Specify the development of tools, environments, and frameworks to be used in the project.
* **Reference**: IEEE 1058.1, Standard for Software Project Management Plans
* **Details**: Ensure compatibility, efficiency, and productivity in the development process.

**Documentation Standards**

* **Purpose**: Outline the requirements for project documentation, including format, content, and maintenance.
* **Reference**: IEEE 1063, Standard for Software User Documentation
* **Details**: Ensure comprehensive and up-to-date documentation for all project artifacts.

**Metrics**

* **Purpose**: Define the metrics to be used for measuring the quality and performance of the software.
* **Reference**: IEEE 1045, Standard for Software Productivity Metrics
* **Details**: Include metrics for code quality, defect density, test coverage, and other relevant aspects.

**IEEE Standards**

* **Purpose**: Reference for relevant IEEE standards applicable to the project.
* **Details**: Ensure compliance with industry-recognized standards for software development and quality assurance.
  + IEEE 730, Standard for Software Quality Assurance Plans
  + IEEE 828, Standard for Software Configuration Management Plans
  + IEEE 829, Standard for Software Test Documentation
  + IEEE 830, Recommended Practice for Software Requirement Specification
  + IEEE 1008, Standard for Software Unit Testing
  + IEEE 1012, Standard for Software Verification and Validation Plans
  + IEEE 1016, Guide to Software Design Description
  + IEEE 1028, Standard for Software Review and Audit
  + IEEE 1042, Guide to Software Configuration Management Plans
  + IEEE 1044, Standard Classification for Software Anomalies
  + IEEE 1045, Standard for Software Productivity Metrics
  + IEEE 1058.1, Standard for Software Project Management Plans
  + IEEE 1059, Guide for Software Verification and Validation Plans
  + IEEE 1061, Standard for a Software Quality Metrics Methodology
  + IEEE 1063, Standard for Software User Documentation
  + IEEE 1074, Standard for Developing SDLC Processes
  + IEEE 1219, Standard for Software Maintenance
  + IEEE 1233, Guide to Developing System Requirement Specifications

**IRMC Policies, Standards, Checklists**

Additional information in CheckLists.EPA

* **Purpose**: Incorporate policies, standards, and checklists provided by the IRMC.
* **Details**: Ensure adherence to organizational guidelines and best practices.
  + IRMC Project Proposal Checklist
  + ETS Project Status Report
  + State of North Carolina SDLC Model (Method/1)

**4.3.1.3. Reviews and Audits**

**Review and Audit Types**

| **What** | **Type of Review/Audit** | **Who** | **How Often** | **Where is Report** |
| --- | --- | --- | --- | --- |
| Software Requirements Review | Requirements Review | Requirements Analyst, Project Manager, QA Team, Stakeholders | Once, at the end of the requirements phase | Project Documentation Repository |
| System Architecture Review | Architecture Review | Software Architect, Senior Developer, Project Manager, QA Team | Once, at the end of the architecture design phase | Project Documentation Repository |
| Technical Design Review | Design Review | Software Architect, Senior Developer, Project Manager, QA Team | Once, at the end of the design phase | Project Documentation Repository |
| Software Verification and Validation Plan Review | V&V Plan Review | Test Engineer, QA Team, Project Manager | Once, at the end of the V&V planning phase | Project Documentation Repository |
| Software Configuration Management Plan Review | Configuration Management Plan Review | Configuration Manager, QA Team, Project Manager | Once, at the end of the configuration management planning phase | Project Documentation Repository |
| Implementation Plan Review | Implementation Review | Development Team, QA Team, Project Manager | Once, at the end of the implementation planning phase | Project Documentation Repository |
| Post-implementation Maintenance Plan Review | Maintenance Plan Review | Maintenance Engineer, QA Team, Project Manager | Once, at the conclusion of the project | Project Documentation Repository |
| Independent Auditor Reviews | Management Review | Independent Auditor (Third-party QA Reviewer) | Quarterly | Management Review Reports Repository |
| Peer Reviews | Peer Review | Development Team | Bi-weekly | Project Documentation Repository |
| Inspections | Inspection | QA Team | Monthly | Inspection Reports Repository |
| Walk-throughs | Walk-through | Development Team, Stakeholders | As needed, typically at major milestones | Project Documentation Repository |
| Prototype Reviews | Prototype Review | Development Team, QA Team, Stakeholders | At the end of the prototyping phase | Project Documentation Repository |

**External Agencies**

The plan should also list the external agencies that approve or regulate any product of the project. These agencies include:

* **Regulatory Body 1**: Approves software compliance with industry standards.
* **Regulatory Body 2**: Oversees data privacy and security measures.
* **Regulatory Body 3**: Provides certification for software quality and reliability.

**IEEE Standard for Software Reviews and Audits**

The purpose of this standard is to define systematic reviews and audits applicable to software acquisition, supply, development, operation, and maintenance. This standard describes how to carry out a review. Other standards or local management define the context within which a review is performed, and the use made of the results of the review. Software reviews can be used in support of the objectives of project management, system engineering (for example, functional allocation between hardware and software), verification and validation, configuration management, quality assurance, and auditing. Different types of reviews reflect differences in the goals of each review type. Systematic reviews are described by their defined procedures, scope, and objectives.

**Review Procedures**

1. **Planning**: Define the scope, objectives, and participants of the review. Schedule the review and prepare the necessary materials.
2. **Preparation**: Participants review the materials individually and identify issues or questions.
3. **Meeting**: Conduct the review meeting, discuss the identified issues, and document the findings.
4. **Follow-up**: Address the identified issues, implement corrective actions, and verify their resolution.
5. **Reporting**: Document the review findings, actions taken, and outcomes. Distribute the report to relevant stakeholders.

**Review Objectives**

* **Project Management**: Ensure the project progresses as planned and within budget.
* **System Engineering**: Validate the allocation of functions between hardware and software.
* **Verification and Validation**: Confirm that the software meets the specified requirements and performs as expected.
* **Configuration Management**: Ensure consistency and traceability of software artifacts.
* **Quality Assurance**: Maintain high standards of software quality throughout the development lifecycle.
* **Auditing**: Verify compliance with standards, guidelines, and contractual obligations.

**Systematic Reviews**

Systematic reviews are characterized by their defined procedures, scope, and objectives. They provide a structured approach to evaluating software artifacts and processes, ensuring thoroughness and consistency.

* Procedures: Clearly defined steps for conducting the review, including planning, preparation, meeting, follow-up, and reporting.
* Scope: The specific aspects of the software or process being reviewed.
* Objectives: The goals and desired outcomes of the review.

By adhering to these standards, the EPA project ensures that all reviews and audits are conducted systematically, providing valuable insights and maintaining high-quality deliverables throughout the project lifecycle.

**Documentation Standards**

* Outline the requirements for project documentation, including format, content, and maintenance.
* Ensure comprehensive and up-to-date documentation for all project artifacts.

**Metrics**

* Define the metrics to be used for measuring the quality and performance of the software.
* Include metrics for code quality, defect density, test coverage, and other relevant aspects.

**IEEE Standards (Refer to Appendix)**

* Reference relevant IEEE standards applicable to the project.
* Ensure compliance with industry-recognized standards for software development and quality assurance.

**IRMC Policies, Standards, Checklists**

* Incorporate policies, standards, and checklists provided by the IRMC.
* Ensure adherence to organizational guidelines and best practices.

**2.REVIEWS AND AUDITS**

This section describes how deliverable reviews, project management audits, and third-party independent reviews will be employed to improve the quality of the system.

**Deliverable Reviews**

**Software Requirements Review**

* Ensure the adequacy of the requirements stated in the business functional requirements.

**System Architecture Review**

* Ensure application compliance with the State Technical Architecture.

**Technical Design Review**

* Evaluate the technical adequacy of the software design and its acceptability to satisfy business functional requirements.

**Software Verification and Validation Plan Review**

* Evaluate the adequacy and completeness of the verification and validation methods defined in the Software Verification and Validation Plan.

**Software Configuration Management Plan Review**

* Evaluate the adequacy and completeness of the configuration management methods defined in the Software Configuration Management Plan.

**Implementation Plan Review**

* Verify that the software and the documentation are consistent. In-process audits may include:
  + Code versus design documentation,
  + Interface specifications,
  + Design implementation versus functional requirements,
  + Functional requirements versus test specifications.

**Post-implementation Maintenance Plan Review**

* Assess the development activities implemented and provide recommendations for appropriate action.

**Management Reviews**

This section explains the role of the independent auditor (third-party independent QA reviewer) in the review of project management practices and process artifacts. It describes the schedule and focus of the audits and specifies the format of the audit reports. It also includes a focus on measuring tracking and progress towards this quality assurance plan.

**Schedule**

* Define the timeline and frequency of management reviews.

**Third-Party Independent Review Reporting**

* Describe the format and process and identify who will receive the reports.

**Product Reviews**

This section explains the role of the project’s quality assurance specialist in the review of design and development processes. It describes the schedule and focus of audits and specifies the format of the audit reports. Product reviews would include:

**Peer Reviews**

* Conduct peer reviews to ensure code quality and adherence to standards.

**Inspections**

* Perform inspections to identify defects and improvement areas in the code and documentation.

**Walkthroughs**

* Facilitate walk-throughs to review the design and implementation with stakeholders.

**Prototype Reviews**

* Review prototypes to validate design concepts and gather feedback.

**Schedule**

* Define the timeline and frequency of product reviews.

**Review Reporting**

* Describe the format and process and identify who will receive the reports.

**3.PROBLEM REPORTING AND CORRECTIVE ACTION**

This section defines the process for tracking and correcting problems identified within the audits. It describes the practices and procedures to be followed for reporting, tracking, and resolving problems identified in both software items and the software development and maintenance process. It also states the specific organizational responsibilities concerned with problem resolution.

**Problem Reporting and Corrective Action Procedures**

* **Documentation of Problems**: Ensure that problems are documented, corrected, and used for process improvement.
* **Assessment of Validity**: Assure that problem reports are assessed for their validity.
* **Implementation of Corrective Actions**: Ensure reported problems and their associated corrective actions are implemented in accordance with customer-approved solutions.
* **Feedback**: Provide feedback to the developer and the user on problem status.
* **Data Collection**: Provide data for measuring and predicting software quality and reliability.

(For more information about software problems, refer to IEEE Standard 1044, Standard Classification of Software Anomalies.)

**Appendix - Standards**

This appendix provides the complete text of the standards developed for the Emotional Pedagogical Agent (EPA) project.

**IEEE Standards**

* **IEEE 730**: Standard for Software Quality Assurance Plans
  + This standard provides guidelines for establishing and implementing a software quality assurance program, ensuring that the software meets specified requirements.
* **IEEE 828**: Standard for Software Configuration Management Plans
  + This standard outlines the processes and procedures for managing changes in software configuration, ensuring consistency and traceability.
* **IEEE 829**: Standard for Software Test Documentation
  + This standard specifies the format and content of test documentation, ensuring comprehensive and repeatable testing processes.
* **IEEE 830**: Recommended Practice for Software Requirement Specification
  + This standard provides guidelines for writing clear and complete software requirements specifications.
* **IEEE 1008**: Standard for Software Unit Testing
  + This standard describes the procedures and guidelines for unit testing of software components to ensure they function correctly.
* **IEEE 1012**: Standard for Software Verification and Validation Plans
  + This standard outlines the processes for verifying and validating software to ensure it meets requirements and performs as expected.
* **IEEE 1016**: Guide to Software Design Description
  + This standard provides guidelines for documenting software design, ensuring clarity and comprehensibility.
* **IEEE 1028**: Standard for Software Review and Audit
  + This standard defines the processes for conducting systematic reviews and audits of software projects to ensure compliance and quality.
* **IEEE 1042**: Guide to Software Configuration Management Plans
  + This standard offers guidance on creating and implementing software configuration management plans.
* **IEEE 1044**: Standard Classification for Software Anomalies
  + This standard classifies software anomalies to aid in tracking, managing, and resolving defects.
* **IEEE 1045**: Standard for Software Productivity Metrics
  + This standard provides guidelines for measuring software productivity and effectiveness.
* **IEEE 1058.1**: Standard for Software Project Management Plans
  + This standard outlines the requirements for creating comprehensive software project management plans.
* **IEEE 1059**: Guide for Software Verification and Validation Plans
  + This standard offers guidance on developing effective verification and validation plans for software projects.
* **IEEE 1061**: Standard for a Software Quality Metrics Methodology
  + This standard provides a methodology for defining and using software quality metrics.
* **IEEE 1063**: Standard for Software User Documentation
  + This standard specifies the requirements for creating user documentation that is clear, comprehensive, and user-friendly.
* **IEEE 1074**: Standard for Developing SDLC Processes
  + This standard provides guidelines for developing effective software development lifecycle (SDLC) processes.
* **IEEE 1219**: Standard for Software Maintenance
  + This standard outlines the processes and procedures for maintaining software post-deployment.
* **IEEE 1233**: Guide to Developing System Requirement Specifications
  + This standard offers guidance on creating complete and comprehensible system requirements specifications.

**IRMC Policies, Standards, Checklists**

* **IRMC Project Proposal Checklist**
  + A checklist used to ensure that all necessary elements are included in the project proposal.
* **ETS Project Status Report**
  + A report format used for tracking and reporting the status of the project.
* **State of North Carolina SDLC Model (Method/1)**
  + A software development lifecycle model used as a guideline for managing software projects.

**4.3.1.4. Test**

**Scope**

The verification and validation (V&V) activities will ensure that the EPA project meets its specified requirements and performs as expected. These activities will be conducted throughout the project lifecycle, from requirements analysis to system deployment.

**Tools and Techniques**

* **Traceability**: Ensure all requirements are traced to design, implementation, and testing.
* **Milestone Reviews**: Conduct reviews at key project milestones to assess progress and compliance with requirements.
* **Progress Reviews**: Regularly review project progress to identify and address issues early.
* **Peer Reviews**: Conduct peer reviews of code, design documents, and other project artifacts.
* **Prototyping**: Develop prototypes to validate design concepts and gather feedback.
* **Simulation and Modeling**: Use simulation and modeling to verify system behavior and performance.

**Validation Techniques**

* **Testing**: Perform various types of testing, including unit testing, integration testing, system testing, and acceptance testing.
* **Demonstration**: Demonstrate the system to stakeholders to validate its functionality and performance.
* **Analysis**: Analyze system performance, reliability, and other metrics to validate its effectiveness.
* **Inspection**: Conduct inspections of code, design documents, and other artifacts to identify defects and improvement areas.

**Automated Tools**

* **Automated Testing Tools**: Implement automated testing tools to streamline the testing process and ensure comprehensive test coverage.
* **Continuous Integration Tools**: Use continuous integration tools to automate the build, test, and deployment processes.
* **Static Code Analysis Tools**: Employ static code analysis tools to identify code quality issues and potential defects early in the development process.

**Responsibilities**

* **Development Team**: Responsible for implementing the software and conducting initial testing.
* **Quality Assurance Team**: Responsible for conducting independent verification and validation activities.
* **Project Manager**: Responsible for overseeing V&V activities and ensuring they are integrated into the overall project plan.
* **Stakeholders**: Involved in validation activities such as demonstrations and acceptance testing.

**Organizational Relationships**

* **Development Activities**: Conducted by the Development Team, with support from the Quality Assurance Team.
* **Verification and Validation Activities**: Conducted by the Quality Assurance Team, with oversight from the Project Manager.

Additional information in Work paper.EPA.

**4.3.1.5. Problem reporting and corrective action**

**Purpose**

The purposes of problem reporting and corrective action systems are to:

* Assure that problems are documented, corrected, and used for process improvement.
* Assure that problem reports are assessed for their validity.
* Ensure reported problems and their associated corrective actions are implemented in accordance with customer-approved solutions.
* Provide feedback to the developer and the user on problem status.
* Provide data for measuring and predicting software quality and reliability.

(For more information about software problems, refer to IEEE Standard 1044, Standard Classification of Software Anomalies.)

**Practices and Procedures**

**Problem Reporting**

* **Standardized Problem Report Form**: Use a standardized form to document all problems.
* **Submission Process**: Problems can be reported by any team member or stakeholder using the standardized form.
* **Logging**: All reported problems are logged in a centralized problem tracking system.

**Problem Analysis**

* **Initial Assessment**: The Quality Assurance Team reviews the problem report to assess its validity and impact.
* **Root Cause Analysis**: The Development Team conducts a root cause analysis to identify the underlying issue.

**Prioritization**

* **Severity Levels**: Problems are categorized based on their severity (e.g., critical, major, minor).
* **Impact Assessment**: Assess the impact of the problem on the project timeline, budget, and deliverables.
* **Prioritization Criteria**: Problems are prioritized based on severity, impact, and likelihood of occurrence.

**Problem Resolution**

* **Assignment**: Problems are assigned to the appropriate team members for resolution.
* **Resolution Plan**: Develop a plan to address and resolve the problem.
* **Implementation**: Implement the corrective actions as per the resolution plan.

**Verification and Validation**

* **Verification**: The Quality Assurance Team verifies that the problem has been resolved as per the resolution plan.
* **Validation**: Validate that the corrective action has effectively addressed the problem without introducing new issues.

**Roles and Responsibilities**

* **Development Team**: Responsible for analyzing and resolving reported problems.
* **Configuration Management Team**: Manages changes to the software and documentation.
* **Change Control Board (CCB)**: Reviews and approvals of changes to the software.
* **Quality Assurance Team**: Verifies and validates problem resolutions.

**Reporting and Tracking**

* **Problem Reports**: Documented using a standardized problem report form.
* **Tracking System**: All problem reports and resolutions are tracked in a centralized problem tracking system.
* **Effort Reporting**: Effort devoted to problem reporting, analysis, and resolution is separately reported to track rework and process improvement.

**Feedback and Improvement**

* **Feedback Loop**: Provide feedback to the developer and the user on the status of reported problems.
* **Process Improvement**: Use data from problem reports to identify areas for process improvement and implement changes to prevent recurrence of similar issues.

By adhering to this plan, the EPA project ensures that problem reporting, and corrective actions are systematically conducted, providing valuable insights and maintaining high-quality deliverables throughout the project lifecycle.

**4.3.1.6. Tools, techniques, and methodologies**

**Tools**

* **Development Tools**: Integrated Development Environments (IDEs) such as Visual Studio Code, IntelliJ IDEA.
* **Version Control**: Git and GitHub for version control and collaboration.
* **Automated Testing**: Tools like Selenium, JUnit, and pytest.
* **Continuous Integration/Continuous Deployment (CI/CD)**: Jenkins, GitHub Actions.
* **Project Management**: Jira, Trello for task and project management.

**Techniques**

* **Agile Methodologies**: Scrum, Kanban for iterative and incremental development.
* **Prototyping**: Creating prototypes to validate design concepts and gather feedback.
* **Simulation and Modeling**: Using simulation and modeling to verify system behavior and performance.
* **Peer Reviews**: Conducting peer reviews to ensure code quality and adherence to standards.

**Methodologies**

* **Software Development Lifecycle (SDLC)**: Following a structured approach to software development.
* **Test-Driven Development (TDD)**: Writing tests before developing the actual code.
* **Behavior-Driven Development (BDD)**: Using BDD to write test cases in a natural language.
* **Configuration Management**: Implementing configuration management to maintain consistency and traceability.

**4.3.1.7**. **Code control**

**Code Control Procedures**

* **Version Control System (VCS)**: Use Git for maintaining version control.
* **Branching Strategy**: Adopt a branching strategy such as Git Flow to manage feature development, bug fixes, and releases.
* **Code Reviews**: Conduct code reviews for all changes to ensure quality and adherence to coding standards.
* **Commit Messages**: Follow a standard format for commit messages to ensure clarity and traceability.
* **Pull Requests**: Use pull requests to review and merge code changes into the main branch.

### 4.3.2. Software Testing

* Scope
  + The verification and validation plan for the development of an emotional pedagogical agent in an e-learning environment outlines the scope, tools, techniques, and responsibilities for verification and validation activities. This plan ensures that the software project meets its requirements and performs as expected in the intended environment.

Tools and Techniques:

* Verification Techniques: Verification planning includes the following techniques to ensure that the software meets its specified requirements
  + Traceability: Establishing traceability matrices to ensure that all requirements are covered by design elements and test cases.

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement title | Requirement description | Test Case ID |
| 001 | Emotional Expression | The agent must be able to display a range of emotions such as happiness, sadness, anger, surprise, and neutrality. | Test\_EPA\_001 |
| 022 | Usability | The agent must be user-friendly and easy to interact with. | Test\_EPA\_022 |
| 102 | Integration | The agent must be compatible with the existing e-learning platform and easily integrable. | Test\_EPA\_102 |

* + Milestone Reviews: Conducting milestone reviews at critical points in the project to assess progress and alignment with requirements. Main milestones to review – Project Plan Approved, Requirements Specification Completed, Design Documents and Prototypes Completed, Basic Features Developed, Advanced Features Developed, Integration and Testing Completed, User Training and Documentation Completed, Project Go-Live.
  + Progress Reviews: Regular progress reviews to monitor the status of development activities and identify any deviation from the plan that is set in Project Management Tool.
  + Peer Reviews: Performing peer reviews of design documents, code, and test cases to identify defects early in the development process.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Title | Preconditions | Step actions | Expected Results | Priority | Requirements ID |
| Test\_EPA\_001\_01 | Emotional Expression (Display Happiness) | 1. The agent is initialized and operational. 2. The agent interface is accessible to the tester.  3. The agent has predefined responses for various emotions (happiness, sadness, anger, surprise, and neutrality) | Trigger a scenario where the agent should display happiness by receiving compliment. | The agent displays a happy expression by smiling. The agent responds with a positive message. | High | 001 |
| Test\_EPA\_001\_02 | Emotional Expression (Display Sadness) | 1. The agent is initialized and operational. 2. The agent interface is accessible to the tester.  3. The agent has predefined responses for various emotions (happiness, sadness, anger, surprise, and neutrality) | Trigger a scenario where the agent should display sadness by hearing bad news. | The agent displays a sad expression by frowning. The agent responds with a message indicating sadness. | High | 001 |
| Test\_EPA\_001\_03 | Emotional Expression (Display Anger) | 1. The agent is initialized and operational. 2. The agent interface is accessible to the tester.  3. The agent has predefined responses for various emotions (happiness, sadness, anger, surprise, and neutrality) | Trigger a scenario where the agent should display anger by receiving an offensive statement. | The agent displays an angry expression. The agent responds with a message indicating anger. | Medium | 001 |
| Test\_EPA\_001\_04 | Emotional Expression (Display Surprise) | 1. The agent is initialized and operational. 2. The agent interface is accessible to the tester.  3. The agent has predefined responses for various emotions (happiness, sadness, anger, surprise, and neutrality) | Trigger a scenario where the agent should display surprises by receiving unexpected information. | The agent displays a surprised expression. The agent responds with a message indicating surprise. | Medium | 001 |

* + Prototyping: Developing prototypes to validate design concepts and gather feedback from stakeholders.
  + Simulation and Modeling: Using simulation and modeling techniques to validate the behavior of the pedagogical agent in various scenarios like Virtual Classrooms where simulated classroom settings where agents interact with virtual learners and Interactive Scenarios that mimic real-world learning situations to test agent responses.
* Validation Techniques
  + Testing: Conducting various levels of testing (unit, integration, system, and acceptance testing) to validate the functionality and performance of the pedagogical agent.
  + Demonstration: Demonstrating the software to stakeholders to validate that it meets their needs and expectations.
  + Analysis: Performing analysis of test result, user feedback, and performance metrics to validate the effectiveness of the pedagogical agent.
  + Inspection: Conducting inspections of design documents, code, and test cases to identify defects and ensure compliance with requirements.
* Automated Tools: The following automated tools will be used in the verification and validation activities:
  + Requirement Management Tools: Tools such as Jira or Azure DevOps to manage requirements and traceability.
  + Code Review Tools: Tools such as GitHub or Bitbucket for conducting peer reviews and managing code quality.
  + Testing Tools: Automated testing tools such as Selenium, JUnit, and LoadRunner for functional and performance testing.
  + Simulation Tools: Tools such as MATLAB or Simulink for simulation and modeling of the pedagogical agent’s behavior.
* Responsibilities: The following roles and responsibilities are defined for the verification and validation activities.
  + Project Manager: Responsible for overall project management, including the planning and coordination of verification and validation activities.
  + Verification Lead: Responsible for planning and executing verification activities, including milestone reviews, progress reviews, and peer reviews.
  + Validation Lead: Responsible for planning and executing validation activities, including testing, demonstration, analysis, and inspection.
  + Development Team: Responsible for developing the software and supporting verification and validation activities by providing necessary documentation and access to the software.
  + Quality Assurance Team: Responsible for ensuring the quality of the software through independent verification and validation activities.
* Organizational relationships and independence: To ensure the independence of verification and validation activities from the development team and reports directly to the project manager.
  + The QA team operates independently from the development team and reports directly to the project manager.
  + Verification and validation leads are assigned from QA team and are not involved in the development activities.
  + Peer reviews are conducted by team members who are not directly involved in the development of the reviewed component.

*Refers to* ***3.4. Monitoring and controlling mechanisms***

# 5. Work packages, timetable and budget

## 5.1. Work packages, and budget

This section outlines the various work packages and their associated budgets for the EPA project. The work packages are decomposed to a level that exposes all project risk factors and allows accurate estimation of resource requirements and schedule duration for each work activity.

**Work Packages and Budget**

| **Work Package** | **Description** | **Budget (euro)** | **Deliverables** |
| --- | --- | --- | --- |
| **WP1** | Project Initiation and Planning | 3,300 | Project charter, project plan |
| **WP2** | Requirements Analysis | 2,600 | Requirements specification document |
| **WP3** | Design and Prototyping | 7,200 | Design documents, interactive prototypes |
| **WP4** | Basic Feature Development Phase | 9,900 | Source code for basic features |
| **WP5** | Advanced Feature Development Phase | 9,000 | Source code for advanced features |
| **WP6** | Integration and Testing | 9,400 | Integration report, testing report |
| **WP7** | User Training and Documentation | 4,000 | User manuals, training materials |
| **WP8** | Project Deployment and Go-Live | 2,600 | Deployment package, final project report |
| **Contingency** | Allocated for unexpected expenses and risks | 2,000 | N/A |

**Total Budget**: 50,000 euro

*Refers to* **1.1. Project summary, 3.4. Monitoring and controlling mechanisms.**

## 5.2. Dependencies

This section outlines the dependencies that may impact the successful completion of the EPA project. Dependence includes factors that the project relies on to progress as planned.

**Technical Dependencies**

* **AI Technologies**: Availability and compatibility of machine learning, natural language processing (NLP), and affective computing technologies.
* **E-learning Platforms**: Compatibility of the EPA with existing e-learning platforms without requiring significant modifications.

**Resource Dependencies**

* **Personnel**: Availability of skilled personnel in AI development, NLP, affective computing, software engineering, and project management.
* **Budget**: Availability of the allocated budget of 50,000 euro for the project duration.

**External Dependencies**

* **Regulatory Compliance**: Adherence to GDPR and AI usage regulations.
* **Educational Standards**: Stability of educational regulations and standards during the project timeline.

**Collaboration Dependencies**

* **E-learning Platform Providers**: Collaboration with e-learning platform providers to ensure seamless integration of the EPA.
* **Stakeholders**: Timely feedback and approvals from project stakeholders.

**Infrastructure Dependencies**

* **Equipment and Technology**: Availability of necessary equipment and technology for developing and testing the EPA.
* **Network and Connectivity**: Reliable network and connectivity for integrating the EPA with e-learning platforms.

Addressing these dependencies is critical to ensure the project progresses smoothly and achieves its objectives within the specified timeframe and budget.

**5.3. Resource requirements**

This section outlines the resource requirements for the EPA project, including personnel, equipment, technology, and other resources needed to successfully complete the project.

**Personnel Requirements**

* **AI Developers**: Skilled in machine learning and AI technologies.
* **NLP Specialists**: Expertise in natural language processing.
* **Affective Computing Experts**: Knowledge of affective computing and emotion recognition.
* **Software Engineers**: Proficient in software development and integration.
* **Project Managers**: Experienced in managing software development projects.
* **QA/Testers**: Skilled in quality assurance and testing methodologies.
* **Technical Writers**: Proficient in creating user manuals and training materials.

**Equipment and Technology Requirements**

* **Development Workstations**: High-performance computers for development and testing.
* **Software Tools**: Development tools, libraries, and frameworks for AI, NLP, and affective computing.
* **Testing Environments**: Simulated environments for testing the EPA's functionalities.
* **Network Infrastructure**: Reliable network infrastructure for integration and testing.

**Training Requirements**

* **Technical Training**: Training for personnel on AI, NLP, and affective computing technologies.
* **Project Management Training**: Training for project managers on project management best practices.
* **User Training**: Training sessions for end-users on how to use the EPA effectively.

**Other Resource Requirements**

* **Office Space**: Workspace for the project team.
* **Administrative Support**: Support for project documentation and coordination.
* **Transportation**: Travel arrangements for team members if required.

These resources are essential to ensure the successful execution of the EPA project and to achieve the project objectives within the specified timeframe and budget.

## 5.4. Timetable

This section provides a detailed timetable for the EPA project, outlining the schedule for each work package and the overall project timeline.

* The project must be completed within **12 months**.

**Project Timetable**

| **Work Package** | **Start Date** | **End Date** | **Duration** |
| --- | --- | --- | --- |
| **WP1**: Project Initiation and Planning | 2025-03-01 | 2025-03-31 | 1 month |
| **WP2**: Requirements Analysis | 2025-04-01 | 2025-04-30 | 1 month |
| **WP3**: Design and Prototyping | 2025-05-05 | 2025-06-30 | 2 months |
| **WP4**: Basic Feature Development Phase | 2025-07-01 | 2025-08-31 | 2 months |
| **WP5**: Advanced Feature Development Phase | 2025-09-01 | 2025-10-31 | 2 months |
| **WP6**: Integration and Testing | 2025-11-01 | 2025-12-30 | 2 months |
| **WP7**: User Training and Documentation | 2026-01-02 | 2026-01-31 | 1 month |
| **WP8**: Project Deployment and Go-Live | 2026-02-01 | 2026-02-28 | 1 month |

**Milestones:**

* **Project Plan Approved**: 2025-03-31
* **Requirements Specification Completed**: 2025-04-30
* **Design Documents and Prototypes Completed**: 2025-06-30
* **Basic Features Developed**: 2025-08-31
* **Advanced Features Developed**: 2025-10-31
* **Integration and Testing Completed**: 2025-12-30
* **User Training and Documentation Completed**: 2026-01-31
* **Project Go-Live**: 2026-02-28

**Gantt Chart**

A detailed Gantt chart will be used to illustrate the project schedule, showing the start and end dates for each work package and the dependencies between them. The Gantt chart will be updated regularly to reflect the project's progress and any adjustments to the schedule.

The timetable ensures that the EPA project progresses in a structured and timely manner, with clear milestones and deadlines for each phase. Regular monitoring and updates will help keep the project on track and achieve the project objectives within the specified timeframe.

# ANNEXES

* 001.SCD.EPA\_1.0
* Business Model Canvas.EPA
* QFD.EPA
* Work paper.EPA (1, 22, 24)
* Check Lists.EPA

1. Salaries taken from algas.lv [↑](#footnote-ref-1)
2. Data protection regulation- <https://gdpr.eu/> [↑](#footnote-ref-2)
3. Web Content Accessibility Guidelines (WCAG) 2.2- <https://www.w3.org/TR/WCAG22/> [↑](#footnote-ref-3)