2. Requirements, Constraints, And Standards

2.1 Requirements & Constraints

To make sure that our project meets the needs of our users, we have defined many requirements of various types. Creating these requirements ensures that we cover all aspects necessary to create a robust, user-friendly, and efficient learning platform.

Our functional requirements include the need to implement homeworks where document-format questions are coded into the Prairie Learn course. Additionally, all but a handful of questions should be able to be autograded, including those involving student-written code segments. We also require the randomization of almost all parameters in questions so that each question can be practiced an unlimited number of times.

For our aesthetic requirements, it is crucial that our project is free of bugs and typos. This ensures a polished and professional appearance, which is essential for maintaining high user satisfaction.

Our user experience requirements focus on creating an engaging and interactive experience. New question types need to be thought of and implemented with a strong emphasis on interactiveness. This is what will set our project apart from others. Questions also should be formatted in a way that is easy for the user to understand and interact with, making the learning process more intuitive and enjoyable.

Lastly, our resource requirements include the implementation of the Virtual/Emulated Cybot interface within our project, allowing students to practice more with embedded programming without having to be in the lab room. Documentation must be written about each aspect of our implementation to support ongoing development for our project and provide clear guidelines. Additionally, we aim to create tutorials for other classes that want to set up their own PrairieLearn server.

2.2 Engineering Standards

2.2.1 Importance

Engineering standards are important because they ensure safety, reliability, and consistency when designing and creating new products and protocols. Engineering standards are defined protocols that can be followed by everyone because they provide a common language across different engineering disciplines, further ensuring any product in any area continues to follow the safety protocols defined by the engineering standards. This is extremely important because everyone uses multiple products on a daily basis, such as driving a car or using a wifi connection. Consumers need to be able to rely on products, and standards help ensure trustworthy engineering practices.

2.2.2 Standards and Descriptions

The first relevant standard we chose was ISO/IEC/IEEE 14764:2022, Software Life Cycle Processes - Maintenance. This standard defines processes for the maintenance of software

throughout its lifecycle. It outlines activities and tasks associated with maintaining software, such as planning, implementing changes, and managing resources. The primary goal of this standard is to guide people to keep maintaining their software which is critical, as it ensures that software remains functional, secure, and up to date, especially as new vulnerabilities or bugs are discovered.

The next relevant standard we chose is ISO/IEC/IEEE 42010:2022, Software, Systems, and Enterprise - Architecture Description. This standard focuses on defining and describing the architecture of systems, software, and enterprises. It sets guidelines for documenting architecture decisions, using viewpoints and models to represent different aspects of the system. Its goal is to provide a structured method for capturing and sharing architectural information, ensuring that all members (or as the document describes them, stakeholders) have a clear understanding of a system's structure and behavior. This helps in making the communication between system design and implementation easier and clearer.

The final standard we chose was ISO/IEC 27001:2022, Information security, cybersecurity and privacy protection. This standard gives a framework for groups to manage the security of their information. It focuses on ensuring the confidentiality, integrity, and availability of our users' information. It also tells us to identify risks to the confidentiality of information and take appropriate actions to ensure the security of it. The intent of this standard is to implement and constantly improve our security system, thereby reducing security risks and boosting the confidence of our users.

2.2.3 Relevance

Software life cycle processes -- Maintenance

As our project has been developed by two teams before us, software maintenance has affected every aspect of our work. We are simultaneously maintaining legacy code, while also creating new code that must remain maintainable for the senior design groups and other Iowa State course developers that come after us. This standard goes in depth on types of maintenance and how problems should be documented. As programmers, we need to document our code and any bugs we find to make it as easy as possible for others to pick up where we leave off.

Software, systems and enterprise -- Architecture description

The architecture design standard is relevant to our project because architecture design is what is used to express the architecture of our project. The architecture of our project helps us to understand the properties of the project we are working on. The architecture descriptions allow us to cooperate and communicate better as we work to integrate all of the architectures of our project. As a team, we want to be thorough in our communication and understand the architectures of our project and we can accomplish that with architecture descriptions. We will make sure to follow the architecture designs that have already been created by Prairie Learn, and we will make sure to create more architectural designs and diagrams that we stick to as the development of our project progresses. We need to follow current architectural designs to ensure that the front-end and back-end of our project will be easy to understand and make changes too. We also want the UI for our project to be intuitive and easy enough for students to understand as they interact with it.

Information security, cybersecurity and privacy protection

The information security standard covers things such as the process for assessing risk, evaluating mitigation effectiveness, security documentation, and improvement. These are things that are certainly relevant to our project, since we will have sensitive information in our application such as student grades and homework answers. It is important to assess the risk our platform has for leakage of such things so that we can come up with solutions to improve the platform's security and prevent issues proactively.

2.2.4 Additional Standards

From everyone on the team, we chose the standards:

- Standard for Configuration Management in Systems and Software Engineering
- ISO/IEC/IEEE International Standard Software engineering Software life cycle processes Maintenance
- ISO/IEC/IEEE International Standard Software and systems engineering --Software testing --Part 1:General concepts
- ISO/IEC/IEEE International Standard Software and systems engineering Software testing -- Part 2: Test processes
- IEEE Standard for Software Verification and Validation

All of the standards we chose are roughly the same, where each standard focuses on one aspect of software and system design and development. We decided to focus on the Maintenance, Architecture, and Security standards as they have the least amount of overlap in the subject matter covered.

2.2.5 MODIFICATIONS AND INCORPORATION

After reading the standards we found, we have a set of modifications we intend to incorporate into our product. One such modification is a security risk assessment of the platform, which is outlined in ISO 27001:2022. A risk assessment allows developers to prevent undesired effects, ensure the intended outcome of the product, and help continuous improvement. Risks are anything that affects the confidentiality, integrity, and availability of the platform, and assessing risk is based on the potential consequences of a risk and the likelihood of that risk happening. We could survey our product and compile a list of risks based on these ideas.

Another modification is an increased emphasis on documentation. IEEE 42010:2022, the standard on architecture description, outlines how we are to document the architecture of the system. Our architecture description will feature system elements, relationships between those elements, the system's relationship with the environment, system behavior, and the principles behind the design. This description will help future developers understand the design of the system and allow for easier improvement.

The IEEE/ISO 14764-2021 standard on maintenance will change our approach to maintenance of our software. We will write code with a focus on readability, adaptability, and scalability rather than just writing code that functions. We will do this through our code conventions and documentation to explain our work. Furthermore, this standard provides terms to classify types of maintenance, such as corrective compared to additive maintenance. Even just being aware of this taxonomy helps us consider the importance and purpose of changes we make to the code base. This approach will help us organize our tasks and understand how our work fits into the system as a whole.