**Question 1:**

1. **Requirement characteristics:**

* **Reliability**:
* The requirements are concise yet clear, and the project may be completed.
* The system must deliver high-quality service and prevent any after-implementation technical or user issues
* **Types and number of requirements:**
* Any last-minute changes to demand will be taken into account. keeping both technical ability and aesthetic appeal in mind.
* The requirements are explicit and unambiguous, and they include both functional and non-functional needs:
  + Functional Requirements: These outline what a system must do, defining its features and functions.
  + Non-Functional Requirements: These outline the performance, dependability, usability, and other qualities that the system must possess in order to carry out its duties.
* **How often the requirements can change:**
* The requirements will be improved in the upcoming weeks or months as they are not yet fully specified.
* **Can the requirements be defined at an early stage:**
* Yes, the business stakeholders have established the high-level vision for the web application.

1. **Development team:**

* **Team size:** 
  + - The team consists of 6 developers and 2 QA.
* **Level of understanding of user requirements by the developers:**
* Due to the fact that Agile is a more team-based model than Waterfall, it is ideal for improving cross-training and interdepartmental cooperation and, as a result, fits the team well. Decisions that have a significant impact on the company's future are still made by the team lead.

1. **User involvement in the project:**

* Users are involved in the product experience and have the ability to recommend improvements based on their needs, therefore we can say that there is a significant amount of user involvement in this project.

1. **Software development methodology:**

* There are several reasons that why I choose **Agile methodologies** used in software development:

- Faster time-to-market: Prioritize delivering working software in short iterations, which allows teams to get feedback from stakeholders and adjust quickly. This can help reduce time-to-market and ensure that the product meets user needs.

- Customer satisfaction: Agile methodologies prioritize delivering value to the customer. By focusing on the customer's needs and feedback, teams can ensure that the product meets or exceeds their expectations.

- Increased quality: Prioritize quality through practices such as automated testing, continuous integration, and continuous delivery. This can help ensure that the software is of high quality and meets user needs.

- Continuous improvement: Agile methodologies encourage continuous improvement by reflecting on past performance and identifying areas for improvement. This helps teams to become more efficient, effective, and productive over time.

- Collaboration: Agile methodologies emphasize collaboration between team members, stakeholders, and customers. This promotes better communication, understanding, and alignment of project goals and expectations.

- Flexibility: Agile methodologies are designed to be flexible and adaptable to changing project requirements. This means that teams can respond quickly to changes in the project scope, customer needs, or market conditions.

* 🡪 **Overall**, Agile methodologies provide a framework for delivering high-quality software that meets the needs of both customers and stakeholders, while promoting collaboration, flexibility, and continuous improvement.

1. **Conclusion**:

* In light of the foregoing evaluation, I feel that the **Agile** model is the most appropriate approach in this situation due to the frequent supply of requirements. Since the requirements system is flexible and subject to change at any time, it is appropriate to deploy the system first rather than developing it in stages.

**Question 2:**

* **Functional Requirements**:
* Allow employees to enroll in upcoming training courses and provide information about course dates, times, and locations.
* Allow managers to track employee performance before and after training courses to measure the impact of training on employee performance.
* Allow employees to provide feedback on training courses and managers to collect feedback through surveys and other mechanisms.
* The system should offer a library of training courses that employees can take to improve their skills and knowledge.
* **Non-Functional Requirements**:
* Performance: After installation, the system should deliver a high-quality service and prevent any foreseeable technical or user issues.
* Security: Information must be safeguarded by the system, and unwanted access must be prevented.

**Question 3:**

* User Story:
* As a manager, I want to be able to track employee performance before and after training courses to measure the impact of training on employee performance, so that I can suggest courses next time to improve skills and knowledge for employee.
* As a employee, I want to be able to enroll in upcoming training courses and provide information about course dates, times, and locations, so that I can manage my courses schedule and deadline.

**Question 4:**





--------------------------------------------------- **Release 1** -----------------------------------------



--------------------------------------------------- **Release 2** -----------------------------------------



---------------------------------------------------- **Release 3** -----------------------------------------



**Question 5:**

* Assumption 1: The course schedule stays the identical during the project.
* Rating: Low impact if wrong, High probability of being wrong.
* Explanation: The curriculum may need to be updated or revised over time, but the impact of incorrect assumptions is not severe enough to significantly affect the overall success of the project. However, given the dynamic nature of remote work and training, it is likely that some changes will be required.
* Assumption 2: Employees enrol in courses primarily online.
* Rating: Low impact if wrong, High probability of being wrong
* Explanation: Online submission methods are the most common, but some employees may prefer other formats such as classrooms or self-study. However, the system must allow registration in all available deployment modes, so the impact of incorrect assumptions will not significantly affect the overall success of the project.
* Assumption 3: Course dates, times, and locations are consistent across projects.
* Rating: High impact of deception, Low probability of deception
* Explanation: Unexpected changes to course dates, times, or locations can cause confusion among staff and affect course participation. While this assumption may be unlikely to be wrong, the consequences of being wrong are severe enough to hinder the success of your project. Therefore, it is important that your system is designed to handle unexpected changes in course schedules and that staff are kept up-to-date quickly.

**Question 6:**

* I would suggest the team implement the following types of testing:
* Non-functional testing: a mobile application or website loads quickly, has adequate color, and is simple to navigate.
* Acceptance testing: to make that the system can be utilized as planned and that it satisfies the end-user's needs.
* Unit testing: in this kind of testing, the software's constituent pieces or components are examined to make sure they function as intended. Early in the development process, unit testing can assist in catching defects and difficulties, making it simpler to identify and resolve them.
* Integration testing: integrity testing examines how various software elements function together. This kind of testing can assist identify problems with component communication, compatibility, or problems with third-party libraries or APIs.
* Regression testing: to confirm that system modifications have not resulted in any unforeseen consequences or defects in the operation of the system.
* Functional testing: to make sure the software satisfies the intended functionality, this sort of testing entails comparing the software to the functional requirements. This kind of testing is necessary to make sure the software complies with the specifications and is appropriate for the intended use.
* In addition to the types of tests the team might consider performing, depending on the specific requirements and needs of the project, to guarantee that testing can be completed fast and accurately throughout each iteration, I would also advise adopting test automation tools. As a result, the team will be able to provide high-quality features more quickly and with fewer faults.