**Question 3**

* Functional Requirements:
  + Lecture can manage constructive questions and student group.
  + Lecturers can import student lists from the FAP system or from Excel files, assign presentation and review groups.
  + Lecturers can create reports on student activities such as answering questions, evaluating (voting) presentations of other students or groups.
  + Student can view constructive questions, give their own critical opinions on the questions, critically evaluate (vote) the presentations of other members or groups...
* Non-Functional Requirements:
  + The system needs to ensure high performance and reliability, requiring little training time to use.
  + Users can use the manual without having to attend official training courses.

**Question 6** :

* As a lecturer, I want to manage constructive questions and student group, so that I can facilitate an interactive learning environment to my classroom
* As a student, I want to view constructive questions, give their own critical opinions on the questions, critically evaluate (vote) the presentations of other members or groups... so that I can understand presentations more

**Questions 1:**

Based on the information you provided:

**Users:**

* Lecturers
* Students

**Customer:**

* FU (presumably the university) - Academic Staff would likely be the representatives

**Team:**

* 4-6 Experienced Developers from FU IT department
* Additional staff from other departments (exact roles not specified)

**Requirements Characteristics:**

* **Functional:**
  + User login with FU email account
  + Manage student lists (import, view)
  + Manage presentations and review groups (assign, view)
  + Create reports on student activities
  + View and interact with constructive questions
  + Evaluate presentations
  + View class statistics on constructivism methods
  + Monitor student feedback
* **Non-Functional:**
  + Secure information handling
  + High performance and reliability
  + User-friendly with minimal training required

**Time Constraints:**

* Tight schedule - First version in 3 months, entire project likely within 9 months

**Manager's Expectations:**

* Deliver a functional FU-EduNext system within the specified timeframe
* The system should support constructivism learning principles
* User-friendly and secure platform requiring minimal training
* Likely focus on Agile methodology for iterative development due to the tight schedule

=> Based on project characteristics above, I suggest Agile because :

**1. Adaptability to Changing Needs:**

* The requirements for the FU-EduNext system might evolve as stakeholders interact with the initial versions. Agile's iterative approach allows for incorporating new features or changes to existing ones based on user feedback throughout the development process.

**2. Fast Delivery of Working Software:**

* The project has a tight deadline, with the first version expected in 3 months. Agile focuses on delivering functional software in short iterations (sprints), enabling early user feedback and course correction if needed.

**3. Improved Communication and Collaboration:**

* Agile methodologies promote close collaboration between developers, lecturers (users), and other stakeholders. This fosters better understanding of requirements and facilitates addressing issues quickly.

**4. Reduced Risk Through Early Feedback:**

* By delivering working software in short cycles, stakeholders can test and provide feedback early. This helps identify and address potential problems early on, reducing overall risk.

**5. Increased Stakeholder Engagement:**

* Agile methodologies typically involve regular meetings with stakeholders to gather feedback and ensure the project aligns with their needs. This fosters a sense of ownership and increases the likelihood of project success.

**Questions 2:**

**1. Unit Testing**

* **Why?** To ensure individual units of code are working correctly. Developers can identify and fix bugs early in the development process, reducing the time and effort required to fix them later.
* **Who?** Software Developers

**2. Integration Testing**

* **Why?** To ensure different parts of the system work together correctly. This is especially important for the FU-EduNext project because it involves multiple components interacting, such as the lecturer portal, student portal, and academic staff portal.
* **Who?** Software Developers and Testers (if a separate testing team exists)

**3. System Testing**

* **Why?** To evaluate the entire system from a user's perspective. This helps to identify any *issues* with *usability, performance, or functionality* before the system is deployed to production. In the context of FU-EduNext, system testing would ensure features like presentation assigning, student feedback evaluation, and report generation function together as expected.
* **Who?** Testers and potentially a small group of Lecturers and Students (depending on the testing approach)

**4. Acceptance Testing**

* **Why?** To ensure the system meets the needs of the stakeholders (in this case, FU). This is the final stage of testing before the system is deployed to production. Acceptance testing for FU-EduNext would involve Lecturers, Students, and Academic Staff using the system to perform real-world tasks and provide feedback to ensure it meets their requirements.
* **Who?** Lecturers, Students, and Academic Staff

QUESTION 4: 4 use-cases and 2 actors

A screenshot of a diagram

Description automatically generated

A screenshot of a computer screen

Description automatically generated

Here are some general benefits of using Waterfall:

1. **Clear Structure and Documentation**:
   * Waterfall follows a linear and sequential approach, which makes it easier to plan and track progress. Each phase must be completed before the next begins, providing a clear structure to the project.
   * Extensive documentation is usually created, which can be helpful for future maintenance and for onboarding new team members.
2. **Well-Defined Requirements**:
   * In Waterfall, requirements are gathered and documented in detail at the beginning of the project. This ensures that all stakeholders have a clear understanding of what is to be delivered, reducing ambiguity.
   * Since the scope is defined upfront, it's easier to estimate timelines and costs, making it suitable for projects with well-understood and stable requirements.
3. **Ease of Management**:
   * The structured nature of Waterfall makes it easier to manage, particularly for large teams or projects. The project manager can track progress through the various stages, and it's clear when each stage is complete.
   * Each phase has specific deliverables, making it easier to manage deadlines and expectations.
4. **Effective for Certain Types of Projects**:
   * Waterfall is particularly effective for projects where requirements are well understood from the outset, such as in construction or manufacturing, where changes are costly or difficult to implement once the project is underway.
   * It works well when there is little uncertainty in the requirements and when a project is unlikely to undergo significant changes.
5. **Easier to Understand for Stakeholders**:
   * The step-by-step approach is straightforward and can be easier for non-technical stakeholders to understand. They can see the progress in a linear fashion, making it easier to visualize how the project is advancing.
6. **Predictability**:
   * Since Waterfall involves thorough planning, it tends to be more predictable in terms of budget, timeline, and deliverables. This predictability is often favored in projects with fixed-price contracts or where clients expect a clear timeline.