

## Activity 6.2 – Learning Highlights

**Key Challenges:** Systems often face issues like outdated technology, scaling difficulties, and technical debt. Human challenges include poor knowledge transfer and resistance to change, which can slow down progress.

**Effective Strategies:** Root cause analysis helps identify deep issues, while Agile methods break updates into smaller, manageable steps. Tools like CI/CD pipelines and automated testing ensure smooth updates.

**Tools and Frameworks:** Project management tools (JIRA, Git) and Agile frameworks (Scrum, Kanban) help teams stay organized and adaptable. Automated testing tools like Selenium ensure system quality during updates.

**Best Practices:** Balancing short-term fixes with long-term improvements, involving stakeholders, and prioritizing knowledge sharing are key for effective system maintenance.

### Application system change management:

1. Request
2. Request Review
3. Plan
4. Implementation
5. Review and Reporting

**Continuous integration and delivery (CI/CD):** CI/CD is a process in software development that makes it easier and faster to **test, integrate, and deploy** code changes. With CI, developers regularly push their code to a shared repository where it's automatically tested to catch any bugs early. CD takes it a step further by automatically deploying the tested code to production, so updates and new features get released quickly and smoothly. CI/CD helps developers work faster, cut down on mistakes, and keep the software up-to-date without tons of manual work.