Adaptive-Software-Development-Lab

1. Aim of the Lab

The aim of this lab is to understand and apply Adaptive Software Development (ASD) by using the Speculate–Collaborate–Learn (SCL) cycle. This lab helps in learning how software projects evolve under uncertainty using adaptive planning, collaboration, and continuous learning.

---

2. Background: Adaptive Software Development

Adaptive Software Development (ASD) is an Agile methodology used in uncertain and rapidly changing environments. Unlike traditional models that depend on fixed plans, ASD focuses on continuous adaptation, stakeholder collaboration, and learning from results. ASD follows the Speculate–Collaborate–Learn cycle to manage uncertainty effectively.

---

3. Characteristics of Adaptive Software Development

Mission-focused development
ASD focuses on achieving the project goal rather than following a rigid plan.

Component-based development
The system is built using small components that can be easily changed or replaced.

Time-boxed iterations
Work is done in fixed time periods to ensure continuous progress.

Risk-driven and change-tolerant planning
Planning is flexible and adapts based on risks and changes.

Continuous learning
Teams learn from each iteration and improve future work.

Why traditional planning fails:
Traditional planning assumes stable requirements, which is ineffective in uncertain and changing environments.

---

4. Speculate–Collaborate–Learn (SCL) Cycle

Speculate
Initial plans are made based on assumptions instead of fixed requirements.

Collaborate
Developers, stakeholders, and users work together and share feedback regularly.

Learn
The team reviews outcomes, learns from mistakes, and adapts the plan.

Example:
In a software project, features are planned based on assumptions, feedback is collected from users, and changes are made in the next iteration.

---

5. Speculate Phase – Written Explanation

In the speculate phase, planning is done based on assumptions such as changing user needs or evolving system requirements. These assumptions help create initial user stories, but they are not treated as final. The goal is to remain flexible and open to change.

---

6. Collaborate Phase – Written Explanation

The collaborate phase emphasizes teamwork and stakeholder involvement. Feedback from stakeholders is continuously collected and discussed. Decisions are made collectively instead of being imposed by management.

---

7. Learn Phase – Written Explanation

In the learn phase, the team reviews what worked and what did not. Lessons learned are documented, and improvements are planned for future iterations. This ensures continuous improvement.

---

8. Learning Review

Some assumptions may turn out to be correct, while others may be wrong. New requirements and insights emerge during collaboration. These learnings help refine the backlog and improve planning.

---

9. Adaptation and Re-Speculation

Based on learning, priorities are changed, user stories are modified, and new stories may be added. The next iteration is planned differently using updated knowledge.

---

10. Conclusion

This lab demonstrates that Adaptive Software Development is well-suited for projects with high uncertainty. By following the Speculate–Collaborate–Learn cycle, teams can continuously adapt, collaborate effectively, and improve their work. ASD focuses more on learning and flexibility rather than strict planning.