

Adaptive Software Development (ASD)

written in the **same academic depth and structure** as your earlier Agile labs, but **more detailed**, explicitly mapping **ASD concepts** → **practices** → **tool execution**.
This is **student-ready**, **printable**, and **PDF-convertible**.

LAB MANUAL

Title

Adaptive Software Development (ASD): Applying Speculate–Collaborate–Learn Cycle

Tool Used: Taiga

Lab Duration: 2 Hours

Lab Type: Hands-on, Tool-based

Mode: Individual / Pair Work

1. Aim of the Lab

The aim of this lab is to help students **understand and apply Adaptive Software Development (ASD)** by practically implementing its **Speculate–Collaborate–Learn (SCL)** cycle using an Agile project management tool (Taiga).

2. Learning Objectives

After completing this lab, students will be able to:

- Explain the philosophy and principles of Adaptive Software Development
 - Differentiate ASD from plan-driven and traditional Agile approaches
 - Apply the **Speculate–Collaborate–Learn** cycle in a project context
 - Use Taiga to support adaptive planning, collaboration, and learning
 - Demonstrate how software projects evolve under uncertainty
-

3. Prerequisites

Students should have:

- Basic knowledge of Agile methodologies
 - Understanding of Scrum roles and iterative development
 - Access to a Taiga account (cloud or institutional)
-

4. Background: Adaptive Software Development

Adaptive Software Development (ASD) is an Agile methodology designed for **high-uncertainty, complex, and rapidly changing environments**.

Instead of fixed planning, ASD emphasizes **continuous adaptation**, learning from results, and evolving requirements.

Core ASD Cycle:

1. **Speculate** – Adaptive planning based on assumptions
 2. **Collaborate** – Intensive stakeholder and team interaction
 3. **Learn** – Continuous feedback, reflection, and adaptation
-

5. Project Context

Project Title: Adaptive-Software-Development-Lab

Project Scenario: Online Course Registration System (OCRS)

Note:

This project is used only for **planning, collaboration, and learning simulation**.
No coding is required.

6. Lab Structure and Time Allocation

Phase	Description	Time
Phase 1	Understanding ASD Concepts	25 minutes
Phase 2	Applying Speculate Phase using Taiga	35 minutes
Phase 3	Collaborate Phase using Taiga	40 minutes
Phase 4	Learn Phase & Adaptation	20 minutes

PHASE 1: UNDERSTANDING ADAPTIVE SOFTWARE DEVELOPMENT (25 MINUTES)

Step 1: Study ASD Characteristics (15 Minutes)

Instructions:

Students study the following key ASD characteristics:

- Mission-focused development
- Component-based development
- Time-boxed iterations
- Risk-driven and change-tolerant planning
- Continuous learning

Student Task:

- Write **one short explanation** for each characteristic
 - Identify **why traditional planning fails** in uncertain environments
-

Step 2: Understand the SCL Cycle (10 Minutes)

Instructions:

Understand the three phases:

- **Speculate** → What do we believe will work?
- **Collaborate** → How do stakeholders and teams work together?
- **Learn** → What did we learn and how do we adapt?

Student Task:

- Write **one example** for each phase from real-world software projects
-

PHASE 2: SPECULATE PHASE USING TAIGA (35 MINUTES)

Step 3: Create ASD Project in Taiga (10 Minutes)

Instructions:

1. Log in to Taiga
2. Click **Create New Project**
3. Project Name:
Adaptive-Software-Development-Lab
4. Select **Scrum Project**
5. Enable:
 - o Backlog
 - o User Stories
 - o Sprints (Milestones)
 - o Wiki

Expected Outcome:

Project dashboard ready for adaptive planning.

Step 4: Speculative Planning (25 Minutes)

Instructions:

ASD planning is **assumption-based**, not fixed.

1. Identify **initial assumptions**, such as:
 - o Students will enroll in courses frequently
 - o Course availability may change
 - o Payment methods may evolve
2. Create **high-level user stories** based on assumptions.

Sample User Stories:

- As a student, I want to browse courses so that I can decide what to enroll in.
- As a student, I want to enroll in a course so that I can attend classes.

- As an admin, I want to modify course availability dynamically.

Student Tasks:

- Assign **tentative priority**
- Do **not freeze requirements**
- Mark stories as “*subject to change*” in description

Expected Outcome:

Backlog reflects **speculative planning**, not final requirements.

PHASE 3: COLLABORATE PHASE USING TAIGA (40 MINUTES)

Step 5: Sprint Creation and Task Collaboration (20 Minutes)

Instructions:

1. Create **Iteration 1** (Sprint 1)
2. Move selected user stories into Sprint 1
3. Break stories into tasks:
 - o UI Discussion
 - o Backend Logic
 - o Validation
 - o Feedback Collection

Collaboration Activity:

- Add **comments** simulating stakeholder feedback
- Assign tasks to different team roles

Expected Outcome:

Evidence of **collaborative decision-making**.

Step 6: Stakeholder Interaction Simulation (20 Minutes)

Scenario:

Stakeholder provides feedback:

“Students want the ability to drop courses after enrollment.”

Student Tasks:

1. Add comments reflecting stakeholder input
2. Modify existing user stories OR add new ones
3. Re-prioritize sprint backlog collaboratively

Expected Outcome:

Sprint evolves through **collaboration**, not command.

PHASE 4: LEARN PHASE & ADAPTATION (20 MINUTES)

Step 7: Learning Review (10 Minutes)

Instructions:

Students analyze:

- What assumptions were correct?
- What assumptions were wrong?
- What new information emerged?

Student Task:

- Document learning points in **Taiga Wiki**
-

Step 8: Adaptation & Re-Speculation (10 Minutes)

Instructions:

Based on learning:

1. Modify backlog priorities
2. Add/remove user stories
3. Plan next iteration differently

Expected Outcome:

Visible evidence of **adaptive evolution**.

7. Deliverables

Students must submit:

- Screenshot of ASD project setup
 - Screenshot of speculative backlog
 - Evidence of collaboration (comments)
 - Wiki page showing learning and adaptation
 - Reflection write-up (150–200 words)
-

8. Evaluation Criteria

- Understanding of ASD concepts
 - Correct application of SCL cycle
 - Evidence of collaboration and learning
 - Quality of reflection and adaptation
-

9. Conclusion

This lab demonstrates that **Adaptive Software Development is designed for uncertainty**. Rather than predicting the future, ASD emphasizes **learning, collaboration, and continuous adaptation**, making it ideal for complex and rapidly changing software projects.
