Agile Testing Flow for AUTOSAR ECU CAN Testing

1. Test Planning

What happens:

In the automotive ECU project, testing starts with **Test Planning**. The team prepares a detailed plan to test the **vehicle speed monitoring ECU** that communicates using **CAN protocol** under **AUTOSAR architecture**.

The team:

- Studies:
    - AUTOSAR layers: **MCAL, BSW, RTE, Application**
    - CAN communication flow
    - Speed range: **0 to 200 km/h**
- Decides:
    - Test types: **Smoke, Regression, Load testing**
    - Tools: **Python, pytest, python-can**
- Prepares:
    - Test cases for CAN transmit and receive
    - Validation rules for speed signals
    - Automation strategy

Output:

✔ Test Plan Document
✔ Test Case Document
✔ Automation approach finalized
✔ Testing scope defined

---

2. Daily Scrums

What happens:

Agile teams conduct **daily stand-up meetings** to track testing progress.

Each day the team discusses:

- Which CAN test cases were executed?
- Is Python script sending and receiving CAN messages correctly?
- Any failures in speed signal validation?
- Any AUTOSAR integration issues?

Based on this:

- Priorities are updated
- Issues are assigned
- Test activities are adjusted

Output:

✔ Daily progress tracking
✔ Faster issue resolution
✔ Better team coordination
✔ Updated test priorities

---

3. Test Agility Review

What happens:

In this phase, stakeholders and the testing team **review the testing progress and quality**.

They check:

- Whether all planned CAN signals are tested
- Whether speed values are validated correctly
- Whether Python automation is stable
- Whether test coverage is sufficient

They analyze:

- Failed test cases
- Defect trends
- Test execution efficiency

Output:

✔ Test coverage review
✔ Quality status report
✔ Process improvement actions
✔ Stability confirmation

---

4. Release Readiness

What happens:

In this stage, the team decides whether the **ECU software is ready for release**.

They verify:

- All critical test cases passed
- Regression testing is successful
- CAN communication is stable
- No critical AUTOSAR issues exist

Decision:

- If all OK → Approve release
- If not → Send back for fixing and re-testing

Output:

✔ Go / No-Go decision
✔ Release approval
✔ Stable and validated ECU software

---

5. Impact Assessment

What happens:

After release or after review, any:

- Change in speed logic
- Change in CAN signal
- Change in AUTOSAR configuration
- Bug fix or new requirement

is analyzed to identify:

- Which AUTOSAR module is affected?
- Which test cases need updating?
- What changes are required in Python automation?

This **starts the next Agile testing cycle** again from **Test Planning**.

Output:

✔ Impacted areas identified
✔ Test cases updated
✔ Automation updated
✔ Next sprint planning input

Short Summary

Agile testing for AUTOSAR ECU follows a 5-step cycle: Test Planning, Daily Scrums, Test Agility Review, Release Readiness, and Impact Assessment. This ensures continuous testing, fast feedback, and high-quality CAN-based ECU software using Python automation.

Agile Testing Flow for AUTOSAR ECU CAN Testing