Shift-Left Testing Checklist

1. Backlog & Requirements

- [ ] All **User Stories** have **clear Acceptance Criteria** (preferably in **Given–When–Then** format).
- [ ] Acceptance Criteria are **testable, measurable, and linked** to the story.
- [ ] **Definition of Ready** includes:
  - Test data identified
  - Automation intent noted
  - Dependencies clarified

---

2. Sprint Planning

- [ ] QA participates in planning to ensure **testability** and **automation feasibility**.
- [ ] Decide **which tests to automate** (unit, integration, e2e, performance, accessibility).
- [ ] Confirm **test environments** and **data strategy**.
- [ ] Identify **non-functional requirements** (performance, usability, security).

---

3. Development Practices

- [ ] The team follows **TDD or BDD** principles.
- [ ] **Unit tests** written for all new/changed code.
- [ ] **Integration tests** for service boundaries.
- [ ] **Static analysis & code quality checks** (e.g., Sonar) integrated into the pipeline.
- [ ] Feature toggles and **observability hooks** (logs, metrics) added.

---

4. CI/CD Pipeline

- [ ] **Automated tests** run on every **commit** and **pull request**:
  - Unit tests
  - Integration tests & E2E tests depending on the situation and used test environment
  - Static analysis
  - Feedback of automation result failures is shared with the developer immediately. For example webhook connected to communication channel (Teams, Slack, Email etc.)
- [ ] **Quality gates** enforced (coverage %, no new critical issues).
- [ ] **Nightly builds** run extended suites:
  - Regression tests
  - Performance smoke tests
  - Accessibility checks
- [ ] **Performance benchmarks** compared against previous runs.
- [ ] Reports and dashboards published for visibility to project management system for example Jira

---

5. Test Environments

- [ ] Scalable test environment available (e.g., **~20% of pre-prod capacity**).
- [ ] Stable test data and scenarios for **benchmark comparability**.
- [ ] Test data is controlled in a single place and updated on regular basis.
- [ ] Environment telemetry captured for RCA.

---

6. Sprint Review & Retrospective

- [ ] Present **test results**, coverage, and performance trends in Sprint Review.
- [ ] In Retrospective:

- Review **test automation effectiveness**.
- Identify **flaky tests**, coverage gaps, and pipeline bottlenecks.
- Update **Definition of Done** if needed.

---

7. Root Cause Analysis (RCA)

- [ ] Issues flagged during Daily for RCA consideration.
- [ ] RCA tasks added to Sprint Backlog when needed.
- [ ] RCA outcomes discussed in Retrospective and preventive actions defined.

---

8. Definition of Done (DoD) Includes

- [ ] All Acceptance Criteria met.
- [ ] Unit & integration tests pass; coverage agreed threshold.
- [ ] Static analysis and security checks pass.
- [ ] No known regressions.
- [ ] Performance within agreed threshold vs. baseline.
- [ ] Documentation updated.