1. Design and implement processes and communications
└ 1.1 Design and implement traceability and flow of work
└ 1.1.4 Implement source, bug, and quality traceability

1. What is traceability in DevOps and why is it important?
2. How do you link code changes to work items in Azure DevOps and GitHub?
3. How is bug traceability achieved in modern DevOps workflows?
4. What tools and features enable quality traceability across pipelines?
5. How do you visualize traceability from source code to deployment?
6. What practices ensure end-to-end traceability for compliance and audits?
7. How can automated testing results be linked to work items and requirements?
8. What are common pitfalls in traceability implementation?
9. How do you configure dashboards or reports to show traceability status?
10. What limitations should be considered when implementing traceability?

---

1. What is traceability in DevOps and why is it important?

- Traceability is the ability to track work items, code changes, bugs, and test results throughout the development lifecycle.
- It ensures transparency, accountability, fast root cause analysis, supporting compliance.

---

2. How do you link code changes to work items in Azure DevOps and GitHub?

- In *Azure DevOps*, include the work item ID in commit messages, PR titles, or branch names (e.g., #123).
- In *GitHub*, use keywords like Fixes #123 in commits or PRs to link issues.
- Both platforms automatically update the work item or issue with commit and PR references.

---

3. How is bug traceability achieved in modern DevOps workflows?

- Bugs are tracked as work items/issues in *Azure Boards* or *GitHub* Issues.
- Link code commits and PRs to these bugs using IDs or keywords.
- Automated and manual test results can be associated with bug work items to show resolution status.

---

4. What tools and features enable quality traceability across pipelines?

- *Azure DevOps* Test Plans, *GitHub Actions*, and pipeline integrations enable linking test runs, results, and code changes to work items.
- Test results, code coverage, and release status can be tracked and visualized against user stories and bugs.

---

5. How do you visualize traceability from source code to deployment?

- Use *Azure DevOps* traceability views, queries, or built-in dashboards to follow links from work items to code commits, builds, releases, and test results.
- Custom dashboards and reports can be built using *Power BI* or *Azure DevOps Analytics* for full traceability visualization.

---

6. What practices ensure end-to-end traceability for compliance and audits?

- Enforce linking of every code change and deployment to a tracked work item or requirement.
- Require automated association of test results and bugs to work items in pipelines.
- Maintain detailed audit trails and use approval workflows.

7. How can automated testing results be linked to work items and requirements?

- Configure pipelines to associate test runs with specific work items.
- Use test case management features in *Azure DevOps* to map test cases to requirements.
- Automated test tools can update work items with results using REST APIs or extensions.

---

8. What are common pitfalls in traceability implementation?

- Inconsistent work item references in commits/PRs.
- Lack of enforcement on associating code changes with work items.
- Poor test case management or missing links between tests and requirements.

---

9. How do you configure dashboards or reports to show traceability status?

- In *Azure DevOps*, use built-in widgets and queries to display links between work items, code, tests, and releases.
- *Power BI* can connect to *Azure DevOps Analytics* for advanced, custom traceability reporting.

---

10. What limitations should be considered when implementing traceability?

- Full automation may not cover all scenarios; some manual linking may be required.
- Cross-tool traceability (e.g., between *GitHub* and *Azure Boards*) may require custom integration or third-party tools.
- Visualization and reporting capabilities may be limited by platform or data availability.