1. Design and implement processes and communications
└ 1.3 Configure collaboration and communication
└ 1.3.1 Document projects with wikis and diagrams

1. What is the purpose of using wikis in DevOps projects?
2. How do you create and manage a wiki in Azure DevOps?
3. What are best practices for structuring project documentation in a DevOps wiki?
4. How do you use Markdown syntax for documentation in Azure DevOps or GitHub wikis?
5. What is Mermaid syntax, and how is it used in project documentation?
6. How do you embed and update diagrams in project wikis?
7. How do you manage permissions and access control for wikis?
8. What are best practices for keeping documentation up to date in fast-moving DevOps projects?
9. How can you automate documentation updates from source control history?
10. What are the benefits of linking work items and documentation in Azure DevOps?

---

1. What is the purpose of using wikis in DevOps projects?
Wikis provide a central, version-controlled location for project documentation, enabling teams to capture knowledge, share processes, and onboard new members efficiently. They support transparency, collaboration, and traceability across the DevOps lifecycle.

---

2. How do you create and manage a wiki in Azure DevOps?
In *Azure DevOps*, create a wiki by selecting "Wiki" in the project sidebar, then clicking "*Create Project Wiki*" or "*Publish code as wiki*." You can add, edit, or organize pages directly in the web interface or by editing markdown files in a code-backed wiki repository.

---

3. What are best practices for structuring project documentation in a DevOps wiki?
Structure documentation with clear hierarchy and navigation: use a home/landing page, group content by feature or topic, and keep pages concise. Regularly update docs, include diagrams, and ensure links between related pages for easy discoverability.

---

4. How do you use Markdown syntax for documentation in Azure DevOps or GitHub wikis?
Write documentation in plain text using *Markdown* (.md) files. *Markdown* provides formatting for headings, lists, code blocks, tables, links, and images. Both *Azure DevOps* and *GitHub* wikis support standard *Markdown*, allowing rich, readable content with minimal effort.

---

5. What is Mermaid syntax, and how is it used in project documentation?
Mermaid is a text-based diagramming syntax supported in *Azure DevOps* and *GitHub* wikis. It allows teams to create flowcharts, sequence diagrams, and other visualizations directly within *Markdown* by embedding code blocks marked with ```mermaid.

---

6. How do you embed and update diagrams in project wikis?
Embed diagrams by uploading image files and linking them using *Markdown*, or by using *Mermaid* code blocks for auto-generated diagrams. Update diagrams by editing the image file or updating the *Mermaid* code, ensuring version control and easy collaboration.

7. How do you manage permissions and access control for wikis?
In *Azure DevOps*, manage wiki permissions through *Project Settings > Permissions*. Assign read, edit, or administrative rights to users or groups. For code-based wikis, repository permissions determine access. In *GitHub*, wiki permissions follow repository settings.

---

8. What are best practices for keeping documentation up to date in fast-moving DevOps projects?

- Integrate documentation updates into the development workflow,
- review docs as part of pull requests,
- assign documentation ownership,
- and automate reminders or checks to prevent outdated content.

---

9. How can you automate documentation updates from source control history?
Use tools or scripts to generate or update release notes, changelogs, and API documentation from Git commit history or pull request descriptions. Automate documentation builds with CI pipelines to publish latest changes to wikis.

---

10. What are the benefits of linking work items and documentation in Azure DevOps?
Linking work items and documentation ensures traceability, provides context for features or issues, helps onboarding, and enables faster decision-making by centralizing information relevant to code, processes, and changes.