DWA_01.3 Knowledge Check_DWA1

---

1. Why is it important to manage complexity in Software?

- Maintainability: It gets harder to comprehend, modify, and improve software systems as their size and complexity increase. Developers may make sure that software is maintainable over its lifetime by actively managing complexity

- Collaboration: It becomes more difficult for team members to work together efficiently when the codebase is very complex. Keeping code simple makes it simpler for different developers to collaborate on the same codebase, understand each other's contributions, and integrate their work.

- Bug identification and correction: Complexity increases the chance of software faults and mistakes. Overly complicated systems make it more difficult to locate and analyze problems.

---

2. What are the factors that create complexity in Software?

- Scale and Size
- Legacy Code
- Team Collaboration
- Unattended Technical debt

---

3. What are ways in which complexity can be managed in JavaScript?

- Code Comments and Documentation: Clearly and simply describe the code's purpose with comments and Jsdocumentation
- Modularization: Divide the code into more manageable, chunks that concentrate on particular functions or features

- Functions: Employ functions to group and arrange logical code chunks. Functions facilitate greater separation of responsibilities, increase readability, and encourage code reuse.
- Coding style: Usage of javaScript coding style which make code more readable, maintainable, and maintains consistency throughout

---

4. Are there implications of not managing complexity on a small scale?
Yes,

- Risk of Bugs and Errors: Unmanaged complexity increases the risk of introducing bugs and errors into the codebase. Complex code structures, intricate dependencies, and lack of clear organization make it easier to overlook edge cases, introduce unintended side effects, or miss critical issues during development or maintenance.

---

5. List a couple of codified style guide rules and explain them in detail.

- Use descriptive variable and function names: using descriptive names for variables and functions. Clear and descriptive names help improve code readability and make the purpose and functionality of the code more evident

- Use consistent indentation and spacing: maintaining consistent indentation and spacing throughout the codebase Consistent indentation improves code

readability and makes it easier for developers to understand the code's structure and hierarchy.

- Prefer single quotes or double quotes for strings: choosing between single quotes (') or double quotes ("") for defining string literals

---

6. To date, what bug has taken you the longest to fix - why did it take so long?

A synthanx error bug, which was a semi colon was missing, it was hard to locate as I did not understand using error messages to help me,also did not have the aid of extensions as I was scripting using notepad

---