DWA_01.3 Knowledge Check_DWA1

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

- Managing complexity helps improve the code's readability, making it easier for developers to understand, modify, and fix issues as they arise.

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

- Inconsistent coding practices: When developers follow inconsistent coding styles and practices, it becomes more challenging to read and understand the code.
- Small bugs can break down a code.

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

- Modularization: Breaking down the code into smaller, self-contained modules can help manage complexity.
- Clear and Consistent Naming Conventions: Using meaningful and consistent names for variables, functions, classes, and modules enhances code readability and reduces cognitive load. It makes it easier for developers to understand the purpose and functionality of different parts of the codebase, reducing complexity.
- Documentation and Comments: Maintaining up-to-date documentation and adding explanatory comments to your code can greatly assist in managing complexity. Documenting complex algorithms, code structures, and design decisions helps other developers understand the codebase and reduces the cognitive load when working on complex parts of the system.

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

- Yes, Difficulty in understanding and modifying code: Even small pieces of complex code can be challenging to understand, especially for developers who are not familiar with it. This can lead to longer development cycles, increased chances of introducing bugs, and reduced productivity.

---

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

- Indentation and Formatting:

Rule: Use consistent indentation and formatting throughout the codebase.
Explanation: Consistent indentation and formatting improve code readability and maintainability. It helps developers quickly understand the structure of the code and facilitates spotting syntax errors.
- Naming Conventions:

Rule: Follow consistent naming conventions for variables, functions, classes, and other code entities.
Explanation: Naming conventions help make code more self-explanatory and improve readability. Consistent naming allows developers to understand the purpose and functionality of different code entities easily. Common conventions include using camel case (e.g., myVariable) for variables and functions, Pascal case (e.g., MyClass) for classes and constructor functions, and uppercase snake case (e.g., MY_CONSTANT) for constants.

---

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

- Debugging Challenges: non-blocking code, especially when involving multiple layers or frameworks, can introduce complexities in debugging. I had to take time to look at my code again to check for errors only to find out that I didn’t write my functions in the same way throughout the code so I had to name them properly using camel cases.
