DWA_01.3 Knowledge Check_DWA1

---

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

Facilitates debugging and troubleshooting, enables scalability, promotes reusability and maintainability, fosters collaboration among team members, and ultimately reduces costs and development time.

---

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

Inadequate requirements gathering and changing conditions, poor architectural design, lack of modularity and separation of concerns, tightly coupled dependencies, excessive code duplication, insufficient documentation, technical debt accumulation, evolving technologies and integration challenges, and the size and scale of the software system itself.

---

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

Using modular design patterns, writing clean and concise code, utilizing meaningful naming conventions, and implementing proper documentation

---

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

Yes, neglecting to manage complexity on small-scale coding can lead to difficulties in understanding and maintaining code, increasing the likelihood of bugs, limited scalability and reusability, challenging collaboration and knowledge transfer and increasing development time and cost when it eventually comes to even large coding.

---

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

Variable Naming Conventions: This rule outlines how variables should be named. It often recommends using meaningful and descriptive names that reflect the purpose or content of the variable. For example, variables representing a person's name could be

named `firstName` or `lastName`. By following consistent naming conventions, code becomes more readable, self-explanatory, and easier to understand and maintain.

Function Naming Conventions: this rule focuses on naming functions appropriately. It suggests using verb phrases to describe the action or behavior of the function. For example, a function that calculates the square root of a number could be named `calculateSquareRoot`. Consistent and descriptive function names enhance code readability and make it easier to comprehend the purpose and functionality of each function.

Indentation and Braces Style: This rule defines the preferred indentation style and placement of braces. It specifies whether to use spaces or tabs for indentation, the number of spaces per indentation level and whether braces should be placed on the same line or a new line. Consistent indentation and brace style enhance code readability, making it easier to follow the code's structure and logic. It also helps in avoiding syntax errors and ensures a uniform appearance across the codebase.

Comments and Documentation: This rule encourages developers to include informative comments that explain the purpose, functionality, or any specific considerations of the code. Documentation guidelines might require documenting public APIs, class interfaces, function parameters, and return values. Well-documented code enables other developers to understand and utilize the code effectively, simplifies maintenance, and promotes code reuse.

---

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

The bug that took the longest to fix was linking the javascript file to HTML file using “defer”, the reason why it took so long was that we were debugging a project and we only had to edit the javascript file so in the HTML, the developer forgot to include defer in the original code.
