DWA_01.3 Knowledge Check_DWA1

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

- Maintainability: By managing complexity, developers can create easier-to-understand and modify software, leading to improved maintainability.
- Collaboration: There are multiple developers working together on big projects. Without managing complexity, it becomes difficult for developers to collaborate efficiently.
- Software quality and debugging: Complexity often introduces risks of bugs. By managing complexity, developers can reduce the risk of introducing errors and improve software quality

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

- Size and scope: The size and scope of a software project can increase it's complexity. Larger systems with numerous modules, components, and dependencies tend to be more complex to design, implement, and maintain
- Requirements: Complex requirements can make software development challenging
- Architecture and design: Poor or overly complex software architecture and design choices can increase system complexity.

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

- Commenting and documentation
- Proper use of functions
- Modularity
- Abstraction
- Code formatting and style guide e.g Prettier extension

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

Yes, there are. JavaScript code can become difficult to read and understand when complexity is not managed properly. This can make it harder for developers to maintain and modify the code in the future. It also becomes more challenging to locate and fix bugs or issues in the code

---

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

Indentation and Braces:
Rule: Use consistent indentation and brace placement.

- Indentation and brace placement affect the readability and maintainability of your code. A common convention is to use a four-space indentation and place opening braces on the same line as the corresponding statement, while closing braces should be on a new line at the same indentation level.

Naming Conventions:
Rule: Use meaningful and descriptive names for variables, functions, and classes.

- Naming conventions improve code readability and make it easier to understand the purpose and functionality of different elements in your code. Some common conventions include using camelCase for variables and functions, and PascalCase for classes and constructor functions.

Line Length and Wrapping:
Rule: Limit the length of lines and wrap when necessary.

- Long lines of code can reduce readability and make it harder to understand the structure of the code. It is recommended to keep lines within a reasonable length, often around 60-80 characters. If a line exceeds this limit, you should wrap it to a new line, making sure the code remains readable. Break long lines after operators, commas, or other logical points.

---

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

On the recent final capstone, I struggled with the book preview code. I realized later that I misspelled my variables and I used a comma instead of a semicolon.