DWA_01.3 Knowledge Check_DWA1

---

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

Managing the complexity of software is crucial for the success of projects. It means making software that's easy to take care of, works well, can grow when needed, and is user-friendly. It also makes things smoother for the people building the software and ensures it does what users want it to do.

---

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

Complexity in software happens because software often needs to do many things at once, like handling different tasks, working with other software, and staying safe from hackers. When software becomes too complicated, it can be hard to change, might have mistakes, and could be slow. To make good software, we need to understand and control this complexity, which means organizing the code well, using clear names, and being careful about how different parts of the software talk to each other. This helps us create software that works well, is easier to manage, and does what people want.

---

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

Managing complexity in JavaScript involves breaking code into modular components, using descriptive variable and function names, adding comments and documentation, and code formatting tools, avoiding global variables, embracing functional programming principles, implementing robust error handling and testing, conducting code reviews, utilizing version control, employing build tools and dependency management, understanding design patterns, handling asynchronous operations effectively, and periodically refactoring code. These practices promote code organization, readability, and maintainability, ensuring that JavaScript applications are easier to develop, understand, and maintain while reducing the risk of bugs and errors.

---

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

Unmanaged complexity can impact projects by making them harder to understand, maintain, and extend, ultimately affecting the efficiency and quality of the development process and the software's reliability and user satisfaction.

---

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

1)Variable Naming Convention - descriptive variable names make your code more readable and maintainable. When someone reads your code, they should be able to understand the purpose of a variable just by looking at its name

2)Use of Semicolons - While JavaScript allows you to omit semicolons in some cases, it can lead to unexpected behavior and bugs. Using semicolons consistently ensures that your code runs as expected and avoids potential pitfalls.

3)Function Declaration vs. Function Expression - Prefer using function declarations over function expressions when defining named functions.Function declarations provide hoisting, which means you can use the function before it's declared in the code. This

enhances code readability and allows you to structure your code logically. Function expressions should be used when you need to assign a function to a variable.

---

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

During ITW there was a challenge we had to work on just before the final capstone. It took me the whole week to work on it. The reason it took me so long is because i was struggling to understand exactly what i need to do and when i didnt i was struggling on implementing my thoughts

---