DWA_01.3 Knowledge Check_DWA1

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

To make it easier to maintain, understand, collaborate and to save time and cost. It also reduces errors and bugs. It makes it easy to read and it keeps the code clean. All that is used to make it easier for both developers and users.

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

Some factors that create complexity in software are: Unclear description of the task or contradicting information which makes it difficult to deliver the right product. Poor design decisions and architecture makes it hard to understand and maintain or improve and add to the existing code. Over complexity of functionality and too many features or data can make it hard to keep track of the flow of the code. Concurrency makes managing synchronisation more challenging. Legacy code can be complex for a new developer to work on if it is poorly documented and unclear. Making code compatible across multiple formats for the interfaces to end users can create more complexity. Less time and resources put pressure on developers lead to shortcuts and quick fixes which results in bugs and errors. When it comes to complexity there are many more factors that can contribute depending on the nature of the project and are not limited to the above factors.

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

To manage complexity in JavaScript modularize code into smaller, self-contained modules. Abstract complex operations into simpler, reusable functions. Encapsulate data and behavior into objects using object-oriented programming principles based on their relation. Separate different concerns to keep code organized and maintainable. Promote code reusability by identifying common functionality. Apply design conventions to structure and manage complex code effectively. Using meaningful names and documenting code makes it more readable. Implement automated testing to ensure that the code is working.Regularly refactor and simplify complex code sections. Utilize tools and frameworks to simplify development and maintenance.

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

Failing to manage complexity in software can result in difficulties in understanding and maintaining the code, reduced code reusability, limited scalability, higher chances of bugs and errors, increased development time and costs, and impeded collaboration among team members. It is important to address complexity early on and adopt good software engineering practices to reduce these implications and ensure a more efficient software development process.

---

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

- **Use descriptive and meaningful function and variable names**
    - When creating variable and function names it is important to choose a name that is descriptive of its purpose and is not confusing to keep track of. This is supported by using naming conventions and formats which are consistent through the code and throughout all of your work.
- **Follow consistent indentation and formatting**
    - This is achieved by ensuring that the formatting and indentation is consistent throughout the codebase making it easier to read through and follow the structure of the code. In javascript there are many nested objects such as loops, if statements and objects/properties. Having these indented in a consistent manner makes it easier to understand the levels of nesting and reduces the complexity.

---

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

When creating the calendar for one of our challenges, I had trouble getting information to appear in the correct place in the table. It took me a long time to fix and eventually found that I was adding the data in the beginning of a loop when it should have been the end of the loop. Using better formatting the loop would have been easier to understand and I would have seen the error sooner.
