DWA_01.3 Knowledge Check_DWA1

---

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

From watching Schlak’s lecture videos I can understand the importance of managing complexity as it directly impacts the software's reliability, maintainability, and overall quality. A complex codebase is harder to understand, debug, and extend, leading to increased development time and a greater likelihood of introducing bugs. Complex code can hinder collaboration among team members, making it challenging to work together effectively. Simplifying code not only enhances the stability and performance but also enables easier long-term maintenance and scalability.

---

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

When requirements are not clear, it can introduce uncertainty and make it challenging to design a clear solution. The use of intricate algorithms and data structures can add technical complexity. A dependency on external libraries or systems (due to integration challenges) can also cause complexity. Multiple large codebases with numerous interrelated components can be hard to comprehend already but with poor documentation and lack of coding standards, the code will be even more difficult to maintain. Technology is evolving everyday at a rapid pace and this evolution could cause complexity as developers adapt to new tools and paradigms.

---

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

Complexity can be managed in multiple ways . By breaking down large tasks into smaller more manageable functions to promote a modular code structure. By maintaining clear and consistent naming conventions to enhance code readability. By leveraging comments and documentation to explain complex logic and usage of functions.

---

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

Not managing complexity in software can lead to various problems. Hard to understand code can be difficult to work with which can be a nest for more errors which will result in more time wasted on debugging. Moreover, making changes in complex code can introduce new bugs and makes it challenging for new team members to learn and contribute to the codebase. This proves that even on a small scale managing complexity is crucial for maintaining code that is efficient, reliable, and easier to work with.

---

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

1. Descriptive variable and function names: This means giving created variables or functions names that clearly describe what they do or what data they hold.
2. Indentation: This means adding space at the beginning of lines of code to show the structure of code. Proper indentation makes code more visually organized by making it easier to see where loops begin and close, where code blocks relate to each other and where functions start and end.
3. Consistent curly brace placement: This means that when using curly braces to define blocks of codes, it's important to place them consistently to enhance code readability and makes the code structure easier to understand.

---

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

As someone who is still new to the tech world, it seems kind of embarrassing when talking about any bugs I've had troubles with. One bug that took me a while to fix was in IWA-19. This bug was when working with the day/night mode. Although both modes did work, the default mode made all my text white. I realized later I was targeting the wrong thing.