DWA_01.3 Knowledge Check_DWA1

Readability, Maintainability, user satisfaction.____

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

- Size of the Codebase: Larger codebases are inherently more difficult to manage, understand, and modify.
- Code Quality: Poorly written, unstructured, or non-modular code increases difficulty in maintenance and debugging.
- Concurrency: Handling concurrent processes and ensuring thread safety adds significant complexity.

- Diverse Technologies: Using multiple programming languages, frameworks, and tools requires knowledge and coordination.
- Legacy Systems: Maintaining and integrating with outdated technologies can be challenging and time-consuming.
- Rapid Technological Change: Keeping up with the latest technologies and best practices requires continuous learning and adaptation.

---

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

- Classes and Objects: Use ES6 classes to define objects and encapsulate related functionality. This makes the code more modular and reusable.
- Inheritance and Polymorphism: Use inheritance to create hierarchical relationships between classes, and polymorphism to override methods in subclasses.
-
- Pure Functions: Write pure functions that avoid side effects and return the same output for the same input.
- Higher-Order Functions: Use functions that operate on other functions to create more abstract and reusable code.
- Immutability: Avoid changing the state directly. Use libraries like Immutable.js or immer to manage immutable data structures.

---

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

1. Increased Errors and Defects:
    - Bugs in Software Development: In software projects, unaddressed complexity can lead to an increase in bugs and defects. When codebases grow without proper management, they become harder to understand, debug, and maintain.
    - Product Defects: In manufacturing, failing to manage complexity can result in product defects, leading to higher rates of returns and customer dissatisfaction.

---

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

Detailed Explanation of Each Rule:

1. Serial (Oxford) Comma (CMOS):
   - Purpose: The serial comma is primarily used to avoid ambiguity in sentences that contain lists. It makes the separation between items clear, which is especially important in complex lists or when individual items within the list are themselves complex or contain conjunctions.
   - Example in Use: "For breakfast, I had eggs, toast, and orange juice." Here, the serial comma makes it clear that eggs, toast, and orange juice are distinct items.
2. Numerals (AP Style):
   - Purpose: The rule on numerals provides a balance between readability and consistency. Spelling out numbers one through nine keeps the text looking clean and avoids the visual clutter that can result from using numerals. Using numerals for 10 and above provides clarity, as larger numbers are more immediately recognizable in numeral form.
   - Example in Use: "The library has eight computers and 24 chairs." This follows the rule by spelling out "eight" and using numerals for "24."

---

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

---

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