# DWA_01.3 Knowledge Check_DWA1

---

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

Managing complexity in software helps keep it working well,making it easier to understand and fix, and allows people to work better together on code.

---

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

Software complexity is created by its size, inter dependencies, changing requirements, diverse technologies, concurrent tasks, varied user interactions, and connections to other systems.

---

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

Modular Code: Break your code into smaller, reusable modules to make it easier to manage and understand.

Use Functions: Write functions to handle repetitive tasks, keeping your code DRY (Don't Repeat Yourself).

Clear Naming Conventions: Use descriptive names for variables and functions to make the code more readable and maintainable.

Consistent Coding Style: Follow a consistent style guide to ensure uniformity and readability across your codebase
.
Comments and Documentation: Add comments and documentation to explain what your code does and why, aiding future maintenance and understanding.

Avoid Global Variables: Minimize the use of global variables to prevent conflicts and unintended side effects.

Use Modern JavaScript Features: Utilize ES6+ features for cleaner, more efficient, and more readable code
.

Testing: Write tests to verify that your code works as expected and to catch bugs early.

Refactoring: Regularly review and improve your code to make it simpler, more efficient, and more maintainable.

---

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

Difficult Maintenance: Unorganized and complex code is harder to update and fix, leading to more time spent on maintenance.

Increased Bugs: Complex code is more prone to errors, making it easier for bugs to occur and harder to find and fix them.

Poor Readability: Code that is not well-structured is difficult to read and understand, which can slow down development and collaboration.

Limited Scalability: As the project grows, unmanaged complexity can make it difficult to scale the software, leading to potential performance issues.

Reduced Re-usability: Poorly managed code is often too tangled to reuse parts of it in other projects, reducing efficiency and increasing redundancy.

Higher Learning Curve: New developers or team members will find it challenging to get up to speed with the project, slowing down progress and increasing on-boarding time.

______________________________________________________________________

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

## Rule 1: Use Consistent Naming Conventions

**Explanation**: Consistent naming conventions involve using a specific pattern for naming variables, functions, classes, and other identifiers in your code. This consistency makes the code easier to read and understand because developers can quickly grasp the role and scope of each element.

**CamelCase for variables and functions**: Start with a lowercase letter, and capitalize the first letter of each subsequent word (e.g., myVariable, calculateArea). This helps differentiate variables and functions from classes and constants.

**PascalCase for classes and constructor functions**: Capitalize the first letter of each word (e.g., MyClass, UserProfile). This convention signals that the identifier represents a class or a constructor function.
Uppercase with underscores for constants: Use all uppercase letters with words separated by underscores (e.g., MAX_HEIGHT, DEFAULT_COLOR). This distinguishes constants from regular variables, indicating that these values should not change.

---

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

Making a button render independently without triggering the other buttons , I wasn't aware that because the function or rather component in this context being modular as it is would not only need that to make it independent and render separately. I managed to fix it by creating a unique ID for each button so it can render independently .