DWA_01.3 Knowledge Check_DWA1

---

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

Managing complexity in software is important because it makes the code easier to understand and change, reducing errors. Simple code runs better and costs less to maintain. It helps teams work together and makes users happier. It also allows the software to grow and improve more easily.

---

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

Software gets complicated for several reasons. Lots of code, bad design, frequent changes, and no clear documentation add confusion. Quick fixes, using many systems, adding features, different coding styles, and managing many tasks also make it complex. Keeping these issues in check helps keep software simple and easy to use.

---

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

To manage complexity in JavaScript, start by breaking a code into smaller parts. Make sure to name everything clearly, like variables and functions, so it’s easy to understand what they do. Adding comments to explain your code helps too. Use tools like React to handle common tasks, and review your code regularly with your team to find mistakes. Simplify your code by removing anything unnecessary. Stick to a consistent way of writing code, and make sure to test it to check it works as expected. When dealing with asynchronous code, use Promises and Async/Await to make it easier to manage. And try to avoid using global variables too much to prevent problems.

---

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

Not managing complexity in small projects can cause several problems. The code can become hard to maintain, making simple changes take a lot of time and effort. This can lead to more bugs and slower development. It also makes it difficult for new team members to understand and contribute to the project. If you want to expand the project, complex code makes it challenging to add new features. All these issues can result in a poor user experience. Keeping code simple and well-organized is important, even for small projects.

---

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

1. **Use camelCase for Names**
   Use camelCase for variable and function names. For example, write let myVariable = 10, and function myFuction() {}. Start with a lowercase letter and capitalize the first letter of each following word.This makes names easier to read and understand.

2. **Use Single Quotes for Strings**
   Use single quotes for strings. For example, write a let message = ‘hello, world!’; This keeps your code consistent and avoids confusion with double quotes inside strings.

3. **End Statements with Semicolons**
   Always end your statements with semicolons. For instance, write let total = 0; Semicolons clearly mark the end of a statement, preventing errors.

---

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

One of the toughest bugs I’ve had to deal with was about names being inconsistent in the code. It took a long time to fix because it confused me and caused mistakes. To fix it, I had to go through the code carefully, clear naming rules, and make changes slowly.
