# DWA\_01.3 Knowledge Check\_DWA1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Software complexity can lead to a number of problems, including: Increased development time and cost, decreased reliability and maintainability, increased risk of bugs and security vulnerabilities and reduced user satisfaction. Managing complexity can help to mitigate these problems and improve the quality of software.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

The size and complexity of the software system

The number of developers working on the system

The use of third-party libraries and frameworks

The use of complex algorithms and data structures

The need to support multiple platforms and devices

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Using a well-defined coding style

Breaking down code into small, manageable functions

Using descriptive variable names

Commenting code to explain what it does

Unit testing code to ensure that it works as expected

Using a linter to check for potential errors

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Failing to manage complexity in software can result in the following:

Increased development time and cost

Decreased reliability and maintainability

Increased risk of bugs and security vulnerabilities

Reduced user satisfaction

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Descriptive variable names make code easier to read and understand. Consistent indentation makes code easier to follow. Comments help to explain what your code does, which can be helpful for other developers who need to maintain your code.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

There were no descriptive variable names and that made code hard to read and understand. There was no consistent indentation making code hard to follow. There was no function or variable declaration and a lot of syntax errors that made code hard to debug or understand. There were no comments to help explain what the sections of code were supposed to do which made the code none-maintainable.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_