

DWA_01.3 Knowledge Check_DWA1

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

It allows you to build your software in a more resilient manner allowing you to keep it more under control and in a structured form. So that when needing to revisit the code you have an understanding of what was intended at that time, and not having your code run away from you.

Because we create so much we have different thoughts being implemented in the creation that when we need to go back we have lost/forgotten that thought. It helps to reduce the risk of involving bugs or unintended behaviour from your code.

It is important to manage complexity as you will also be working with other coders, so documenting your code helps everyone stay on the same path of the desired outcome.

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

It could be the development of the software, needing to make it better or more user friendly, causing the code to grow and grow.

Lack of planning, which adds to the above mentioned. Not following a specific style or structure could cause in confusion and restructuring of the code causing more build up to the complexity of the software.

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

By use of clear code styles. There are different styles you could use in JavaScript the two standard are, Standard style and Airbnb style.

You could use clear naming variables that are more descriptive and direct to it's usage. It revolves a lot more around stylistic things such as (const or let) grouping them on their own and separate because they carry different functions.

Make sure your variables visually look different to keep separation.

Building the code in such a way that it does not run with the wrong input in place.

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

Yes, could start to lead to tech debt. You end up with a back log of data that needed to be documented down for understanding and brings in more debugging which consumes time and resources delaying the final product.

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

1 Making use of Constant naming conventions - there are different conventions to use such as camelCase or PascalCase. camelCase is used for naming variables, functions or methods. It starts with the first letter being lower case and the second word starting with upper case.

Pascal is used more for different classes or interface, the first letter is capital and so is the second word's letter.

2 Comments - Comments are important to writing good code. It clarifies the intent of the code and helps developers understand it better.

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

Syntax has been one of my struggles as well as understanding the purpose of what is needed and with that I mean comments. Remind myself what a specific segment is supposed to do and finish that in order to move on.
