DWA_01.3 Knowledge Check_DWA1

---

1. Why is it important to manage complexity in Software?
To make it easier to understand, maintain, collaborate with the team and to save time and costs. It also reduces errors and bugs. It makes it easy to read and it keeps the code clean, it makes it easy for both developers and users.

---

2. What are the factors that create complexity in Software?
Unclear description of the task,contradicting information which makes it difficult to deliver the right product. Poor design decisions and architecture makes it hard to understand and maintain and understand, also add to the existing code. Over complexity of functionality and too many features or data can make it hard to keep track of the code. Lack of documentation and unclear code is hard for any developer to work on, Legacy code can be very difficult for a new developer to work on, making code compatible across multiple formats for the interfaces to end users can create more complexity. Less time and resources put pressure on developers and lead to shortcuts and quick fixes which results in bugs and errors.
When it comes to complexity there are many more factors that can contribute depending on the nature of the project and are not limited to the above factors.

---

3. What are ways in which complexity can be managed in JavaScript?
Break your code into smaller parts,use different strategies, hide implementation details and promote code reusability, seperate your code according to their responsibilites, this makes sure that it is easy to test different aspects of the software.
Promote the reuse of code by creating reusable classes, functions and modules, always use descriptive names for variables, functions and classes.

---

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

Yes there are complications, such as difficulties in understanding and debugging the code, making it impossible to read and reuse the code. It can lead to increased costs and impact collaboration amongst developers. It is important to address complexity early and adopt good software practices.

---

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

1. Use meaningful variables and functions:
    - When writing code it is important to use descriptive and meaningful names for variables and functions. This helps you understand the purpose of the code.
    - Use names that clearly indicate what the variable or function represents.

2. Indentation and spacing should be consistent .

    - Consistent code make the code easier to read and understand, Indentation is used to organize the structure of the code, such as using loops and statements
    - Adding spaces and around operators, commas and brackets enhances cpde readability.

---

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

Null Pointer exception bug because I struggled with understanding why the big was there in the first place and for that reason it has taken me the longest to fix, and it would not really point me in the right direction.

Logic error: This has happened to me when I do not get the expected outcome that I need from the written code, this has happened a lot to me because I am still learning.
