DWA_01.3 Knowledge Check_DWA1

---

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

It is important to manage complexity in software for future maintainability and upgradability. Managing complexity allows complex projects to be broken down and be understood by everyone interacting with them. If complex projects are not managed, bugs most likely will be introduced and therefore the overall quality of the product or software is reduced. Another big issue is that it will take much longer to complete a complex project that has not been managed properly because it is not understandable by everyone and there are most likely many bugs.

---

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

A project that has a large number of features or requirements will create complexity in software because there will be many components within a singular project that need to be written and tested. A project that is poorly written or poorly organized will be hard to read and understand therefore causing complexity within the project. A project with a large number of bugs is a sign that the project is complex as this indicates that there are a large number of components within the software that are prone to errors. A project that takes a long time to add new features or make changes to current features is an indication of a complex project.

---

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

Ways to manage complexity include incorporating a coding style so that the code that is written is more understandable and structured in a standardized way, sticking to a code style will make code more readable for anyone working on a project, simplifying code and eliminating unnecessary code are two ways to manage complexity by making the code base less complex and less repetitive, abstraction can be used to manage complexity by only exposing the essential features and hiding details, decomposition can be used to manage complexity by breaking down large amounts of code into smaller modules.

---

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

Yes, the implications of not managing complexity include the possibility of bugs being introduced therefore the overall quality of the project is reduced, and the code may be unreadable therefore the time it takes to complete the project will be longer.

---

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

“When you access a primitive type you work directly on its value” - this means that when a primitive type is being called or modified, only the name of the variable is directly used, “Use const for all of your references, avoid using var” - this means that when declaring variables, const should be used instead of var in order to avoid potential bugs, “Use the literal syntax of object creation” - this means that when creating an object, instead of creating a new object constructor, a literal object must be created by using curly brackets.

---

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

The bug that took me the longest to fix was when I was debugging code and a single pipe was used as an or operator instead of a colon as it was an if else statement. The reason it took so long was because the console directed the error to a different line below, which didn’t actually have an error.
