DWA_01.3 Knowledge Check_DWA1

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

Managing complexity helps with saving time and money while making it simpler to maintain, comprehend, and collaborate. It also lessens bugs and faults. It maintains the code tidy and makes it simple to read. All of this is done to simplify things for both users and developers.

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

- It might be challenging to offer the proper product due to unclear task descriptions or conflicting information, which are some elements that contribute to software complexity.
- It is challenging to comprehend, maintain, or add to the existing code due to poor design and architecture decisions.
- It can be challenging to follow the flow of the code if there are too many features or data, or if the functionality is very complex.
- It is more difficult to manage synchronization when there is concurrency. If legacy code is unclear and inadequately documented, it can be difficult for a new developer to work on.
- Complexity can increase while trying to make code compliant with various end-user interface formats.
- Less time and resources put developers under strain, which causes them to take shortcuts and apply fast patches, which leads to faults and errors. There are numerous other variables that, depending on the nature of the project, may affect complexity in addition to those mentioned above.

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

Complexity can be managed in JavaScript as follows:

- Code should be modularized into smaller, independent modules to make maintenance easier.

- When it comes to abstraction and encapsulation this promotes code reuse, exposes key interfaces, and hides implementation specifics more of like putting the really important information and comment on what it does.
- Separation of Concerns, this is for easier comprehension and maintenance of the code and also segments the code into modules according to roles.
- Code organization keeps the codebase organized and consistent in its name selection, grouping, and documentation. Also creating reusable classes, modules, or functions in your code to cut down on repetition and enhance maintainability decreases complexity in JavaScript.
- Using of expressive and clear naming to make your code easier to understand.
- Decompose complex code into more digestible chunks, and then refactor repeated or complicated code. Writing thorough tests and documentation will help you ensure the correctness and clarity of your code for me and others working on the code.
- Static analysis and linting are technologies that can also be used to enforce coding standards and identify potential problems early.
- Encourage teamwork, reviews, and information sharing among members of the team by encouraging pair programming and code reviews. Code constructions should be made simpler, and performance should be optimized while taking readability and maintainability into account.

These procedures support cooperation, maintainability, reusability, and quality code throughout the development process.

---

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

Managing complexity in software can lead to challenges with understanding and maintaining the code, decreased code reusability, constrained scalability, increased likelihood of faults and errors, increased development time and costs, and make communication within the team broken and the work doesn’t get done in time. Therefore, to decrease these effects and ensure a more effective and reliable software development process, addressing complexity in the early stages of creating your code is important.

---

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

These rules encourage uniformity, readability, and maintainability in the code, which makes it simpler for developers to work with the codebase in the long run.

- **Descriptive and meaningful variables** - these improve the readability and maintainability of your code, give your variables and functions meaningful, descriptive names. For developers, descriptive names make it simpler to comprehend the function and behavior of variables and functions.
- **Consistent indentation and formatting** - using consistent indentation and formatting to make your code easier to read and maintain helps with making the code simpler to navigate, comprehend, and collaborate on when styles are consistent.

---

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

Syntax Errors and Run-by-One Errors,

- When it comes to Syntax Errors sometimes I will forget the right return code or close it quickly and couldn't run because my system didn't see it being called,
- Also when it comes to Run-by-One Errors the content of incorrect indexing or counting of elements in loops, arrays, or data structures will be lost along my code, and don't catch the typing errors I've made.
