

# DWA\_01.3 Knowledge Check\_DWA1

---

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

- You can attain more predictability in handling software tasks if you understand the level of complexity of the code being kept.
  - You can reduce the danger of presenting flaws into manufacturing if you prepare for and manage software complexity.
  - You can reduce software maintenance expenses if you proactively keep your software from ending up being exceedingly or needlessly complex.
  - You can maintain the value of your software possession and extend its helpful lifetime if you keep it from ending up being exceedingly complex.
  - You can approximate when it is much better to reword code than to keep preserving it.
  - Managing complexity in software promotes maintainability, scalability, debugging efficiency, team collaboration, time and cost efficiency and enables software to evolve effectively. It leads to more reliable, maintainable, and adaptable systems, reducing the risk of software failures and enhancing overall software quality.
- 

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

- The number of edges, nodes, and connected components in the software.
  - The size of the code and the length of a function.
  - The features and functionality of the system.
  - The type of problem to be solved.
  - Changing requirements.
  - Performance optimization.
  - Lack of documentation.
- 

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

- Identify issues.
  - Analyze possible solutions.
  - Be mindful that small things can break large things easily.
  - Break down your code into smaller, self-contained modules.
  - Work on code readability like proper naming and documentation
  - Eliminate useless code.
-

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

- The code might not run, it can make a code difficult to understand, modify and maintain.
  - Code might not be user friendly, other developers will not know what you have done.
  - Gets Hard to maintain, if not managed correctly.
  - It can introduce new bugs.
  - The damage or aftermath could cost a lot of money/time for a company or yourself
- 

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

- Choosing a format – keeps a code consistent, and this is purely for readability.
  - Use descriptive variable and function names, ('getUserData' instead of just 'get') – makes it easier for developers to read.
  - Use single quotes for string literals unless escaping is necessary – this rule defines a consistent approach to handling string literals, it improves readability and reduces unnecessary variations in the codebase.
  - Spacing and indentation – spaces between characters and the spaces between lines should be the same throughout the code, also the indenting is the indentation of the first line of a text.
  - Arrays – use literals and not constructors. ([ ]) not new arrays ().
  - Comments – Comments are critical good code examples; they clarify the intent of the code and help developers understand it.
  - Function names – use camelCase.
  - Loop initialization – when loops are required, choose the appropriate one For;;, for ...of, While, etc
  - Control statement – there is one notable case to keep in mind for the if ...else control statement. If the if statement ends with return do not add 'else' statement.
- 

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

- Basically, one of my first couple of tasks in JavaScript - all the JavaScript, html and CSS files was given to us. I did not know about debugging a code, also did not know about referring to or importing the JavaScript file into the html file. So, I struggled to run the code to get the JavaScript file to work on the html file.
  - I struggled a long time just to figure out why the code did not run with the html file. In the head of the html files, you should always refer to the ./scripts.js file, (JavaScript file) and the defer type should be Module. Very simple but showed me the hard way of not forgetting this.
-