DWA_01.3 Knowledge Check_DWA1

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

I think it is important to manage complexity in software because this is a very effective way to avoid undetected bugs that might reside in the code you are working with. Managing these complexities can not only help companies save money if a product fails because of an undetected software problem, but also pick these bugs up before they reach a certain stage in the production of said product/service.

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

Some causes of software complexities may include; the use of bad practices by programmers such as not structuring their code properly and leaving it vulnerable to bugs that may not be seen initially. Not extensively testing their code before merging/finalizing their code,not fully understanding the code that you are working with.

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

One of the ways in which complexity in javaScript can be managed is to thoroughly test your code before pushing/finalizing it. This can be done by creating error scenarios for bugs that may not show up or register on the console.

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

There are implications of not managing complexities on a small scale, however the consequences are rarely as severe as when this is not done properly on a larger scale. Eg. The knight Capital group stock crash - this was a big scale software problem that affected the entire company costing them an estimated $440 000 000

---

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

The following are from the readme in the Airbnb github link and concern JavaScript Variables: 13.1) Always use ‘const’ or ‘let‘ to declare variables.Not doing so will result in global variables. We want to avoid polluting the global namespace.Captain Planet warned us of that.eslint: `no-undef` `prefer-const`

```
// bad
superPower = new SuperPower();

// good
const superPower = new SuperPower();
```

—---

13.2) Use one `const` or `let` declaration per variable or assignment. eslint: `one-var`
Why? It’s easier to add new variable declarations this way, and you never have to worry about swapping out a `;` for a `,` or introducing punctuation-only diffs. You can also step through each declaration with the debugger, instead of jumping through all of them at once.

```
// bad

const items = getItems(),

    goSportsTeam = true,

    dragonball = 'z';



// bad
// (compare to above, and try to spot the mistake)

const items = getItems(),

    goSportsTeam = true;

    dragonball = 'z';


// good

const items = getItems();

const goSportsTeam = true;

const dragonball = 'z';
```

---

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

There have been lots of big bugs that have taken me hours to solve, however the one that immediately springs to mind is a bug that I encountered during our last capstone project which took me almost a whole day to solve /work out. It took me this long to realize the eventual problem was an error in my function naming which was done with camelCasing. E.g noahLovesCode was referred to later as noahlovesCode.
