Nicole P. Satiembre **February 28, 2024**

Cebu Technological University-Main Campus- Intern

97 Things Every Programmer Should Know

Chapters 66-70

66. Prevent Errors - Giles Colborne

Errors usually occur when there is a wrong input from the user. However, people make mistakes in a predictable, systemic way. So, instead of blaming it all the user, there is actually a way to improve the interaction between the user and the system so as to limit the occurrence of mistakes. For instance, we would try to limit our user input for dates, instead of letting them type on any format, we let them use calendar input showing only the allowed dates. Another common problem is formatting errors and a way to avoid them is by offering cues– for instance, a label. In general, instructions are not highly effective at preventing errors. Another way to avoid errors is to offer defaults. Logging and analyzing undo actions can also highlight where the interface is drawing users into unconscious errors, such as persistently clicking on the “wrong” button. No matter which approach we might take, above all, errors are caused by misunderstandings between the user and the software which is why we need to understand how users think, interpret information and make decisions.

67. The Professional Programmer - Robert C. Martin (Uncle Bob)

One important trait of a Professional Programmer is having a sense of “Responsibility”. We do not blame others for our mistakes, rather we own it. We take responsibility of our choices, our career, our schedule, our workmanship. As a professional, we take responsibility of our own career and how to stay in the game by continuous learning. It is not someone else’s job to train us, not even our own employer, it is our responsibility to train ourselves. As professionals, we take responsibility of the code that we write, we do not release a code unless it works. We don’t release codes we are not sure of. Professionals are team players, we take responsibility for the output of the whole team and not just our own work. We help one another, teach one another, and learn from one another. Professionals do not tolerate big bug lists. Professionals do not make a mess. As professionals, we take pride in our workmanship. We keep our codes clean, well-structured, and easy to read. Professionals follow agreed-upon standards and best practices. They never rush. To recap, the mark of professionalism is being responsible. When the pressure mounts, professionals hold ever tighter to the disciplines they know are right.

68. Put Everything Under Version Control - Diomidis Spinellis

There are a lot of free version control tools we can use today, personally I use Git and Github. We only to to learn two basic operations: committing and updating our code. Version Control helps us track the history of our code, who wrote the code and refer to a file or project version through a unique identifier. Most importantly, we are confident to make changes because we have a back-up of the older version. It is also helpful when working with a team of developers. When we work on independent software parts, version control helps us integrate our codes easily, we get notified for each committed change. In addition to the source code, we must also include the documentation, tools, build scripts, test cases, artwork and even libraries. There are also a few guideline that helps us more effective when using version controls: Commit each logical change in a separate operation; Accompany each commit with an explanatory message; Finally, avoid committing code that will break a project’s build.

69. Put the Mouse Down and Step Away from the Keyboard - Burk Hufnagel

Often times when facing a difficult problem, we used up large amounts of time but still find ourselves stuck on it. The reason behind this is the when we're coding, the logical part of our brain is active and the creative side is shut out. When this scenario occurs, we might want to do ourselves a favor and take a break. Sometimes, we think that our first attempt was the best solution to the problem, however, it doesn't always work and we need to give ourselves a break. Once we really understand the problem, we can go and something else that will activate the creative side of our brain such as sketching out the problem, listening to some music or just take a walk outside. Sometimes, the best thing we can do in order to find solutions to a challenging problem is to put the mouse down and step away from the keyboard.

70. Read Code - Karianne Berg

As a programmer, we must make it a habit to read other people's code. Although we may find it hard because our brains are so used to writing code and solving problems that reading codes feels tedious, boring, and impossible. When reading other people's code, we , stop and think for a moment. Is the code easy or hard to read? If it is hard to read, why is that? Is the formatting poor? Is naming inconsistent or illogical? Are several concerns mixed in the same piece of code? Perhaps the choice of language prohibits the code from being readable. Try to learn from other people's mistakes. Aside from learning from other people's mistakes, we can also learn how to differentiate between elegant codes and unreadable codes. We can also try reading our old code, this can also be an enlightening experience. We can find some embarrassing entertainment value. This way we can appreciate how we have developed our skills over the years, it can be truly motivating. Although reading a book and documentation could help us improve our knowledge, reading codes is one way to improve our programming skills.