Software design principles(Basics)
DRY, YAGNI, KISS

Upcode Software
Engineer Team

-2023-

Content

1. What is software design principles?
2. What is Dry?
3. Why we need Dry principle?
4. What is KISS?
5. Why we need KISS?
6. What is YAGNI?
7. Why we need YAGNI?
8. Summery
9. Resources

What is software design principles?

Software Design Principles are a set of guidelines that helps developers to make a good system design.

Software Design is also a process to plan or convert the software requirements into a step that are needed to be carried out to develop a software system. There are several principles that are used to organize and arrange the structural components of Software design. Software Designs in which these principles are applied affect the content and the working process of the software from the beginning.

Software design principles

Don't Repeat Yourself (DRY)

Every piece of knowledge must have a single, unambiguous, authoritative representation within a system.

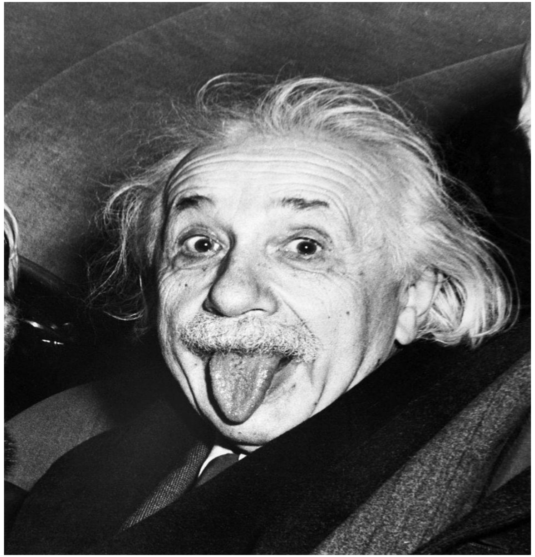
DRY or the “Don’t Repeat Yourself” principle states that pieces of repeating code should be extracted into methods, services or class hierarchies, depending on the circumstances. While it may seem to be a technique to decrease the code base, the main purpose of this principle is to **discourage Copy-Paste Programming** and make our code more maintainable by having only single instances of reusable code in our application.

Why we need Dry principle?

- DRY principle is to improve the maintainability of code during all phases of its lifecycle.
- if the information needs to be changed, there is only one place to change it.
- It prevents mess of code.
- It helps to write more clear code.

What is KISS? (1/n)

If you can’t explain it, you don’t understand it well enough.



What is KISS? (2/n)

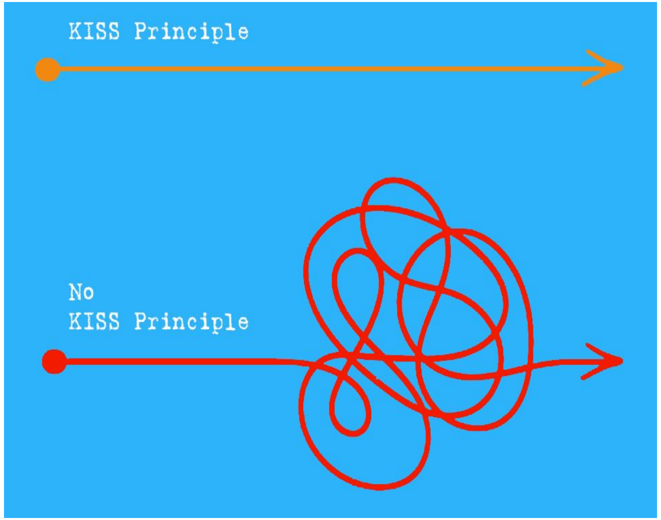
The simpler the explanation and the simpler the product, the more likely it is that the output will be useful to others.

The phrase; “keep it simple, stupid” is thought to have been coined by the late Kelly Johnson, who was the lead engineer at the Lockheed Skunk Works (a place responsible for the S-71 Blackbird spy plane amongst many other notable achievements). It is worth noting that Kelly’s version of the phrase had no comma and was written “keep it simple stupid”.

What is KISS? (3/n)

The KISS principle is also offered in two other forms (for those who feel delicate about the inclusion of the word “stupid”):

- Keep it short and simple
- Keep it simple and straightforward



How to Apply KISS?

To keep our program as simple as possible, we should:

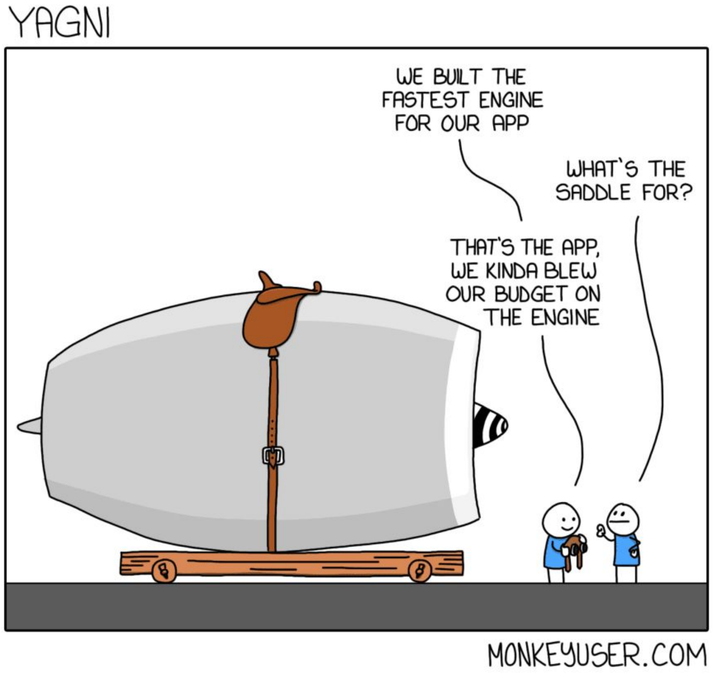
- Write smaller programs
- Remove the methods and instances in our code that are never used
- Write readable and transparent programs
- Use composition, which helps us reuse existing code
- Use modular programming

We can remember that **each class in our code should have only one responsibility. For new jobs, we should create new classes. This is also called the Single Responsibility Principle, one of the SOLID principles of object-oriented design.**

Why we need KISS?

- It makes our code easier to change and maintain in the future
- It makes code easier to read and understand by other developers.
- Simple code is easier to test when doing automated testing

What is YAGNI?


YAGNI
WE BUILT THE FASTEST ENGINE FOR OUR APP
WHAT'S THE SADDLE FOR?
THAT'S THE APP, WE KINDA BLEW OUR BUDGET ON THE ENGINE
MONKEYUSER.COM

YAGNI stands for ***You Ain't Gonna Need It.*** It's a principle from software development methodology of Extreme Programming (XP). This principle says that you should not create features that it's not really necessary.

Why we need YAGNI?

- The principle helps developers avoid wasted effort on features that are assumed to be needed at some point.
- YAGNI focus on removing unnecessary functionality and logic, and KISS focus on the complexity.
- The main goal of the YAGNI principle is to avoid spending time and money on overengineering things that you think you will need later on.

Summery

Software Design Principles are a set of guidelines that helps developers to make a good system design.

According to the DRY principle, every discrete chunk of knowledge should have one, unambiguous, authoritative representation within a system.

Keep it simple, stupid (KISS) is a design principle which states that designs and/or systems should be as simple as possible.

YAGNI principle ("You Aren't Gonna Need It") is a practice in software development which states that features should only be added when required.

Resource

- DRY, KISS & YAGNI Principles
- YAGNI Principle in 100 seconds
- DRY: Do not repeat yourself
- Keep it Simple, Stupid – How to Use the KISS Principle in Design
- KISS, YAGNI, DRY – three principles that every developer should know about

Thank you !

Prepared By Abbos Fayziboev

abbosfayziboev@gmail.com