

# Clean Code:

# A Handbook of

# Agile Software Craftsmanship

Chapter 01: Clean Code

## What is clean code? What good is it to me?

---

As Robert C. Martin states, writing **dirty code** will lead to **failure** in our project. He also refers that writing clean code may look a **difficult** and **time consuming** job to do, but it actually **speeds** the development and debugging **up**. Even if it is re-doing to clean a dirty code and rewrite it, it is worth it. Because it leads to creating **value**.

Spending time for cleaning the code is not only **cost-effective**, but also the condition of **professional survival**. **Disturbances** quickly lead to **slowness** and violate the **deadline**. The only way to move **quickly** and meet the **deadline** is to keep the code clean at its highest level and at all times. So the importance of clean code can not be denied. In the first chapter of this book we learn more about the concept of clean code and we try to check several different definitions of it.

An illustration of a spiral-bound notebook with a white page and an orange cover. The spiral binding is at the top. The text is written on the page.

Why should my code be clean?

---

Martin Fowler(British Software Engineer):

Any fool can write a code that a computer understands. Good programmers write a code that humans can understand.

So the code has to be clean in order to be easy-to-understand by a human.  
Which can be the author of the code or someone else.

A spiral-bound notebook with a black wire spiral at the top, set against a solid orange background. The notebook page is white and contains handwritten text and a large percentage.

Is it worth to consume time to keep my code clean?

---

%100

Considering the **time** and **energy** consumed for debugging a dirty code,  
we find it absolutely effective to clean our code.

## What happens if my code is not clean?

---

It is said that bad code is never gone. Meaning it doesn't matter how much we clean the code. It still **smells bad**. But the bad smell better be reduced as much as possible. There is a phase in software development where adding new features to the program or editing the codes that have a bug will lead to more **dysfunctionality** in the code. The code is so **unrecognized** that in some point it will be impossible to change it. This phase is called **wading**. Because the software is taking its final steps. The programmer is struggling in a swamp and walks among the hidden traps, hopping to find a way out. But the only thing he finds is more senseless codes.

## What do the pioneers of Software Industry say about Clean code?

---



Bjarne Stroustrup

I like my code to be **elegant** and **efficient**. The logic should be straightforward to make it hard for bugs to hide.



Grady Boosh

Clean code is **simple** and **direct**. Clean code is read like a **well-written prose**. Clean code never obscures the **designer's intent** but rather is full of crisp abstractions and straightforward lines of control.

## What do the pioneers of Software Industry say about Clean code?

---



Dove Thomas

Clean code can be **read**, and **enhanced** by a developer other than its original author. It has **unit** and **acceptance** tests. It has **meaningful names**. It provides **one way** rather than many ways for doing **one thing**. It has **minimal** dependencies, which are explicitly defined, and provides a clear and minimal API. Code should be **literate** since depending on the language, not all necessary information can be expressed clearly in code alone.



Michael Feathers

Clean code always looks like it was written by someone who **cares**. There is nothing obvious that you can do to make it better. All of those things were thought about by the code's author, and if you try to imagine improvements, you are led back to where you are, sitting in appreciation of the code someone left for you—code written by someone who **cared** deeply about the **craft**.

A spiral-bound notebook with a white page and an orange background. The spiral binding is at the top. The text is written in a casual, handwritten style.

In brief clean code has these features:

---

- It is short and minimal.
- It has comprehensive tests.
- It is efficient and easy to maintain and develop.
- Reading it is like reading a newspaper.
- Naming is the way that the reader understands the intent of the author by reading the name of modules(class, method, variable, ...).
- Every module is made for one and only one purpose.