

Mohammadmahdi Farrokhy

Clean Code:

A Handbook of

Agile Software Craftsmanship

Chapter 06: Boundaries

Third Party Code

Third Party Code is a code that we use, but did not write. Like calling an API. This could lead to multiple issues:

Issue #1

1. We must learn to work with the API.
 - **Solution - Learning Tests:** When dealing with third-party code, it's important to understand it. However, this can be a challenging task. There are a few ways to go about it.
 - Reading the API's documentation, which can be time-consuming.
 - Search online, especially if we're using a well-known library or framework.
 - The most effective method: Create learning tests. This approach is quick, cost-effective, and secure.

Issue #2 and #3

2. We might depend too heavily on APIs in our program. If the API's main functionality changes, we'll be forced to change our code to adapt with the new version; and this violates *Open-Close Principle* from *S.O.L.I.D*.
3. This API might offer more functions than we actually require.

Issue #2 and #3

- **Solution To Problems #2 And #3: Wrapper Module:** Once we've got familiar with the third-party code, we can start incorporating it into our program. There are two approaches for that:
 - **The less effective approach:** Using the third-party code directly all over our program, which can lead to problems when we update API's version. We'd be forced to modify anywhere we used the API.
 - **The better approach:** Creating a wrapper class that interacts as an interface between the API and our codebase. This approach separates the API from the rest of the software. Any issues or conflicts resulting from different API versions will be isolated in the wrapper class.

Benefits Of Using A Wrapper Class

- **Controlled API Usage:** We can restrict the API's functions to necessary.
- **Reduced Dependency:** By relying on the wrapper class, we decrease dependency to the API.
- **Easy Adaptation:** Changing the API or library becomes simpler since modifications are only limited to the wrapper class.
- **Enhanced Testing:** While we can't directly test the API itself, we can test the wrapper class effectively.
- **Forward Planning:** Imagine a scenario where the API isn't available yet. We could create other modules of our program, test and use them without problems. This is because our dependency on the API is limited to just one class. This grants us and the API development team more flexibility and time for development.