

Clean Testing



Tests are mostly **done manually**; we struggle to test some part of the code due to its design or lack of knowledge. We **ignore the coverage** of a large part of our code base. Our feature additions cause many tests to be modified.



We write **good unit tests**, we **name** them according to the application behavior being tested, using as much production code as possible. We **take care** of our tests as much as the production code under test. We rarely add tests after pushing our code.

Clean Code

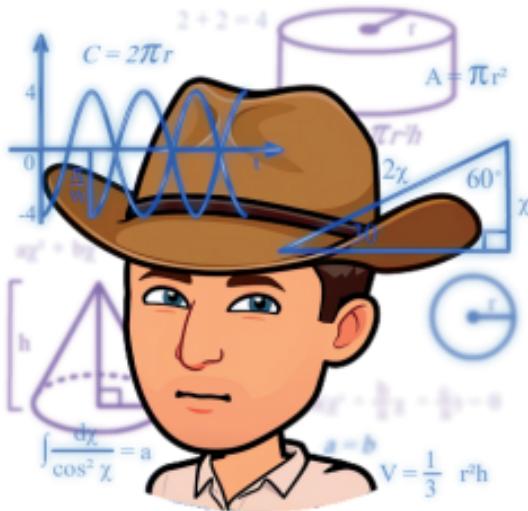


We are **not aligned** on what is a "good piece of code".
We also have code that is **difficult to maintain**.



We are **all aligned** in saying that our code is Clean.
We have a clear definition of our development standards.

Solution Design



We work **alone** on our team topics and define our solutions **by our own**.



We **collaborate** with other individuals from the organization to design our solutions.
Our solutions are **aligned** with architecture guidelines.

CI / CD



None or only part of our source code is built through a CI platform. We often have failures, which can last for a long time... Integration problems are frequent, deployments are done mostly manually.



Each commit triggers a build of the software, then a series of automated tests provide us a quick feedback on it. Our CI is adapted to our processes/practices. We are collectively responsible for the failures.

Tech Debt Management



The technical debt increases uncontrollably either because of a lack of competence or because of the delivery pressure. We have elements of debt reported with code analysis tools. We have **not prioritized the fixes** to be made or allocated time to improve our code.



We are concerned about code quality and **works daily to fight our technical debt**.
We have a few debt items identified in our backlog, prioritized according to their criticality.
We process the corrections over the course of the sprints.

Collective Ownership



Some of us are **specialized** on certain types of tasks / parts of the code.
We each work on **our own** User Stories.



We have a **strong culture of sharing** within the team.
Pair / Mob programming and **Code Review** are used to improve our products and grow together.
We are all able to work on any piece of code.

Refactoring



We are **afraid to touch / change some pieces of code because of its design or lack of tests.**
Refactoring is not part of our development practices.



We continuously **apply the "Boy Scout Rule"**, so we always leave the code we are editing a little better than we found it.