Unit testing?

The measurable benefits of unit testing.

Maciej Polańczyk
maciej.polanczyk@stxnext.pl

Schedule

- What is a unit test?
- What is said about unit tests
- Can we measure if unit tests provide any benefits
- Do I need unit tests

What is a unit test?

What is a unit test?

A fast test of a small unit of application.

What is said about UT?

What is said about UT?

- Less bugs
- Solving problems with less code and better code architecture (TDD)
- Faster implementation of new features

Can we measure it?

- Less bugs (very hard to measure)
- Solving problems with less code and better code architecture (TDD) (very hard to measure)
- Faster implementation of new features (let’s focus on this)

Can we measure it?

Let’s build a car

Can we measure it?

We will work with electronics engineers

Can we measure it?

We need a car door window controller

Can we measure it?

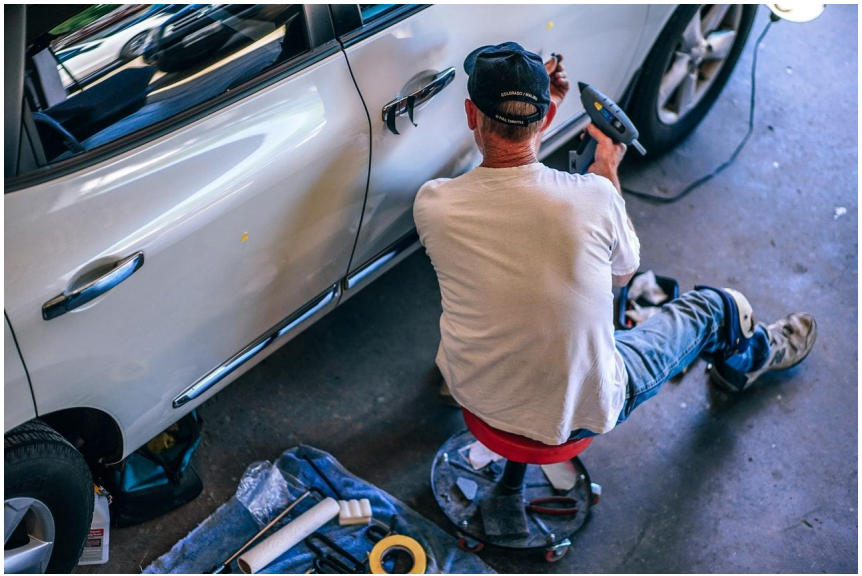
We give him the documentation

Can we measure it?

Few days / weeks of hard work

Can we measure it?

Done, let’s check if it works



Can we measure it?

Time spent on testing without unit tests:

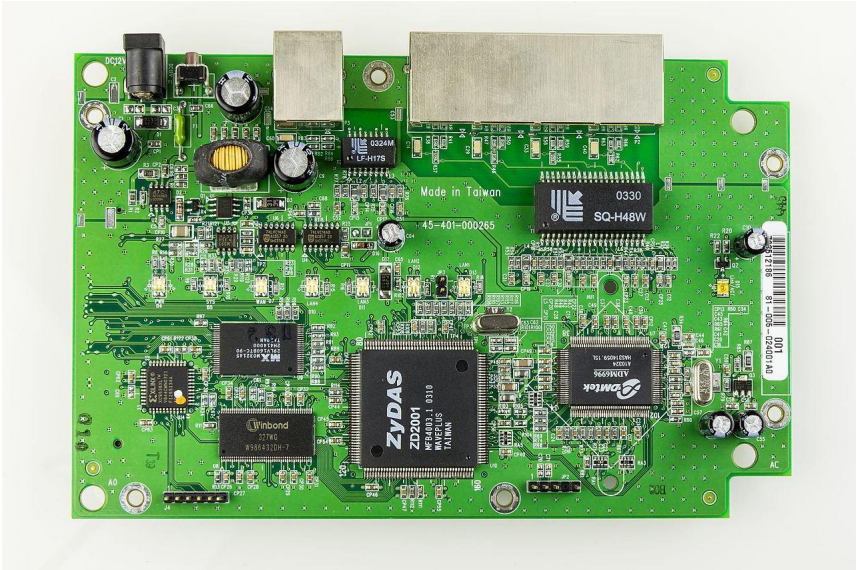
*testing_time_without_unit_tests = n * (installation_time + testing_time)*

Can we measure it?

Can we do it differently?

Can we measure it?

Build and use circuit board for testing



Can we measure it?

Small change in a project

Can we measure it?

Car will have **four** doors instead of **two**

Can we measure it?

Imagine his face

Can we measure it?

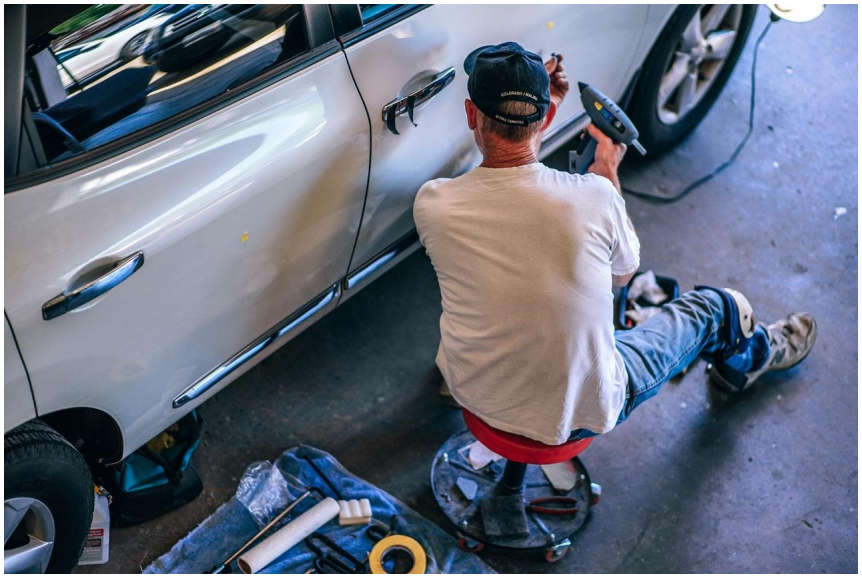
We give him more documentation

Can we measure it?

Few days / weeks of doing **witchcraft**

Can we measure it?

Done, let’s check if it works



Can we measure it?

Time spent on testing without unit tests:

*testing_time_without_unit_tests = n * (installation_time + new_feature_testing_time) + previous_feature_testing_time*

Can we measure it?

Time spent on testing without unit tests:

*testing_time_without_unit_tests = n * (installation_time + fifth_feature_testing_time) + first_feature_testing_time + second_feature_testing_time + third_feature_testing_time + fourth_feature_testing_time*

Summary

Without unit tests:

- with each feature **testing time grow, it may grow very fast**
- **testing time is unpredictable** because depends on number of bugs (the more complex project, the more bugs you will find)

Can we measure it?

Time spent on testing with unit tests:

*testing_time_with_unit_tests = n * testing_time + testing_board_extension_time*

Can we measure it?

Time spent on testing with unit tests:

testing_time_with_unit_tests = testing_board_extension_time

Summary

With unit tests:

- with each feature **testing time grows very slow**
- **testing time is predictable** because development time can be easily estimated

Summary

Yes, we can implement features faster

Do I need unit tests?

*n * (installation_time + new_feature_testing_time) + previous_features_testing_time*

> creating_unit_tests_time ??

Do I need unit tests?

You will benefit from unit tests when:

- your project will **grow in time**
- your **features are complex**
- your **environment is complex**
- **predictable time to market** is crucial for your project
- **bugs** in your project can **cost you a lot of money**

Thank you!