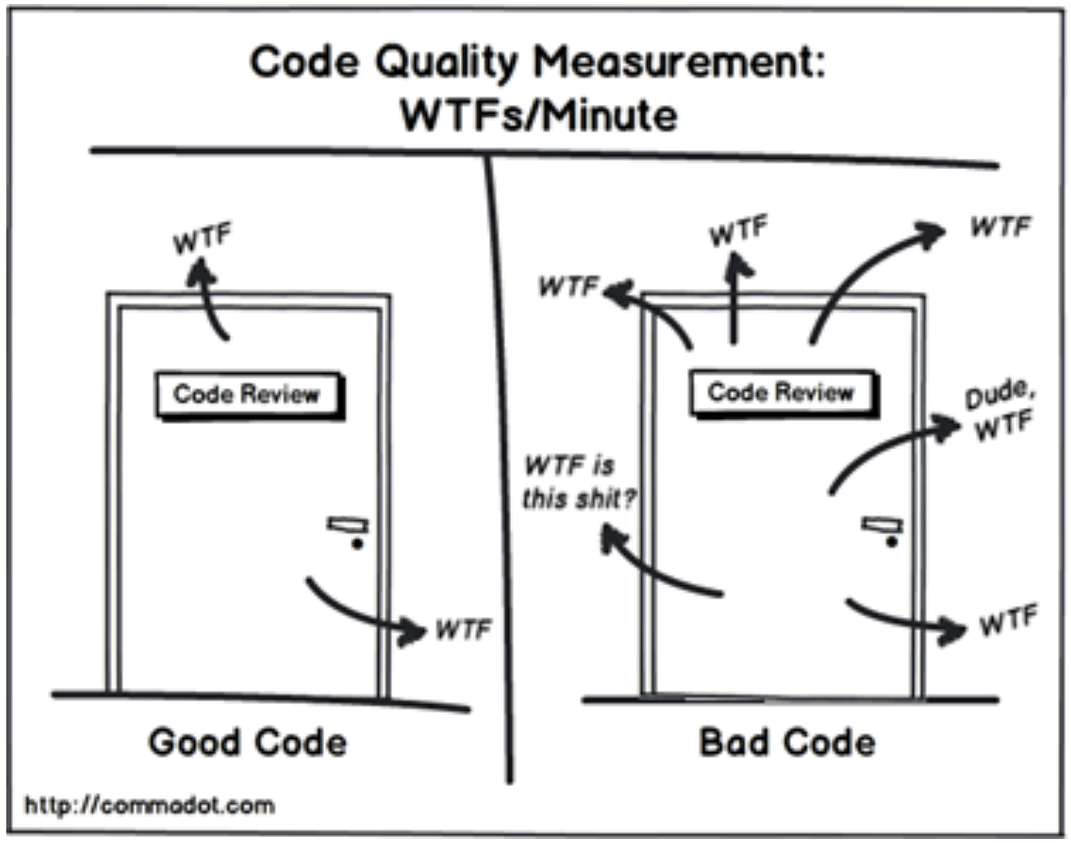
UNIVERSITAT DE BARCELONA

Code Quality Assessment Techniques

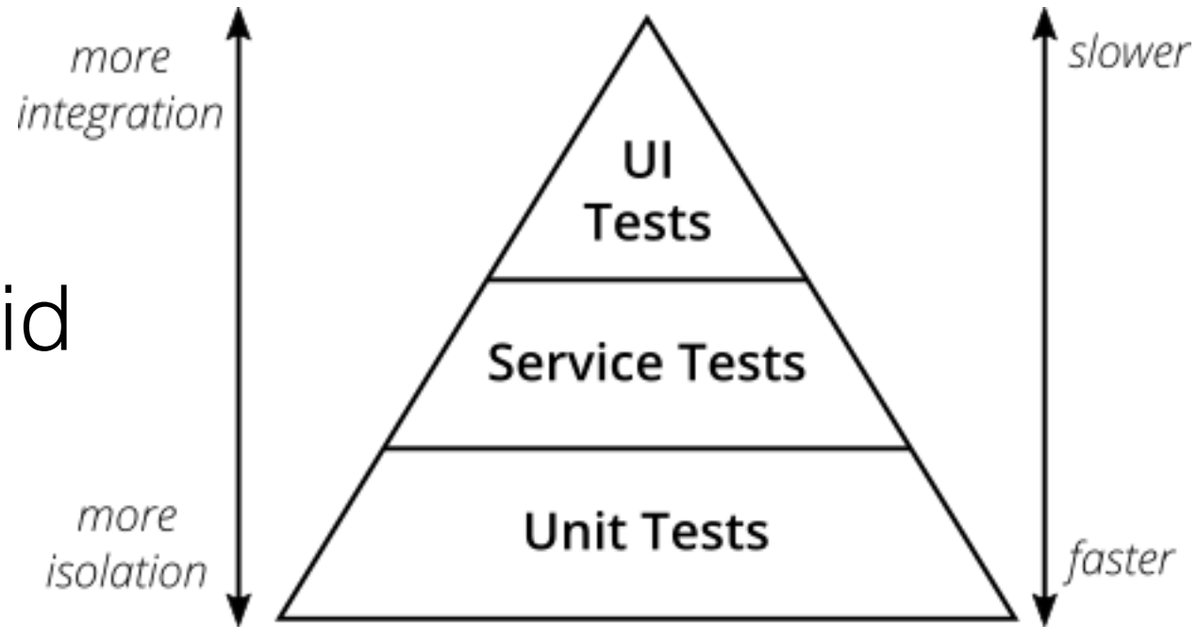


by Eloi Puertas

Levels of **Software Analysis**

- Unit Level: Analysis that takes place within a specific program or subroutine, without connecting to the context of that program.
- Technology Level: Analysis that takes into account interactions between unit programs
- System Level: Analysis that takes into account the interactions between unit programs, but without being limited to one specific technology or programming language.
- Mission/Business Level: Analysis that takes into account the business/mission layer terms.

Test Pyramid



Testing Best Practices

- Testing should be a collaboration of Development, QA, and Operations

- Tests should be logically-componentized, incremental, and repeatable; results must be deterministic and meaningful

- All tests need to be run at some point in the build pipeline, but not all tests need be run all the time.

- Eliminate test data and environment constraints so that tests can run constantly and consistently in production-like environments

- Teams should emphasize API Testing over GUI testing

What is **Static Code Review**

- **Code Review** (AKA Peer Review) (By humans)
- **Static Code Analysis** (By QA software tools)

What is **Code Review**

- One or several people **check**
  a program mainly by **viewing** and **reading**
  parts of its source code, **without** executing it.
  The reviewers are not the code's authors.

Goals in **Code Review**

- Better code quality: Improve internal code quality
- Finding defects: Improve quality, performance, security…
- Learning/Knowledge transfer.
- Increase sense of mutual responsibility
- Finding better solutions
- Complying to QA guidelines. Code reviewers are mandatory in some contexts.

What is **Static Code Analysis**

Code Review performed by an automated tool:

- Highlighting possible coding errors (e.g., the **lint** tool)
- **Formal methods** that mathematically prove properties about a given program

List of SCA softwares in wikipedia

What is **Dynamic Code Analysis**

Is the analysis performed on programs while they are executing.

- **Code Coverage:** How much source code is executed by testing code (gcov, coveralls)
- **Memory Error detection:** memory leaks (valgrind)
- **Security Analysis:** security leaks (pentesting…)
- **Concurrency Errors:** race conditions, deadlocks…
- **Performance Analysis (Profiling).** Measures the **space** or **time** complexity of a program, the **usage** of particular instructions or the **frequency** and **duration** of function calls. It serves for program optimization. **List of profiling programs in wikipedia**

SW Quality Assurance (QA) techniques table

Technique	WHO	HOW	WHEN
Self Check	Code Author	Debugs Asserts	When Developing
Testing	Code Author	Unit Tests	When Developing
Pair Programming	Two Developers	One writes code, the other reviews	When Developing, switch roles often
Code Review	Not the code Author	Reading the code	Reviewing phase, after Developing
Static Code Analysis	Automatic process	Using a QA software tool	Before Integration in the CI/CD
Dynamic Code Analysis	QA team	Using QA tools	After Integration in the CI/CD

Web application testing

- Web application Testing

QA and Branching

- Source code branching strategy and DevSecOps Pipeline at different stages of CI/CD

Final remark.

"Finding all possible run-time errors in an arbitrary program (or more generally any kind of violation of a specification on the final result of a program) is ***undecidable****: there is no mechanical method that can always answer truthfully whether an arbitrary program may or may not exhibit runtime errors."*

- Church, Gödel and Turing (1936)