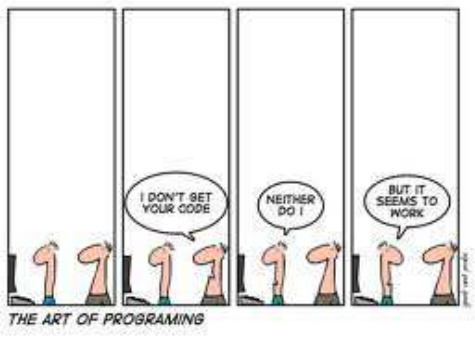


Software Development II

www.cs.uoi.gr/~zarras/soft-devII.htm

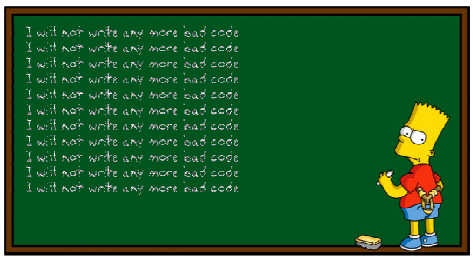
from Clean Code by R. C. Martin, a.k.a "Uncle Bob"

What are the general objectives of the course?

- Identify issues of poor software design/implementation.
- Improve the quality of software that suffers from issues of poor software design/implementation by applying good practices, refactoring, patterns.

What will we study ?

- Clean Code Rules
  - Meaningful Names
  - Comments
  - Formatting
  - Functions/Methods
  - Error Handling
  - Classes



What will we study ?

- Code Smells
  - Long Method
  - God Class
  - Long Param List
  - Primitive Obsession
  - Data Container
  - Duplication
  - Feature Envy
  - Shotgun Surgery
  - Message Chains
  - Middle Man
  - ……..

- Refactoring Techniques
  - Composing Methods
  - Simplifying Conditionals
  - Moving Features
  - Generalization
  - Simplifying Method Calls

Readings

- Αντικειμενοστρεφής Σχεδίαση, Α. Χατζηγεωργίου
- Ανάπτυξη Προγραμμάτων σε Java, B. Liskov, J Guttag

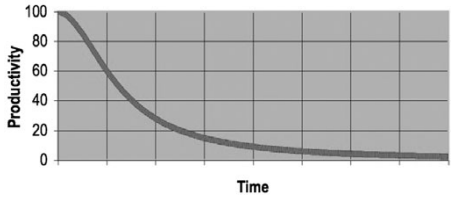
- Clean Code - A Handbook of Agile Software Craftsmanship, R. C. Martin
- Refactoring - Improving the Design of Existing Code, M.Fowler

Project, Final Exam & Grades

- We will develop a project in Java in groups of up to 2 person.
  - Part I: Review and analyze the problems of an existing software
  - Part II: Refactoring and extension of the software
- We will have a final examination at the end of the semester

- Final Grade
  - if (exam >= 5 and project >=5) then
    - grade = 0.7 * exam + 0.3 * project
  - else
    - grade = min(exam, project)

Bad Code

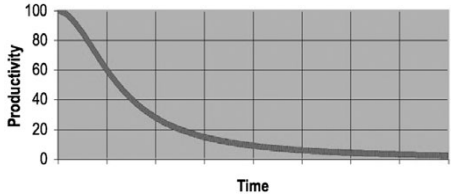


The total cost of owning a mess:

Over the span of a year or two, teams that were moving very fast at the beginning of a project can find themselves moving at a snail's pace.

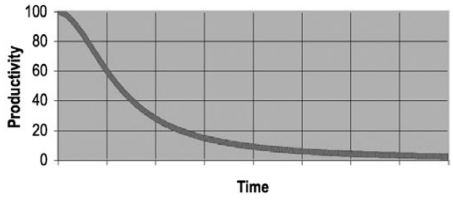
As the mess builds, the productivity of the team continues to decrease, asymptotically approaching zero.

Bad Code



What would you do If you were the manager of such project ??

Bad Code

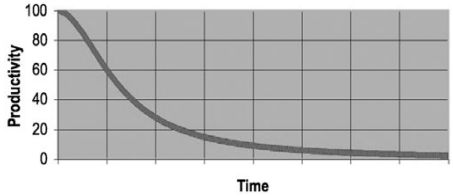


A managerial solution:

As productivity decreases, management does the only thing they can; they add more staff to the project in hope of increasing productivity.

Would that work ??

Bad Code



A managerial solution:

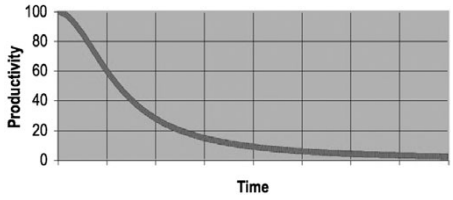
As productivity decreases, management does the only thing they can; they add more staff to the project in hopes of increasing productivity.

But that new staff is not versed in the design of the system.

They, and everyone else on the team, are under horrific **pressure to increase productivity.**

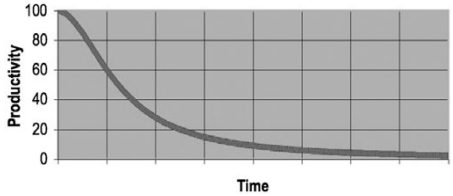
So they all make more and more messes (!!!!!), driving the productivity ever further toward zero.

Bad Code



What would you do if you were a developer in such a project ??

Bad Code



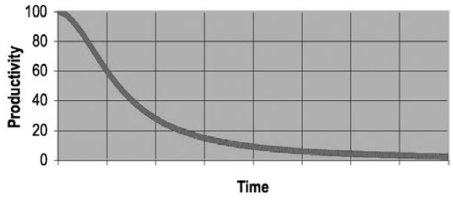
A technical solution:

Eventually the team rebels. They demand a redesign. Management eventually bend to the demands of the developers and authorize the grand redesign in the sky.

A new tiger team is selected. Everyone else must continue to maintain the current system.

Would that work ??

Bad Code



A technical solution:

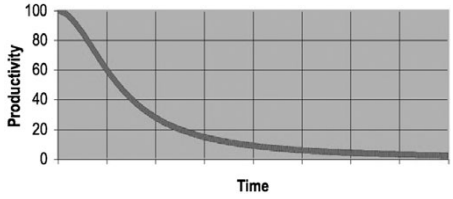
Now the two teams are in a race.

The tiger team must build a new system that does everything that the old system does.

Not only that, they have to keep up with the changes that are continuously being made to the old system !!!!

And by the time it's done, the tiger team members are demanding that the new system be redesigned because it's such a mess !!!!

Bad Code



The moral:

Spending time keeping your code clean is not just cost effective; it's a matter of professional survival !!!!

Bad Code

Why does good code turn into bad code ???

Bad Code

Why does good code turn into bad code ???

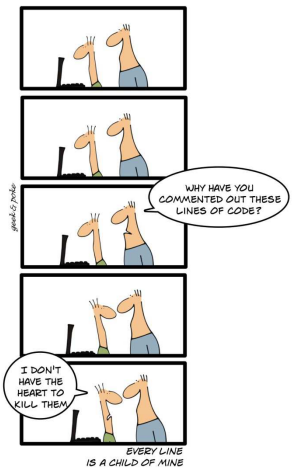
All developers with more than a few years experience know that previous messes slow them down.

And yet all developers feel the pressure to make messes in order to meet deadlines !!!

In practice, …. the mess will slow you down instantly, and will force you to miss the deadline. The *only way to make the deadline—the only way to* go fast—is to keep the code as clean as possible at all times.

Clean Code

What is clean code ???



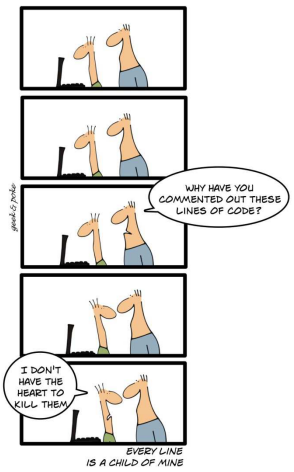
Clean Code

What is clean code ???

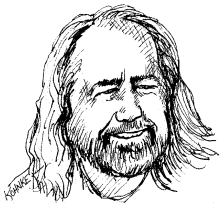
Probably as many definitions as programmers !!!

But, still the baseline is the same,

code is clean if its can be easily understood, tested and maintained….



Clean Code



Grady Booch, author of *Object Oriented Analysis and Design with Applications*

*Clean code is simple and direct. Clean code **reads** like well-written prose….*

Clean Code

Bjarne Stroustrup, inventor of C++ and author of *The C++ Programming Language*

…elegant and efficient.

The logic should be straightforward to make it hard for bugs to hide,

the dependencies minimal to ease maintenance, error handling complete according to an articulated strategy, and performance close to optimal so as not to tempt people to make the code messy with unprincipled optimizations.

Clean code does one thing well.

Clean Code

Dave Thomas, godfather of the Eclipse strategy

*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.…

Clean Code

Michael Feathers, author of *Working Effectively with Legacy Code*

….. *Clean code always looks like it was written by someone who cares*….

Clean Code

Ron Jeffries, author of *Extreme Programming Installed* and *Extreme Programming Adventures in C#*

Simple code:
- *Runs all the tests;*
- *Contains no duplication;*
- *Expresses all the design ideas that are in the system;*
- *Minimizes the number of entities such as classes, methods, functions, and the like.*

Clean Code

Ward Cunningham, motive force behind Design Patterns. The godfather of all those who care about code.

You know you are working on clean code when each routine you read turns out to be pretty much what you expected….

Clean Code

R. C. Martin, a.k.a. "Uncle Bob"

The next time you write a line of code, remember you are an author, writing for readers who will judge your effort.

It's not enough to write the code well. The code has to be *kept clean over time. We've all* seen code rot and degrade as time passes.

The Boy Scouts have a simple rule that we can apply to our profession.

Leave the campground cleaner than you found it.