ACADEMIA DE STUDII ECONOMICE DIN BUCUREȘTI
FACULTATEA DE CIBERNETICĂ, STATISTICĂ ȘI INFORMATICĂ ECONOMICĂ

Calitate și Testare Software

Conf. univ. dr. Bogdan IANCU

Avertizare

Utilizarea acestui material de curs ca resursă unică de învățare a acestei discipline reprezintă o abordare superficială

Să ne cunoaștem

- Programez din 2003, predau din 2012
- Am lucrat peste 15 ani ca programator în diverse companii multinaționale
- Trainer acreditat de peste 10 ani
- Email: **bogdan.iancu@ie.ase.ro**

Obiective

- Prezentarea noțiunilor standard din industrie ce pot duce la creșterea calității codului sursă și la o mentenabilitate crescută a acestuia
- Însușirea aspectelor practice și teoretice legate de clean code, design patterns și testare unitară

Evaluare

Seminar - 50% din nota finală

- test practic Clean Code & DP Creaționale (săptămânile 6-7) - 20% din nota finală
- test practic DP (săptămânile 10-11) - 20% din nota finală
- activitate de seminar - 10% din nota finală

Examen - 50% din nota finală

- sunt obligatorii minim 2,5 puncte din 5 pentru a putea susține examenul în sesiunea normală
- Nota minimă din examen pentru a putea promova este 5

Structura disciplinei

1. **Clean Code**

   Principii de scriere a codului sursă

2. **Soluții de versionare**

   Git, SVN

3. **Design Patterns**

   Creaționale, structurale comportamentale

4. **Testare unitară**

   Unit testing - JUnit

Precondiții

- Programare Orientată Obiect (în C++)
- Programare Multiparadigmă Java
- Programarea Aplicațiilor Windows (C#)

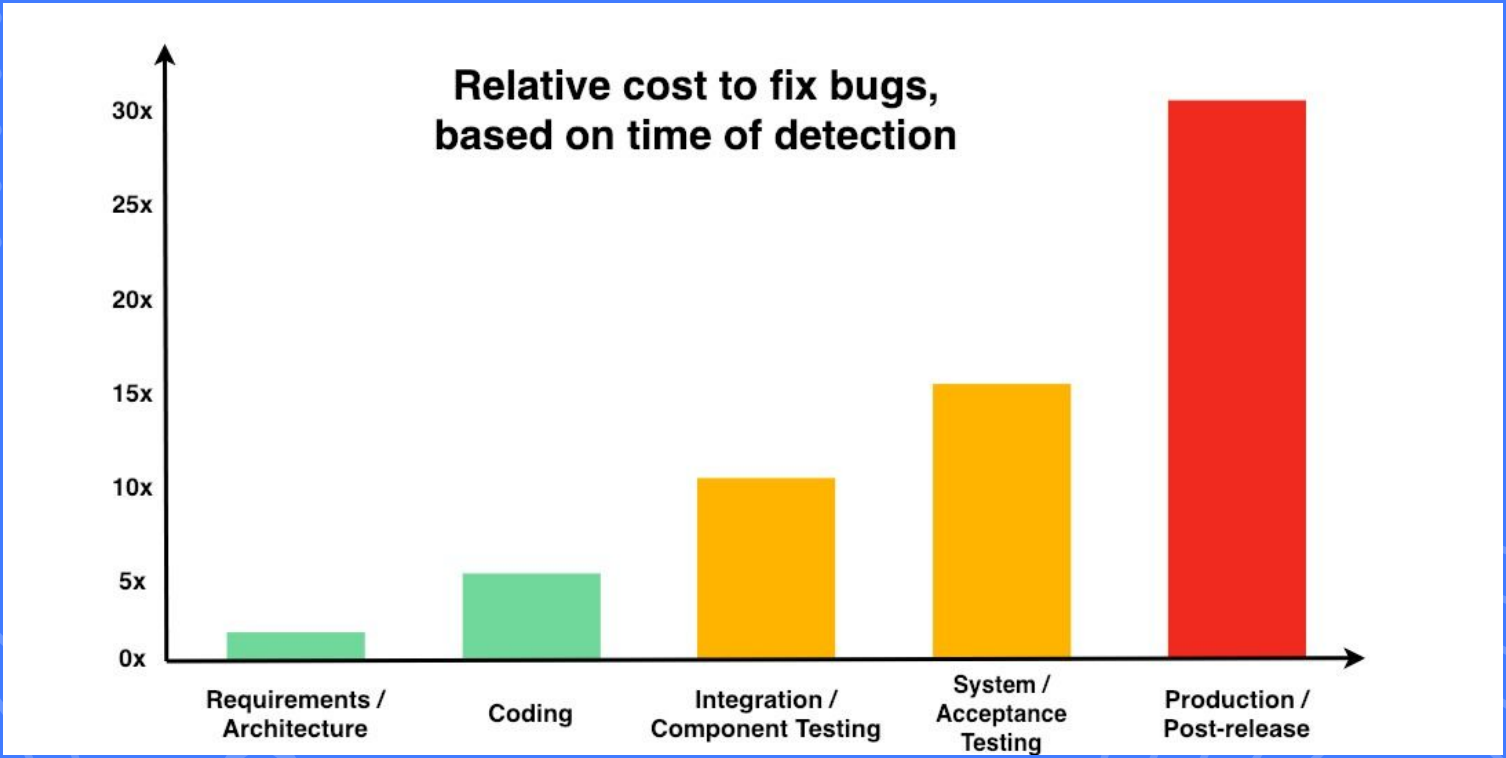
Bibliografie

- Madalina Zurini, Alin Zamfiroiu - Calitate si testare software. Studii de caz, Editura ASE, 2017
- Robert C. Martin - Clean Code, A Handbook of Agile Software Craftsmanship, Prentice Hall, 2009
- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides - Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995
- Eric Freeman, Elisabeth Robson - Head First Design Patterns, O'Reilly Media, 2020
- Steve Holzner - Design Patterns for Dummies, Wiley, 2006
- Lasse Koskela - Effective Unit Testing, Manning, 2013
- Lasse Koskela - Practical TDD and Acceptance TDD for Java Developers, Manning, 2007
- Alasdair Allan - Pragmatic Unit Testing in Java with JUnit, The Pragmatic Programmers, 2004
- Scott Chacon, Bean Straub - Pro Git, 2nd edition, Apress, 2014, disponibila online la adresa http://git-scm.com/book/en/v2
- http://git-scm.com/docs

Clean Code

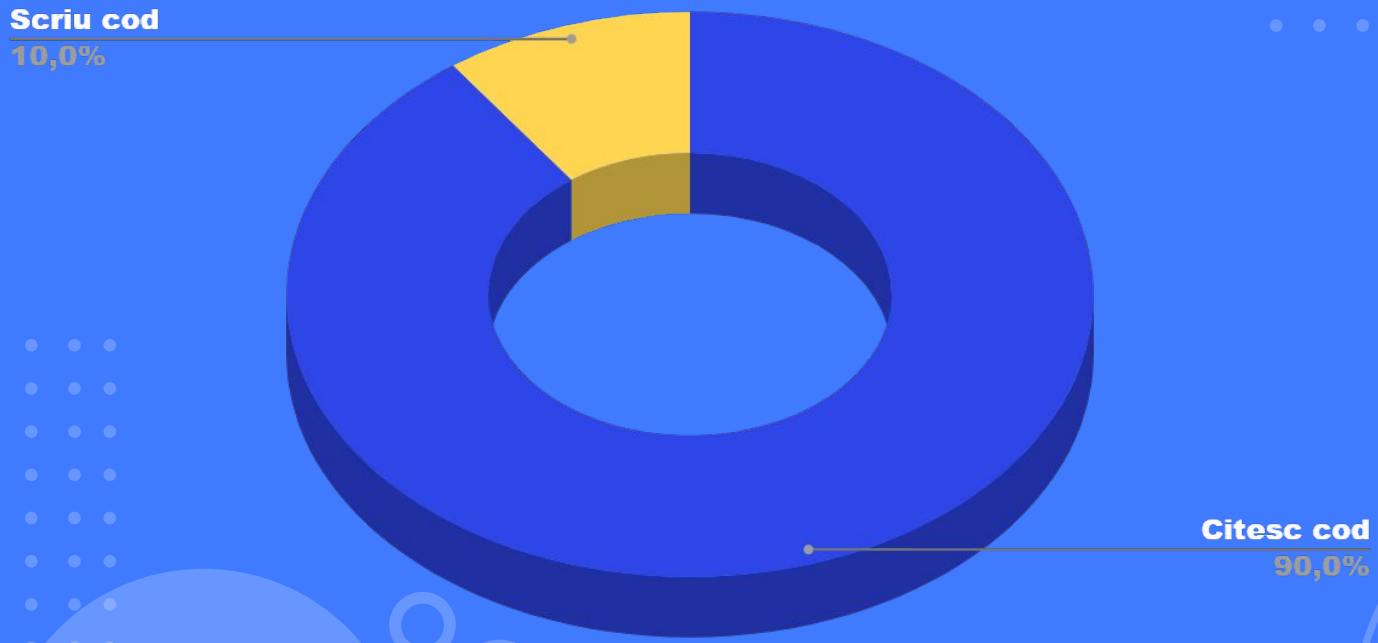
De ce este mai important felul în care scriem decât ceea ce scriem?

Costul rezolvării unui bug



Sursa: https://assets.deepsource.io/569f19f/images/blog/cost-of-fixing-bugs/chart.jpg

Ce fac programatorii la serviciu



De ce Clean Code?

- Programarea nu constă în a spune computerului ce să facă
- Programarea constă în a spune altui om ce vrem să facă un computer
- Din păcate câteodată acel „alt om” suntem chiar noi
- Până la urmă suntem autori
- Nu avem timp să fim leneși

Creierul nostru nu este un computer

- Când citim cod creierul nostru încearcă să joace rolul unui compilator
- Conform studiilor oamenii pot reține simultan doar 7 elemente (±2) în memorie
- Rubber Duck Programming (Debugging)

Un cod clean (curat) trebuie

- să fie ușor de citit
- să fie ușor de înțeles
- să fie ușor de modificat
- ... de către noi sau oricine altcineva

Principii de clean code

- DRY
- KISS
- YAGNI
- SOLID