

Software testing

(there is more than Unit tests)

Ing. Jiří Kiml,
21/11/2012, ZCU

Agenda

- ▶ **Motivation**
- ▶ **Defects**
- ▶ **Testing methods**
- ▶ **Testing levels**
 - SYSTEM TESTING
- ▶ **Best practices**
- ▶ **Summary**
- ▶ **Questions**

Motivation

*Programming is similar to a game of golf.
The point is not getting the ball in the hole
but how many strokes it takes.*

~Harlan Mills

- ▶ **We have Unit tests! Is more testing necessary?**
- ▶ **Can we prevent defects? NO.**
- ▶ **How much does it cost to fix a defect ?**

Agenda

- ▶ Motivation
- ▶ Defects
- ▶ Testing methods
- ▶ Testing levels
 - SYSTEM TESTING
- ▶ Best practices
- ▶ Summary
- ▶ Questions

What is defect?

- ▶ is an error, flaw, mistake, failure, or fault in a computer program or system that produces an **incorrect or unexpected result**, or causes it **to behave in unintended ways**.
- ▶ caused by coding errors
- ▶ caused by requirement gaps
- ▶ non-functional requirements such as testability, scalability, maintainability, usability, performance, and security

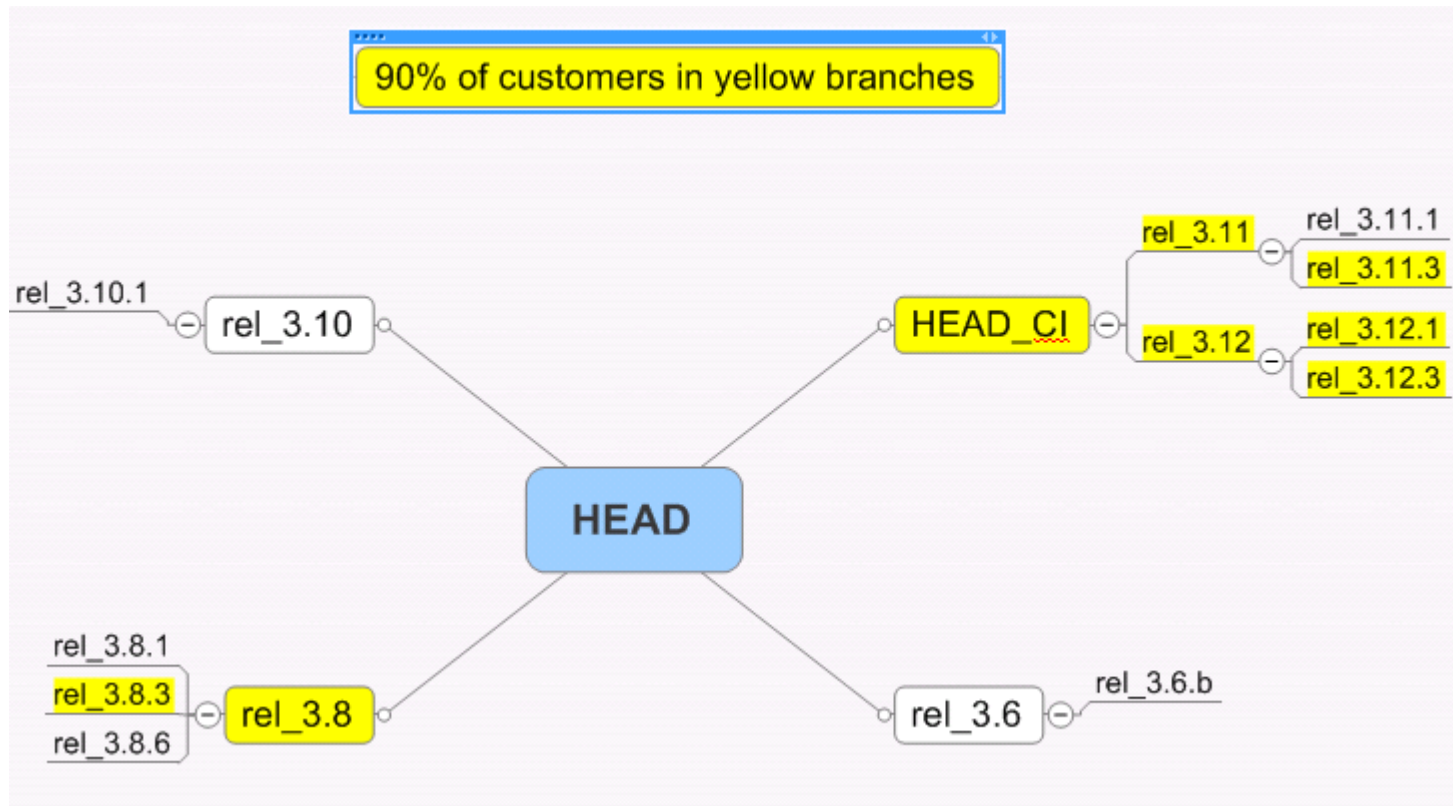


MPA – medical process assistant

- ▶ **16 active branches**
- ▶ **34.000 classes**
- ▶ **250 eclipse projects**
- ▶ **About 50 programmers**
- ▶ **12 years old (jdk 1.1)**
- ▶ **More than 100 hospitals**
- ▶ **Austria/Germany/Czech Republic/Serbia**
- ▶ **<http://www.systema.info/en/solutions/mpa-medical-process-assistant/>**

*Programming is like sex. One mistake
and you have to support it for the rest of your life.*
~Michael Sinz

MPA branches



Agenda

- ▶ Motivation
- ▶ Defects
- ▶ **Testing methods**
- ▶ **Testing levels**
 - SYSTEM TESTING
- ▶ Best practices
- ▶ Summary
- ▶ Questions

Testing methods

- ▶ **Static versus dynamic testing**
 - **Static: reviews (various types)**
 - **Dynamic: execution of test cases**
- ▶ **The box approach**
 - **White-box**
 - **Black-box**
 - **Grey-box**

Agenda

- ▶ Motivation
- ▶ Defects
- ▶ Testing methods
- ▶ **Testing levels**
 - SYSTEM TESTING
- ▶ Best practices
- ▶ Summary
- ▶ Questions

Testing levels

- ▶ **Unit testing**
- ▶ **Integration testing**
- ▶ **System testing**
- ▶ **Acceptance testing**

Unit testing

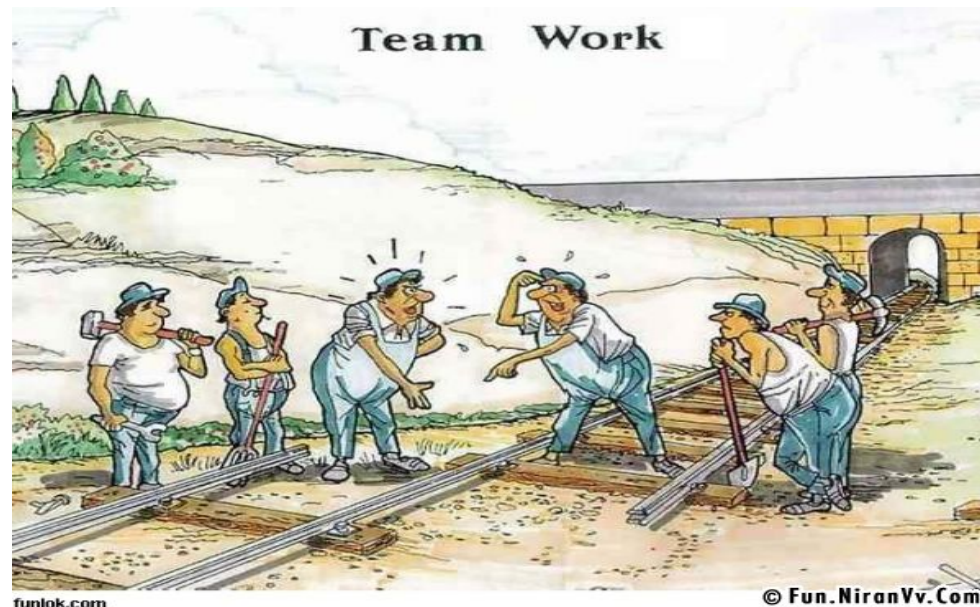
- ▶ **isolate individual part of the program and show that it is correct**
- ▶ **Mocks, stubs**
- ▶ **Supports good api design**
- ▶ **Documentation**
- ▶ **Safe refactorings**
- ▶ **will not catch integration errors or broader system-level errors**

Wait! I get it!
You think about how
the module will be used
because the Unit Test
has to use the module.



Integration testing

- ▶ the phase in which modules are combined and tested as a group.
- ▶ Dedicated environment is typically needed

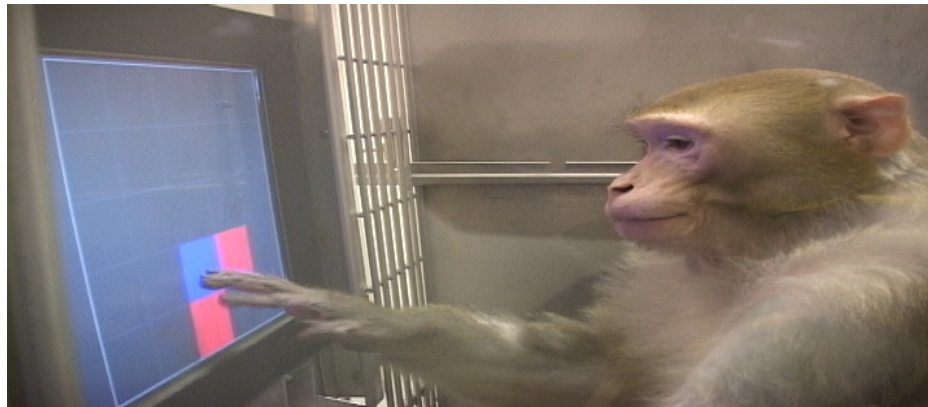


Agenda

- ▶ Motivation
- ▶ Defects
- ▶ Testing methods
- ▶ Testing levels
 - SYSTEM TESTING
- ▶ Best practices
- ▶ Summary
- ▶ Questions

System testing

- ▶ **System testing tests a completely integrated system to verify that it meets its requirements.**
- ▶ **Typically black box testing**



System testing - types

Graphical user interface testing

Usability testing

Software performance testing

Compatibility testing

Exception handling Load testing

Volume testing Stress testing

Security testing

Sanity testing Smoke testing

Exploratory testing Ad hoc testing

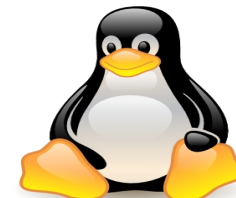
Regression testing Installation testing

Maintenance testing

Recovery testing and failover testing.

Accessibility testing

Graphical user interface testing



Performance testing

- ▶ **how a system performs in terms of responsiveness and stability under a particular workload**
- ▶ **goal is not to find bugs, but to eliminate bottlenecks and establish a baseline for future regression testing**



Load testing

- ▶ **to determine a system's behaviour under both normal and anticipated peak load conditions**



Stress testing

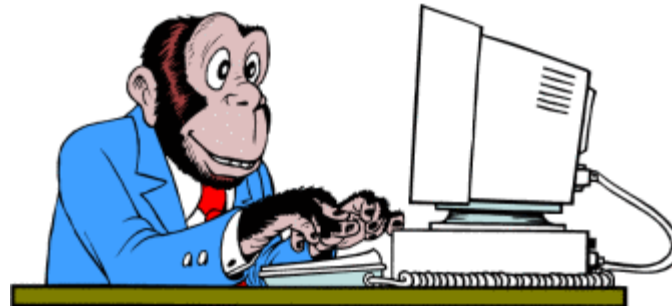
- ▶ **testing beyond normal operational capacity, often to a breaking point, in order to observe the results**



Regression testing

- ▶ **Should ensure that a changes, such as bugfixes and enhancements, did not introduce new faults**

Always Remember



***We Could Hire A Trained Monkey
To Do Your Job!***

Recovery and failover testing

- ▶ **how well an application is able to recover from crashes, hardware failures and other similar problems**

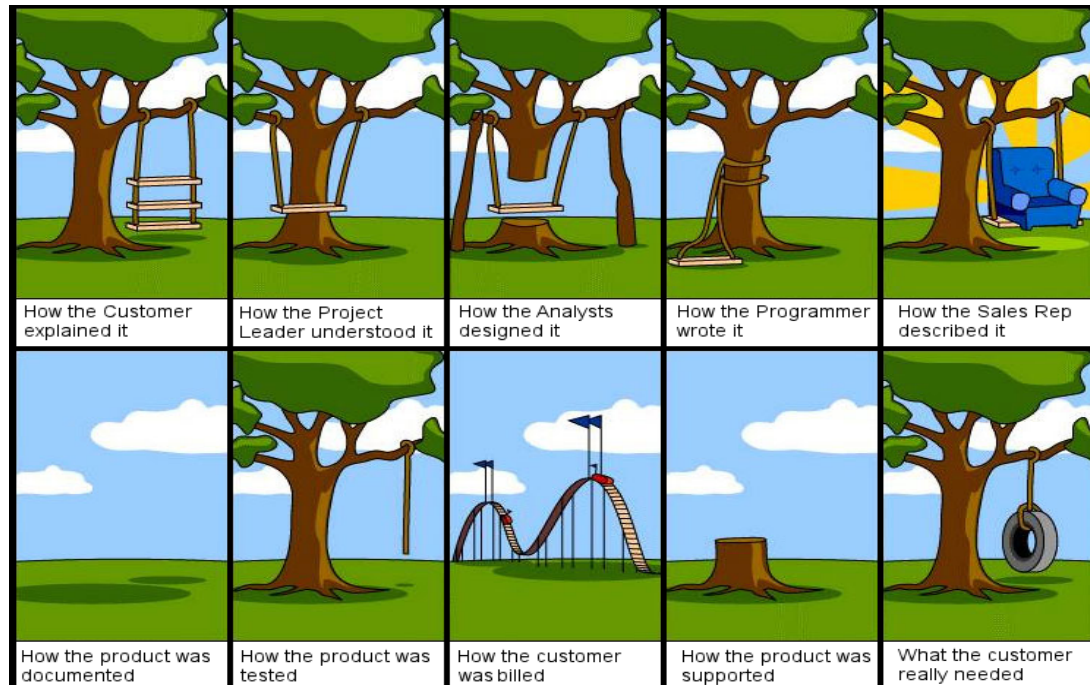


Agenda

- ▶ Motivation
- ▶ Defects
- ▶ Testing methods
- ▶ **Testing levels**
 - SYSTEM TESTING
- ▶ Best practices
- ▶ Summary
- ▶ Questions

Acceptance testing

- ▶ Done by user/customer
- ▶ to determine if the requirements of a specification or contract are met



Agenda

- ▶ Motivation
- ▶ Defects
- ▶ Testing methods
- ▶ Testing levels
 - SYSTEM TESTING
- ▶ **Best practices**
- ▶ Summary
- ▶ Questions

Best Practices

- ▶ **Unit test to support development/refactoring**
- ▶ **Automatic integration tests**
- ▶ **Automatic regression tests (basic usecases)**
- ▶ **Automatic acceptance tests**
- ▶ **Semiautomatic/manual tests before release upgrade**
 - **load/stress tests**
 - **recovery and failover**

Agenda

- ▶ **Motivation**
- ▶ **Defects**
- ▶ **Testing methods**
- ▶ **Testing levels**
 - SYSTEM TESTING
- ▶ **Best practices**
- ▶ **Summary**
- ▶ **Questions**

Summary

- ▶ **How much does it cost to fix a defect ?**
- ▶ **Can we prevent bugs?**

The only way for errors to occur in a program is by being put there by the author. No other mechanisms are known. Programs can't acquire bugs by sitting around with other buggy programs.

~Harlan Mills

Agenda

- ▶ **Motivation**
- ▶ **Defects**
- ▶ **Testing methods**
- ▶ **Testing levels**
 - SYSTEM TESTING
- ▶ **Best practices**
- ▶ **Summary**
- ▶ **Questions**

Links

- ▶ http://en.wikipedia.org/wiki/Software_testing
- ▶ <http://jmeter.apache.org/>
- ▶ <http://www.ej-technologies.com/products/jprofiler/overview.html>

Questions ...

... and maybe answers



Thank you



Ing. Jiří Kiml,
21/11/2012, ZCU