Mutation testing

How good your tests really are?

Marcin Zajączkowski

DEVCONF.CZ
DEFINE FUTURE

Brno, 2014-02-08

About me

Java architect TDD practitioner
Team mentor Clean code developer
Software Craftsmanship Evangelist
Fedora packager FOSS developer Linux enthusiast
IT Trainer Code quality freak Blogger

Presentation plan

- What is and how works mutation testing?
- Why is almost not known and rarery used?
- PIT – tool which works
- Mutants in action – live coding
- Benefits of using
- Adaptation in real life project

http://www.townofdarienny.com/

http://www.oldwestlawmansforgottenmemoir.com/memoir_bringsV13.html

WANTED

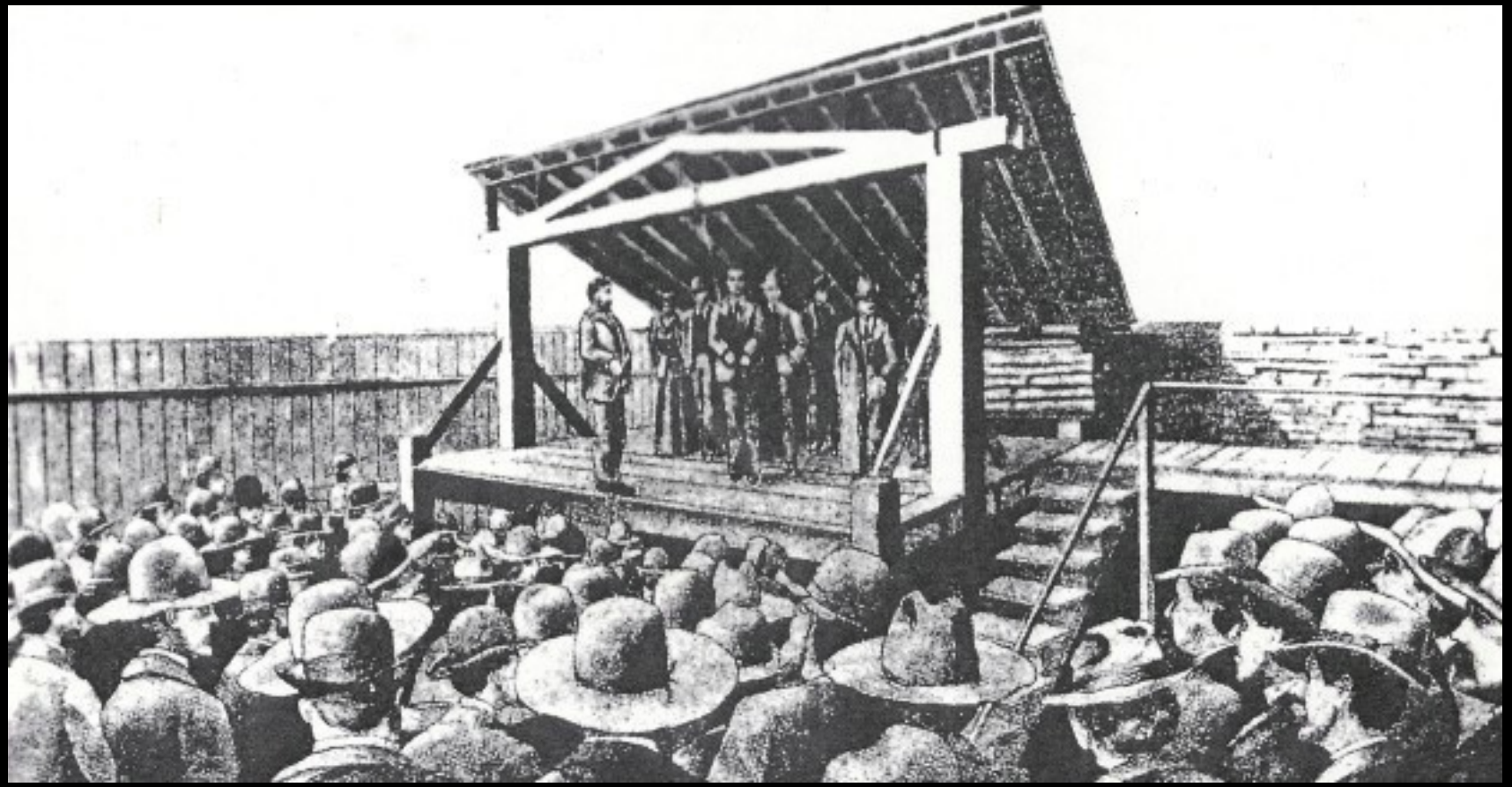
DEAD OR ALIVE

BILLY THE KID

WILLIAM BONNEY

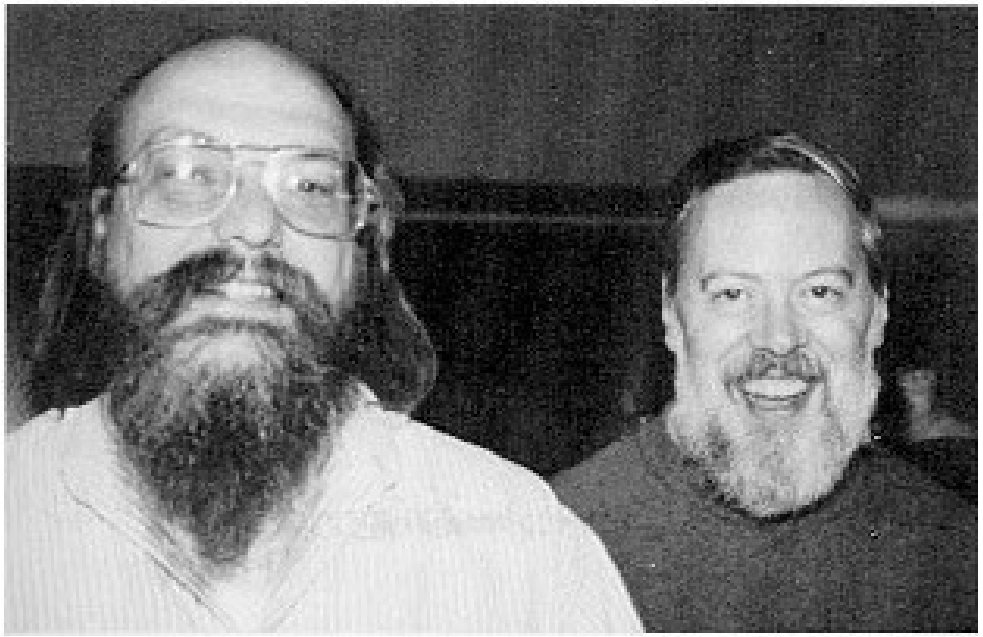
$5,000 REWARD

NOTIFY- Marshall Pat Garrett



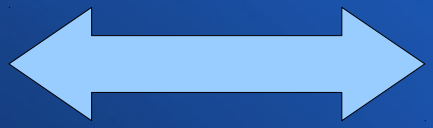
http://www.nps.gov/fosm/historyculture/executions-at-fort-smith-1873-to-1896.htm

https://secure.flickr.com/photos/7402220@N02/491093210/



http://publish.illinois.edu/libraryitnews/2012/06/

Analogies



Project	Town
Bugs in code	Crimes
Automated tests	Sheriffs
Code coverage	Patrol paths
Mutants in code	Provocations

The idea with an analogy to criminality and law enforcement taken from Chris Rimmer

Mutation testing
What's it about?

- Intentionally break selected line of production code (introduce a mutation)
- Check if any test detects a modification (if it fails)
- Survived mutation (which were not detected) are a potential bugs which would not be detected by automated tests

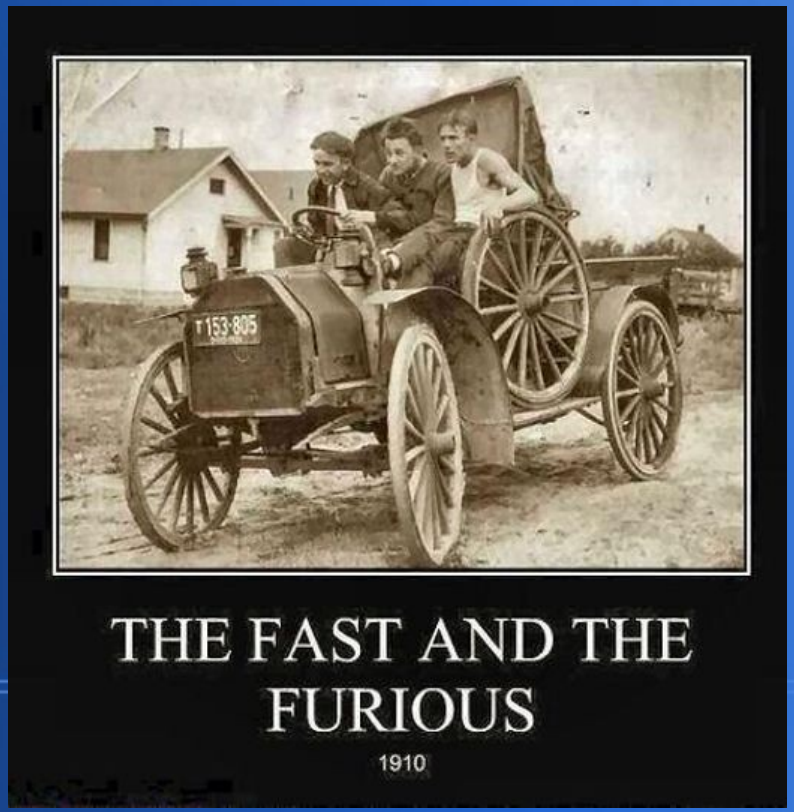
Common issues

- Small number of tools for Java (many not maintained anymore)
- Long execution time
- Required production code modification
- Infinite loops
- Stack overflow

http://bestclipartblog.com/27-tools-clip-art.html/tools-clip-art-2

PIT – mutations fast

- Bytecode manipulation
- Mutation of the lines with standard coverage only
- Execution of related test only
- Parallel execution
- Incremental analysis



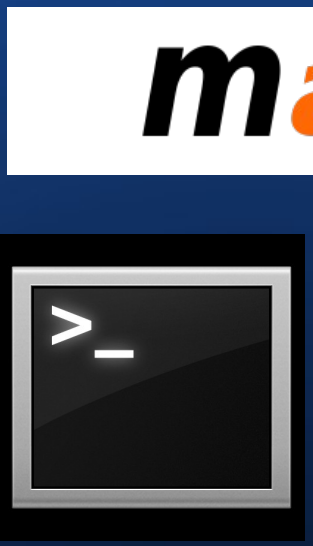
http://carhumor.net/blast-from-the-past/

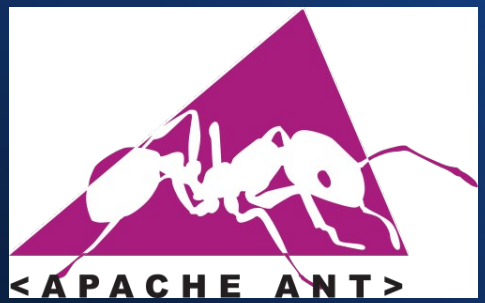
PIT – various mutations

- Conditionals Boundary Mutator
- Negate Conditionals Mutator
- Math Mutator
- Increments Mutator
- Invert Negatives Mutator
- Return Values Mutator
- (Non) Void Method Calls Mutator
- And more...

1 - http://blog.spoongraphics.co.uk/tutorials/create-a-cute-furry-vector-monster-in-illustrator
2 - http://blog.spoongraphics.co.uk/tutorials/create-a-cute-vector-monster-from-a-pencil-sketch

PIT – rich ecosystem





TestNG

Spock

Logos – home pages of mentioned projects

PIT – pros

- Fast
- Powerful
- Widely supported

Some Java alternatives

- Javalanche – small ecosystem,
  last commit in 2012
- µJava – limited access to source code
- Jester – no longer maintained
- Jumple – no longer maintained
- Judy – no longer maintained

Other languages
Selected tools

- MutPy – Python – actively developed
- Mutagenesis – PHP – actively developed (judgedim fork)
- Mutant – Ruby – actively developed
- NinjaTurtles - .NET – last commit in 2012

Mutants in action

http://www.adolescentadulthood.com/2013/01/23/how-did-the-teenage-mutant-ninja-turtles-get-their-names/

What can you get?

- Information how good your tests really are
- Places in code that are not properly tested
  - Better than with „normal” code coverage
- Better code quality
- Less bugs in production
- Job satisfaction
- ... (other benefits from writing testable code)

When to use?

- Greenfield project developed with high quality in mind
- High coverage, but still bugs in production which could (and should) be detected by tests
- Doubts about test suite quality
  - HLD requirement – 95% minimal code coverage level with the development team with no experience in automated testing
- Improve legacy system with low code quality and/or without tests

Prepare your project

- **Write automatic tests**
- Write fast automatic unit tests (not only slow integration ones)
- Separate fast unit tests from slow integration tests
  - Be able to run only selected group of tests

Does anyone use it
in the real life project?

- Yes :-)
- British Sky Broadcasting
- TheLadders
- Jumi
- Maybe you?

http://www.mysciencework.com/fr/MyScienceNews/10027/de-l-in-opportunite-des-open-spaces-dans-les-labos

Summary of benefits

Verification of effectiveness of automatic test

More reliable code

Less troubles at work

More time for interesting things

Increased job satisfaction

Questions?

Online feedback: http://devconf.cz/f/86

Thank you for your attention

Marcin Zajączkowski
m.zajaczkowski@gmail.com

http://blog.solidsoft.info/

@SolidSoftBlog