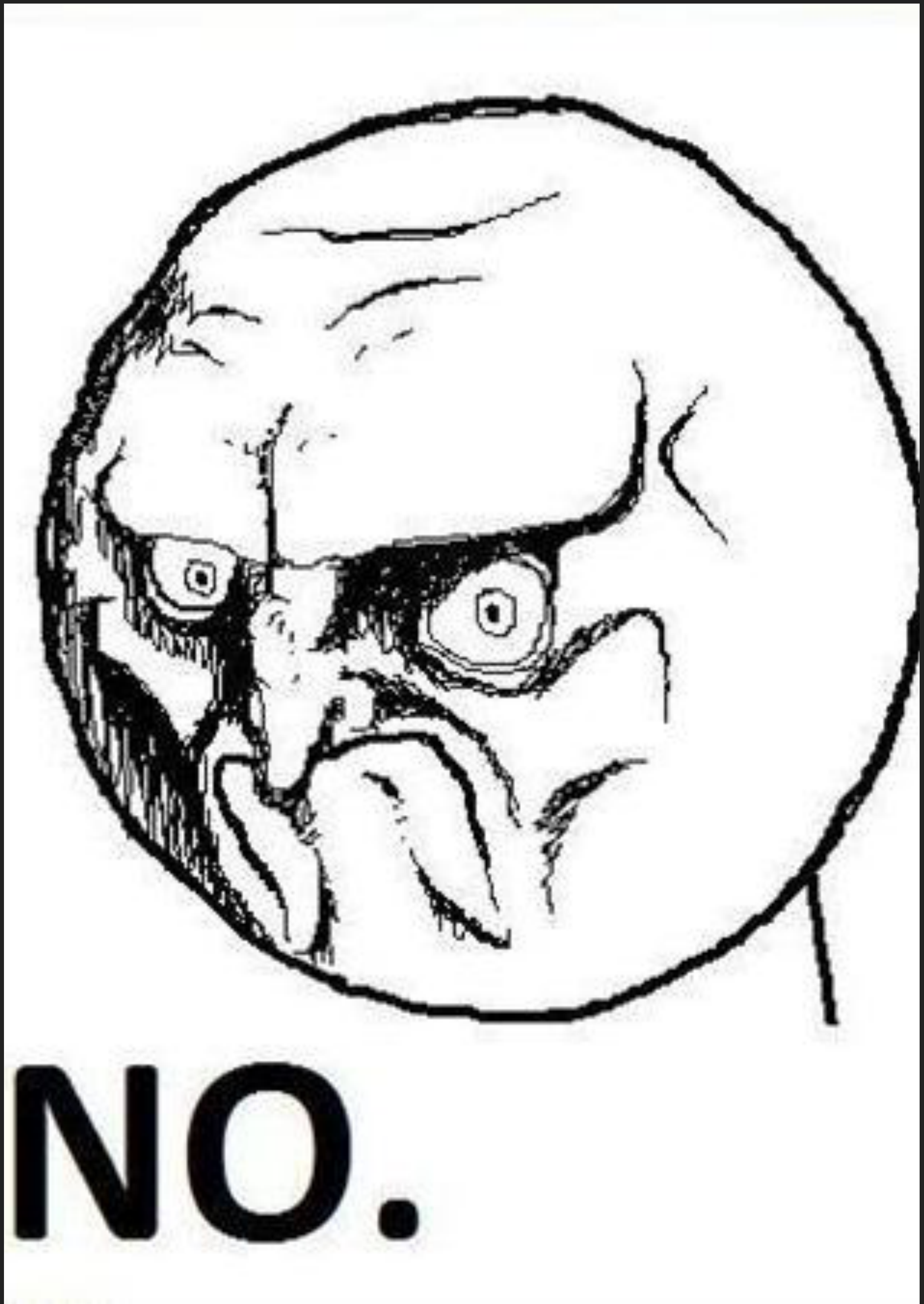
PRESENTED BY EUGENE

SOFTWARE TESTING

DISCLAIMER

- not an expert
- just a regular guy who is interested in the topic of testing
- feel free to challenge / correct me nicely and politely (don't be rude)



SOME THOUGHTS…

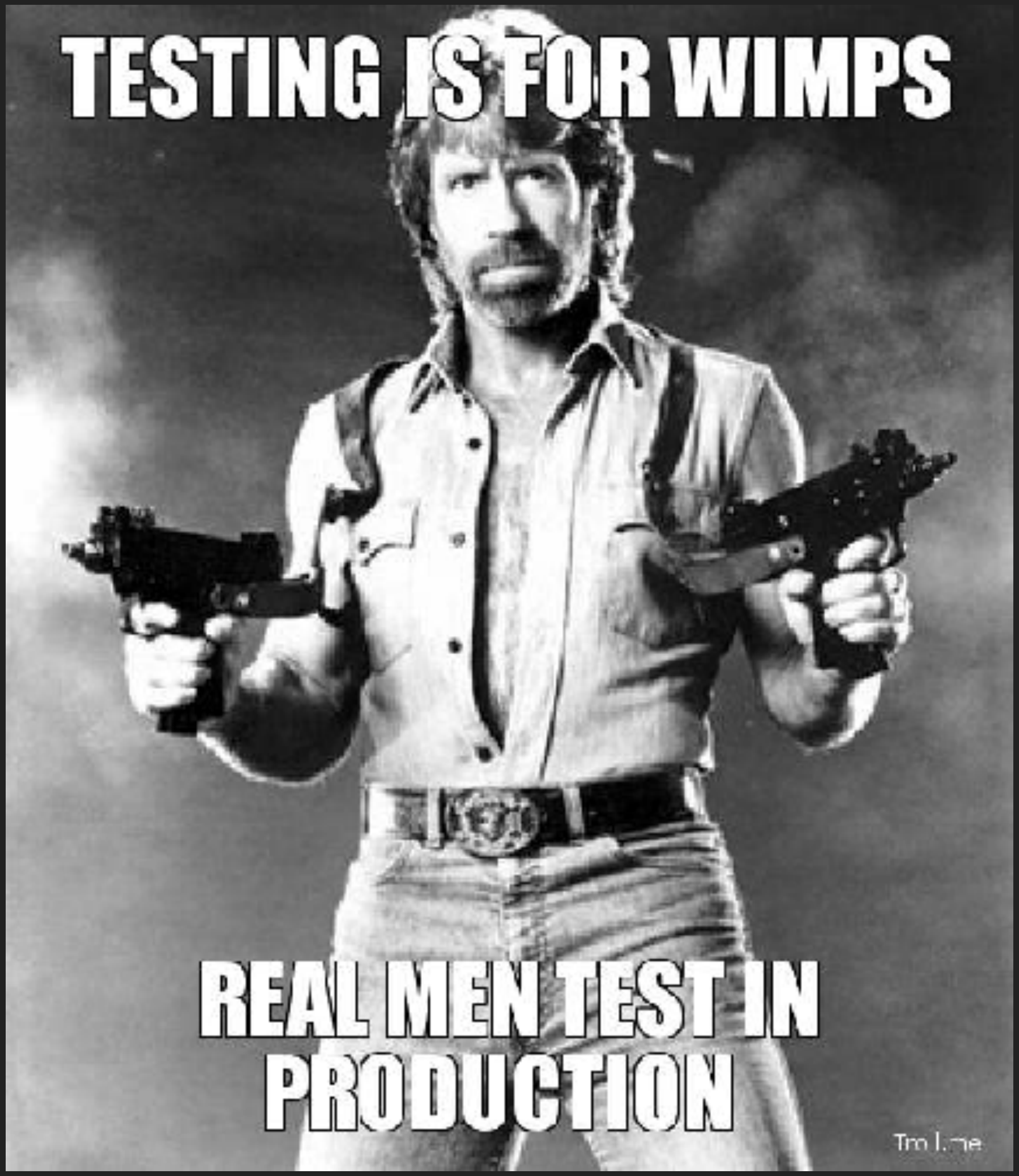
- testing is a huge topic
- testing can be boring/intimidating
- testing is important

OBJECTIVE

- share basic knowledge / concept of testing
- make sure everyone is on the same page
- cultivate good testing practices (in the future)

AGENDA

- 7 principles
- types of testing
- tools
- demo



7 PRINCIPLES

> TESTING SHOWS THE PRESENCE, NOT THE ABSENCE OF BUGS.

Edsger W. Dijkstra

1) TESTING SHOWS PRESENCE OF DEFECTS

- testing reduces the probability of undiscovered defects remaining in the software
- even if no defects are found, it is not a proof of correctness

EXHAUSTIVE TESTING IS IMPOSSIBLE

2) EXHAUSTIVE TESTING IS IMPOSSIBLE

- impossible to test all possible combinations of data and scenarios
- risks and priorities are used to concentrate on the most important aspects to test

EARLY TESTING

3) EARLY TESTING

- testing activities should start as early as possible and should be focused on defined objectives
- much cheaper to fix a defect in early stages of testing
- tdd?

DEFECT CLUSTERING

4) DEFECT CLUSTERING

- majority of the defects are caused by a small number of modules
- pareto principle: 80% of the problems are found in 20% of the modules

PESTICIDE PARADOX

5) PESTICIDE PARADOX

- if we keep running the same set of tests over and over again, chances are no more new defects will be discovered by those test cases
- it is very important to review the test cases regularly

TESTING IS CONTEXT DEPENDENT

6) TESTING IS CONTEXT DEPENDENT

- use different approach, methodologies, techniques and types of testing depending on the application type

ABSENCE OF ERRORS – FALLACY

7) ABSENCE OF ERRORS - FALLACY

- if the system built is unusable and does not fulfil the user's needs and expectations then finding and fixing defects will not help
- testing is not mere finding defects, but also to check that software addresses the business needs

SOFTWARE TESTING

- 2 keywords
  - verification
  - validation

VERIFICATION

- ensure that the product is being built according to the requirements and design specifications
- are we building the product *right*?

VALIDATION

- ensure that the product actually meets the user’s needs, and that the specifications were correct in the first place
- are we building the *right* product?

TYPES OF TESTING

- 2 basic types
  - white-box testing (verification)
  - black-box testing (validation)

WHITE-BOX TESTING

- unit testing
- integration testing
- regression testing

BLACK-BOX TESTING

- integration testing
- regression testing
- acceptance testing
- penetration testing
- stress testing
- usability testing

INTERESTED FOR MORE?

- list of 105 different testing types http://www.guru99.com/types-of-software-testing.html

TOOLS

- static code analysis / lint
- frameworks
- libraries

CODE ANALYSER

- sonarqube

LINT

- jslint
- jshint
- jscs
- eslint

JSLINT

- oldest
- not configurable
- not extensible
- shitty documentation

JSHINT

- configurable version of jslint (fork)
- not extensible
- hard to tell which rule causes error

TOOLS

JSCS

- only for coding style
- slowest

ESLINT

- newest
- configurable
- extensible
- good es6 and jsx support
- more rules than others
- easy to understand output

FRAMEWORKS

- jasmine
- mocha (usually with chai and sinon)
- karma
- nightwatch
- chakram
- gatling

LIBRARIES

- chai
- sinon
- request
- supertest
- istanbul
- rewire

ENOUGH TALK
SHOW ME A
DEMO!!!
memegenerator.net