

Test driven Non-Functionals?!

--- Combine Two Big Ideas and What do You Get? ---

Wilco Koorn
Senior consultant
Xebia IT Architects

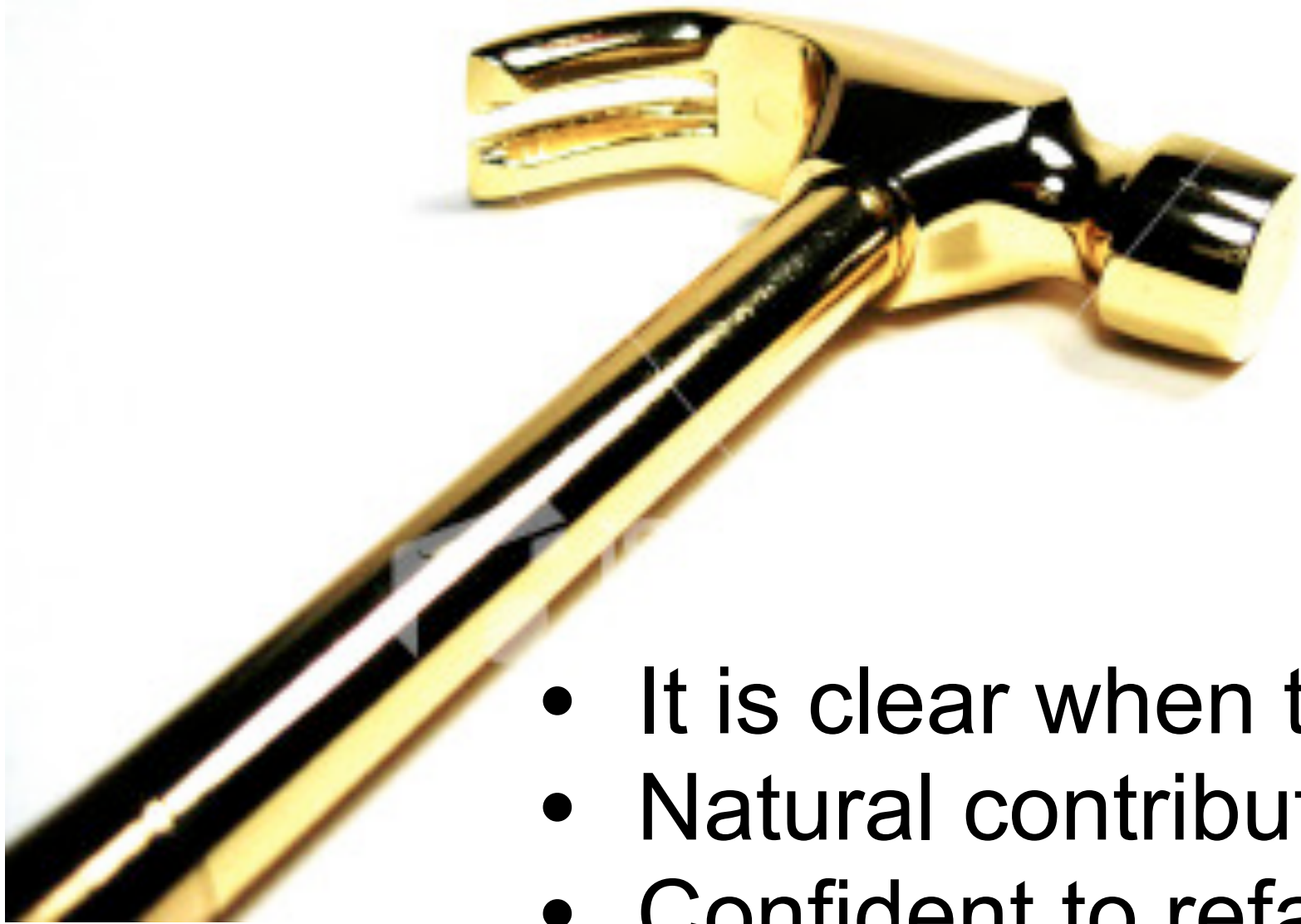


“Test driven development” is Common Practice in Java world

- Test a little, code a little, test, code, ...
- Tools: JUnit, TestNG, Fitnesse, ...
- Libraries: EasyMock, Mockito, ...
- Tests also used in regression testing
- Build never breaks



Advantages of “Test driven development”



- It is clear when the job is done
- Natural contribution to regression tests
- Confident to refactor
- Quality up front
- Testing not pushed against deadline

Test driven Non-Functionals...



Test driven Non-Functionals!



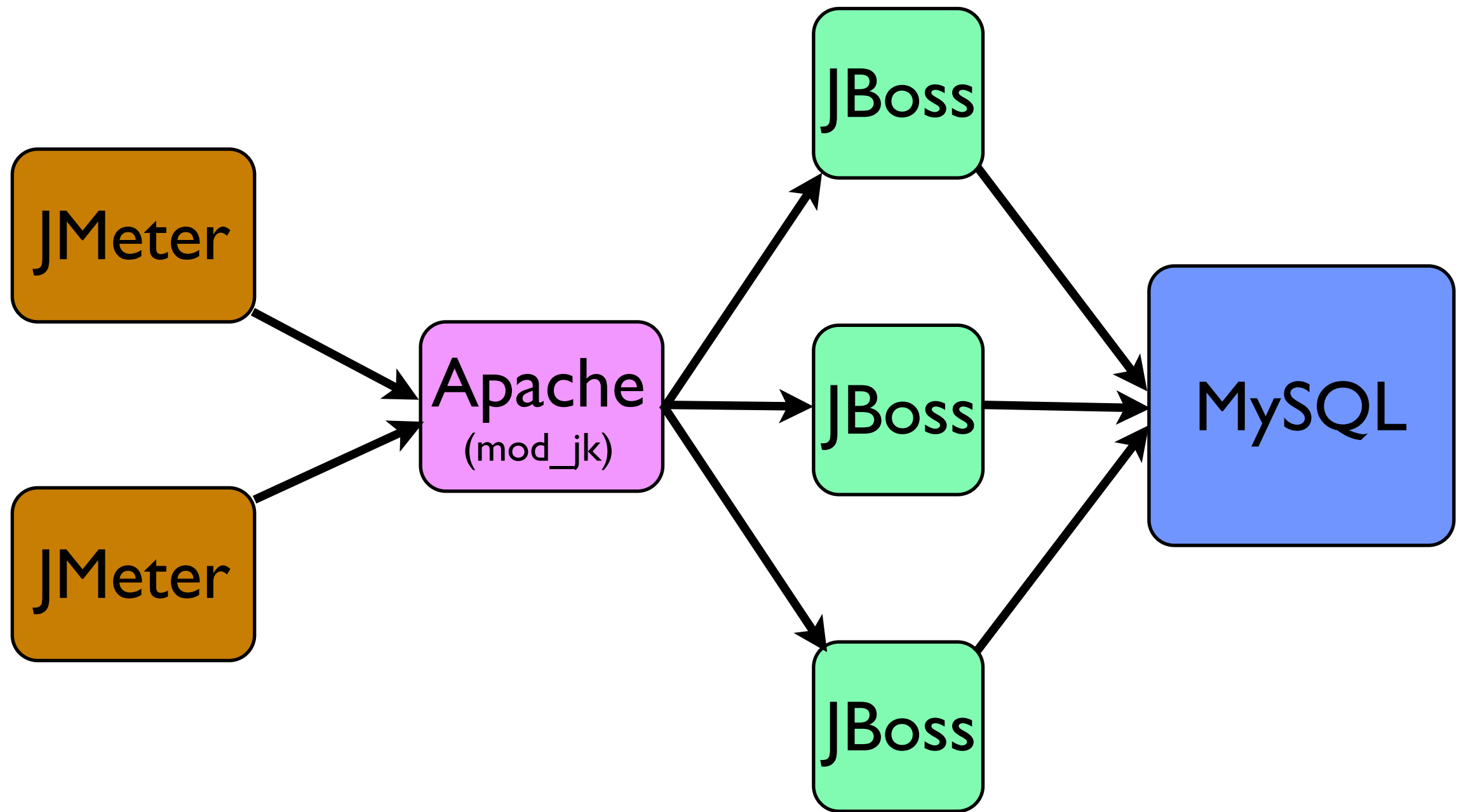
- Need for a scale
- Need for a test
- Tests also used in regression testing
- Tests are automated
- Build never breaks, but less frequent

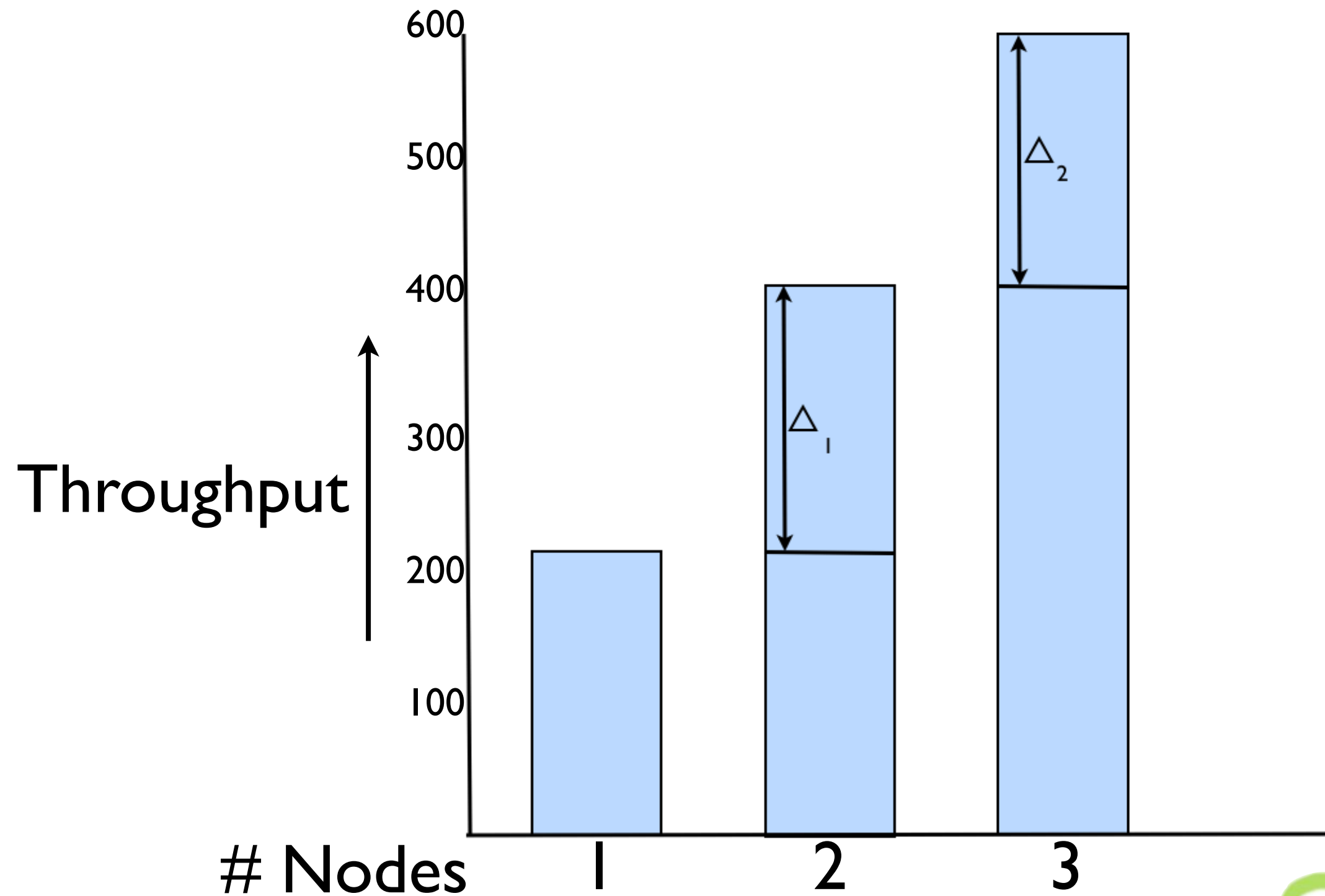
Experience report: Scalability



- Build a test that measures throughput
- Deploy on single node, measure
- Deploy on two nodes, measure
- On three, measure
- Ultimate test: do results form a “line”?

Setup: 7 machines





Test: $\frac{\Delta_1}{\Delta_2} > 90\%$

Actual: $\frac{186}{199} = 93.5\%$

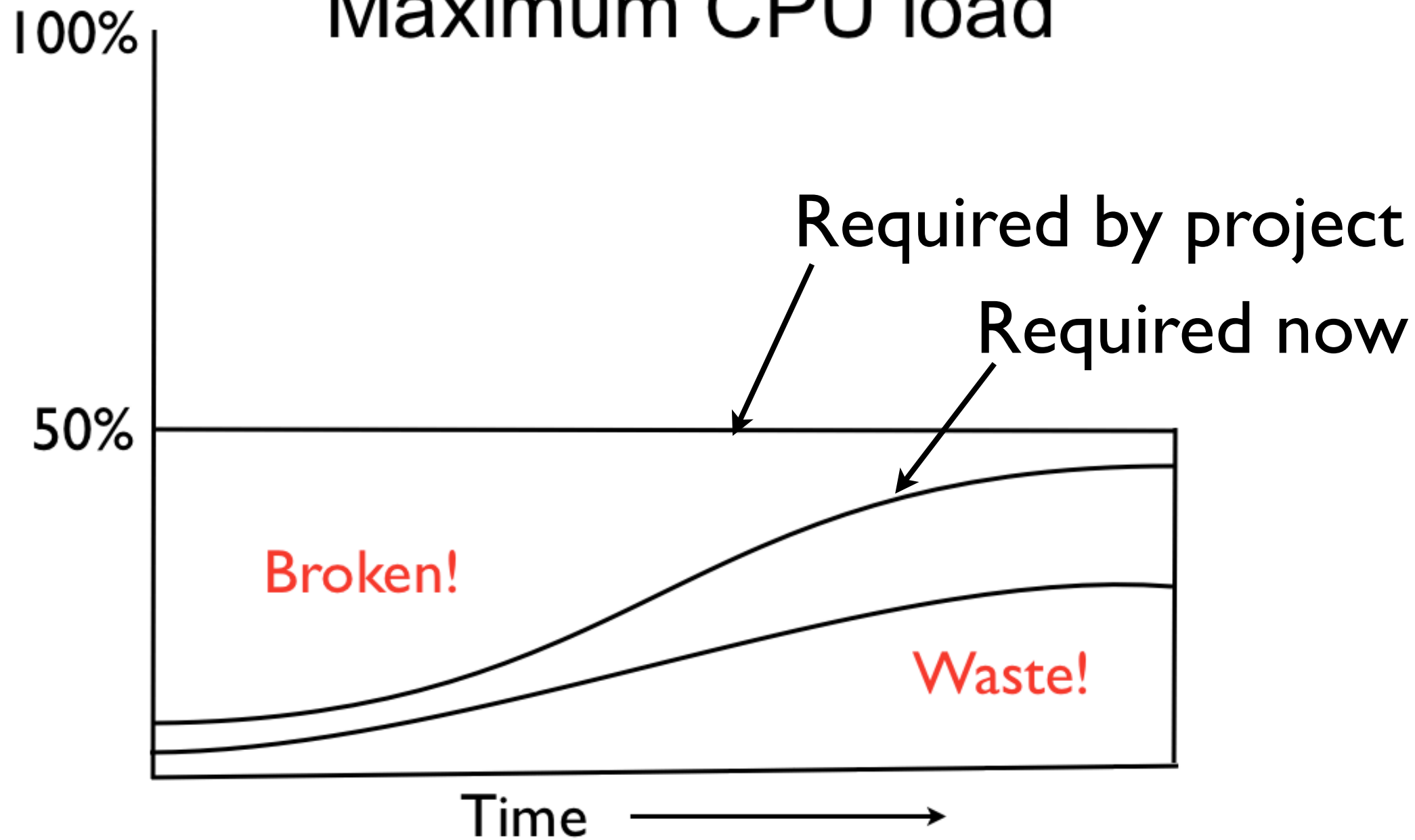


Handling broken builds

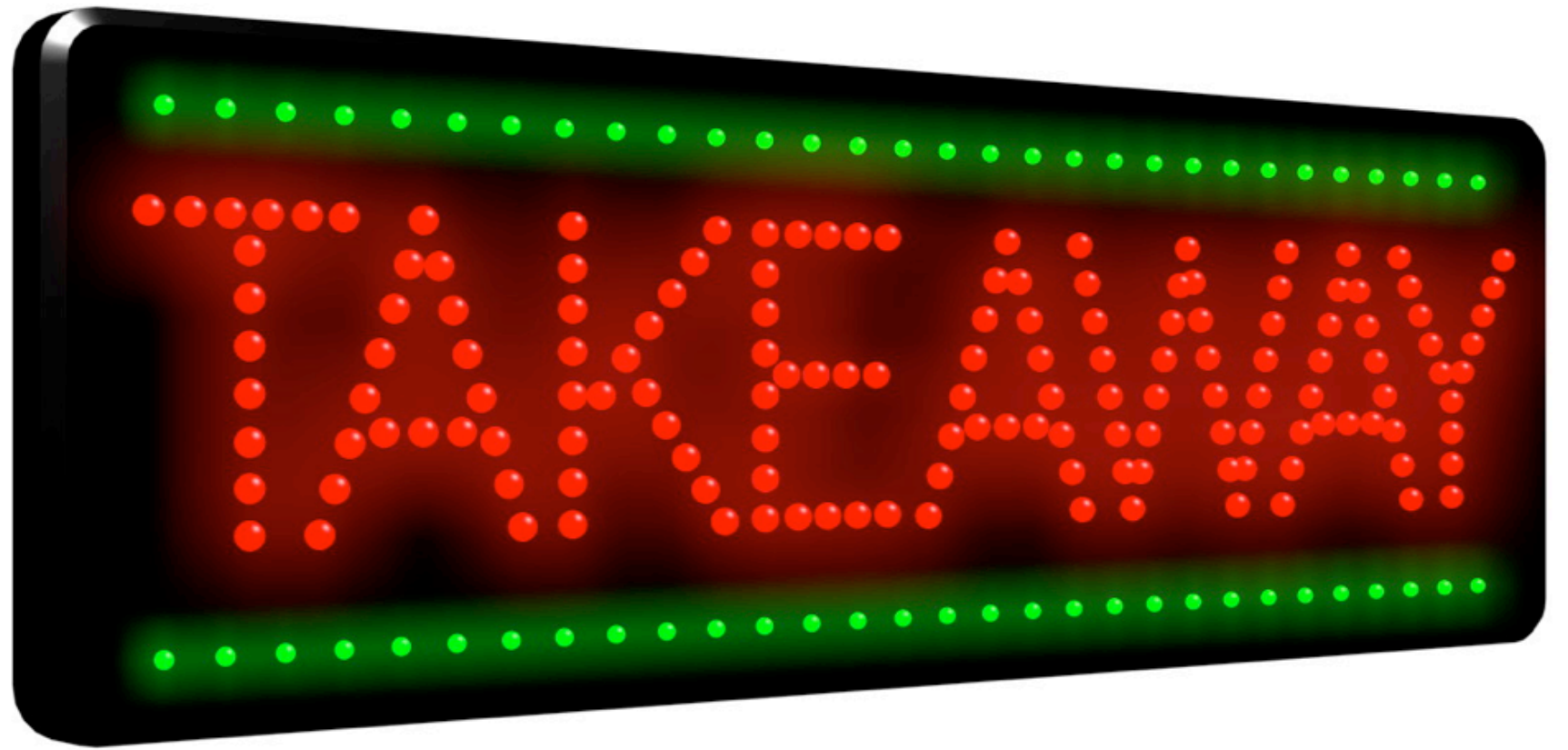


And how to prevent waste

Range test example: Maximum CPU load



Summary



- Test driven non-functionals is uncommon
- Applicable when automatable
- Should be part of regression test suite
- Prevent waste: use ranges

Wilco Koorn

wkoorn@xebia.com

+31 (0)6-10376212

