SYSTEM TESTING,
TEST AUTOMATION CHALLENGES,
AND WHAT TESTING IS ALL ABOUT IN THE DIGITALIZATION ERA

Sigrid Eldh, PhD, Adj. Prof.
Ericsson Radio System and Technology
Stockholm, Sweden
Twitter @DrSEldh

PAST TO PRESENT

- Software Testing is IT! Top 10 Future jobs!
- Most popular PhD subject at ICSE/Software Engineering 2016, 2017!! ‘
  - A key factor for economic success!
  - A road to Security ….
- Lack of sufficiently serious (university) education on Test
  - Getting better beyond formal verification
  - When programming is taught – is making sure it works (testing) equally important?
- Many businesses still do not recognize the issues within testing “it is a cost”
  - Still a lack of Know-how on Test – One word or a subject area?
- DIGITALIZATION is going on
  - EVERY BUSINESS has business dependent software
  - Yes, the can buy development/system, but testing – acceptance testing is the shit

TESTING HERE AND **NOW!**

ERICSSON

- Transformation to more Agile Practices
- System Test – and why it is hard
- Test Automation Challenges in Industry
  - & Automation for the future
- What Testing is all about in the Digitalization Era


HOW CAN WE KNOW THE QUALITY OF OUR SOFTWARE?
?
?
?


TEST/EXECUTION
IS THE
MEASUREMENT
WE HAVE OF
QUALITY
ERICSSON


HOW CAN WE KNOW THE QUALITY OF OUR TEST?
?
?
?


Mutation test? Coverage?
(thank you for all the thousands of coverage metrics there exists…)

NOT EASY – SINCE IT IS NOT ONLY ABOUT THE %

- What are factual scientific results “did we do it right”
  - Just because we do a case study it might not be a “right” result
    - We measured the Unit test? Yes, duh …
    - We “only” measured the GUI/UI? Yes, duh…
    - We only measured out model? Yes, duh…
- What are “opinions” or “alternative facts”?
- Is our selection the biggest bias? Method? Choice of SUT?
- How we prioritize?
  - And we prioritize based on what we know so far!

And in testing – there are a lot of Opinions

"WHAT IS RIGHT" IN TEST IS ONE AREA WE DIFFER – AND IS BASED ON EXPOSURE (OF) KNOW-HOW

Because "I say so"
ISTQB – a group of consultants ++

Because "I say so"
James Bach
Michael Bolton and co

We can agree on some things…

Because "I say so"
SWEBOOK
Eda Marchetti & Antonia Bertolina ++

Because "I say so"
Any authority…

Because "I say so"
IEEE, ISO/IEC, DoD, ETSI, ITU/IEC
….standard org..

Because "I say so"
IEEE, ACM, Springer, google Databases…

!

!

!

WHAT HAPPENED WITH TESTING IN THE AGILE CONTEXT?

System test (and testers role) is diminished, testing has become more of a developers task, this has some consequences:

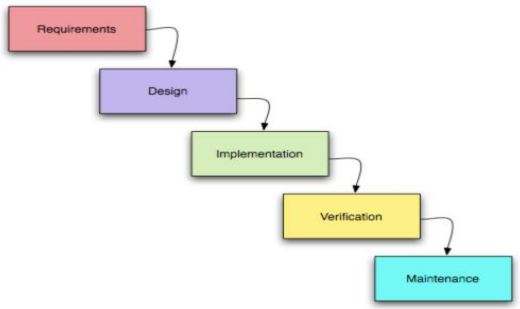
- More tests – boost and management quality engagement
  - (but are they good test, or just many in number?)
- Test cases more focused on code level (not system level)
- End-to End hard in large complex (telecom) systems
- Hard to get “users in team” – Requirement deterioration
  - Requirements? User Stories? Detail? Specification?
- “Quality Police”/gating and trust – back to hacker culture?
- Some lack architectural support – great hack in big bang
- TDD – is really a low level specification…not so easy with “Many layers”

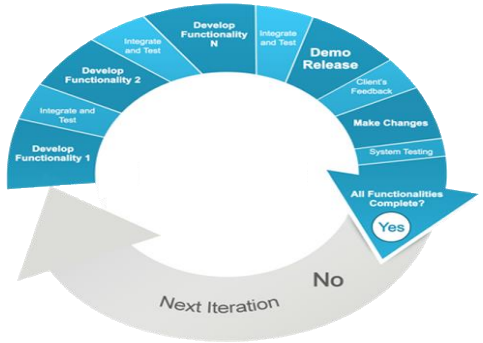
Much better TOOLS

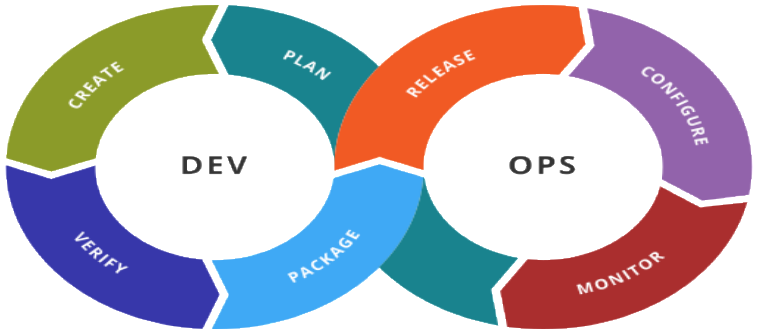
Faster

THE AGILE TRANSFORMATION

Past → Present → Future







Waterfall
Manual Integrations

Agile
Scrum, Test Automation

DevOps
Continuous Feedback between Development and Operational Phases

Shortening of Release Cycles / Towards Extended Digital Enterprises

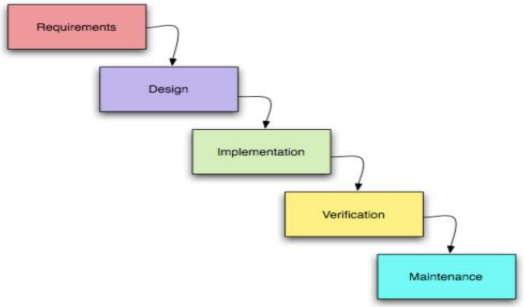
Long iterations (e.g .,releases every 18 months)
Point-to-point integration

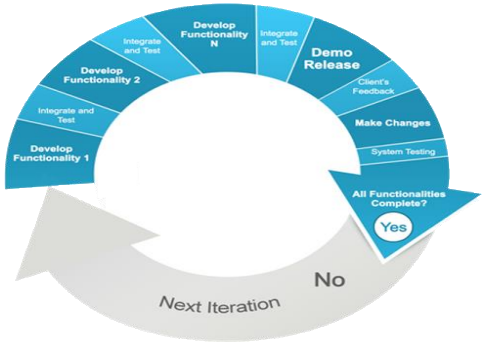
Shorter iterations (e.g., from 2 weeks to every 6 months)
Enhanced people cooperation

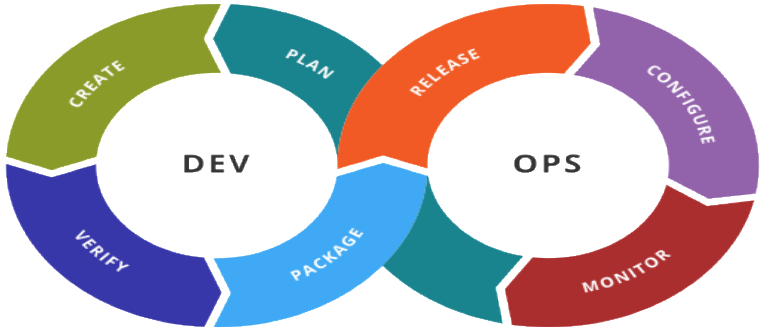
Very short iterations (e.g., every weeks, or even days!)
Continuous Deployment
Big Data Analytics
Optimizing the value stream
Federated Environments

THE AGILE TRANSFORMATION

Past → Present → Future







Waterfall
Manual Integrations

Agile
Scrum, Test Automation

DevOps
Continuous Feedback between Development and Operational Phases

Shortening of Release Cycles / Towards Extended Digital Enterprises

Manual handover – rebuilding due to new environment

- **Test suites partly automated within a level**
- **Integration found new faults**
- **Developers distant from ownership of Quality**

Focus on Test Execution Automation

- **Build – retest**
- **Test Prioritization focus**
- **Maintenance of test suite growth costly**
- **Focus on "unit tests"**

Federated Environments – Key to Testing (Architecture,

- **Assume fast & automatic fault find & fix**
- **You need: exact identification of what has changed" – and what it impacts!**
- **Targeted automated test "**
- **E2E – and System**
- **Metrics at "all levels"**

CHALLENGES WITH TEST AUTOMATION

- Type of Tests
- Manual Test Cases got automated
  - Architecture, utilization of sw –libraries, loops etc
  - Overlap – Cloning –copy paste
- Test Suites grow - Test Selection/Test Regression….Test Refactoring?
  - Power, cost of maintenance, “finding a test case”
- What got tested if “big bang CI/CD test”? Fault finding? Causes?
  - Changes, one or many? Dependencies in software!
- Lack of Test Design
- Issues with Test Environment

CHALLENGES
- FROM MANUAL TEST TO AUTOMATED

Result

› In-effective tests

Challenge

› A mess of TC

Solution

› Architecture:
  – Redesign?

CHALLENGES
- FROM MANUAL TEST TO AUTOMATED

Result

- In-effective tests
- Abundance of Duplications!

Challenge

- A mess of TC
- Waste, Cost

Solution

- Architecture:
  - Redesign?
- Cloning, Coverage, Libraries, ++

CHALLENGES
- FROM MANUAL TEST TO AUTOMATED

Result

- In-effective tests
- Abundance of Duplications!
- Manual tests are doing things once

Challenge

- A mess of TC
- Waste, Cost
- Poor test cases

Solution

- Architecture:
  - Redesign?
- Cloning, Coverage, Libraries
- Refactor test cases, Analytics of “value”, New Techniques (loop, constraints…)

CHALLENGES
- FROM MANUAL TEST TO AUTOMATED

Result

- In-effective tests
- Abundance of Duplications!
- Manual tests are doing things once
- Hardcoded DATA in test cases

Challenge

- A mess of TC
- Waste, Cost
- Poor test cases
- Low data coverage

Solution

- Architecture:
  - Redesign?
- Cloning, Coverage, Libraries, ++
- Refactor test cases,
  - Analytics of “value”
  - New Techniques (loop, constraints…)
- Separation of data & execution:
  - Triplets: “steps”, Input, output
  - Constraints
  - Property based or “category partitioning”

CHALLENGES
- FROM MANUAL TEST TO AUTOMATED

Result

- In-effective tests
- Abundance of Duplications!
- Manual tests are doing things once
- Hardcoded DATA in test cases
- Developers code = Unit tests (and at best low-level functions)

Challenge

- A mess of TC
- Waste, Cost
- Poor test cases
- Low data coverage
- System tests (E2E and user aspects) still manual or "poor"

Solution

- Architecture:
  - Redesign?
- Cloning, Coverage, Libraries, ++
- Refactor test cases,
  - Analytics of "value"
  - New Techniques (loop, constraints…)
- Separation of data & execution:
  - Triplets: "steps", Input, output
  - Constraints
  - Property based or "category partitioning"
- Many…

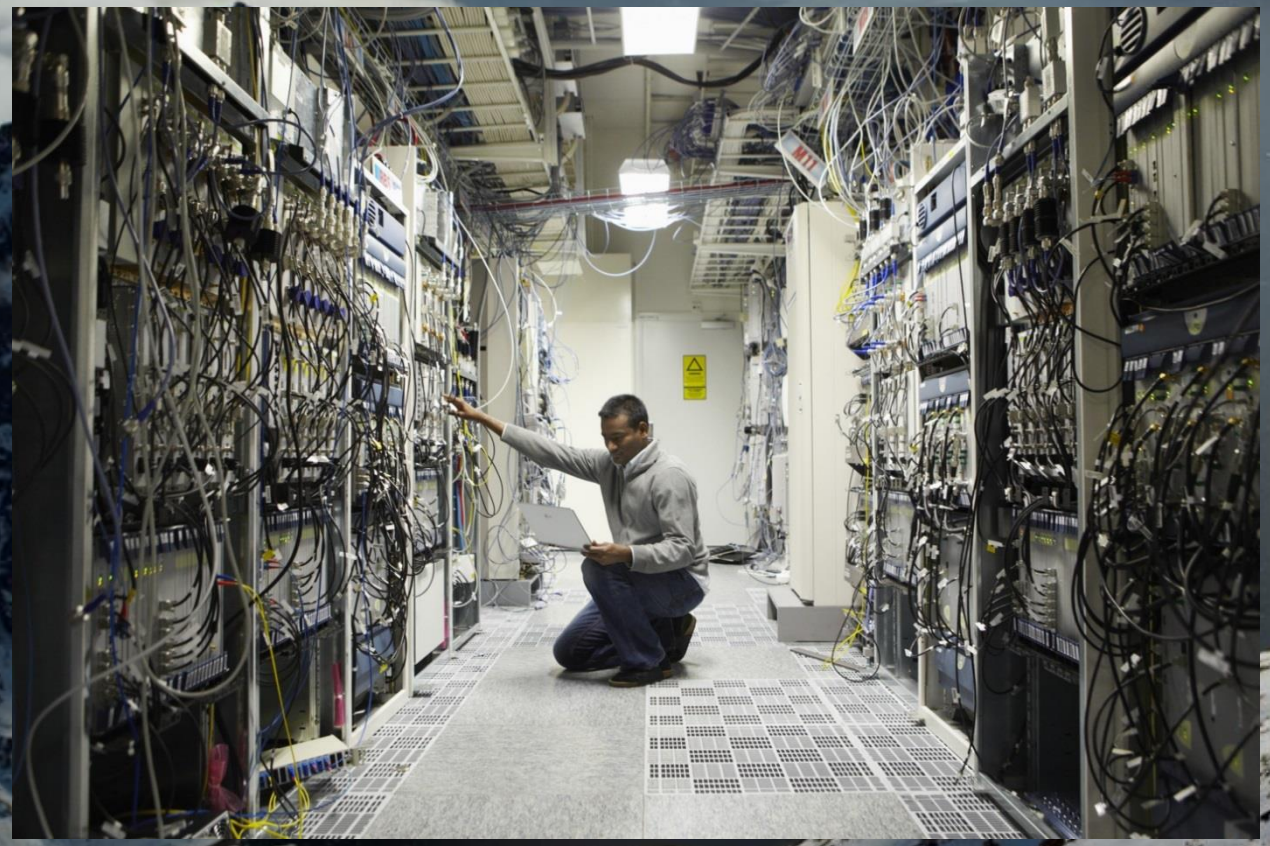
CHALLENGES - AUTOMATED TC

- For Industry 99% = Automated TEST EXECUTION
  - (if you are not in Germany ;-) or university educated..
- Oops! – Test Verdicts? Test Results? “Post processing”Test oracles??
  - Easy for functional – but for system tests?
- Industry is NOT EVEN thinking test design technology
- Most are thinking “requirements should be tested”
  - Of course Model based testing is a great cure for describing and defining “what the system should do”
  - Easy to miss – what the system does – reality- environment, timing and fault handling!
- Search-based testing ….ok?
- Mutation testing? – But that is for unit tests, right?

THE TEST SUITE MAINTENANCE CHALLENGE

- Keep adding new tests, but do not dear to remove old
  - The copy paste “curse” with small change
- We have analytics in place, but – you maybe need this test case “every year”… when we change that part…
- Test Architecture (never important enough) compare to CODE (and usually bad)…
- Now all our focus talks about
  - Order of execution – test priorities, what to select in the different “loops” of execution…. (smoke test/short loop, to night and week)

THE REALITY OF TESTING MORE COMPLEX SYSTEMS



Difficult Test Environments

Simulations

SOME CALL IT "FLAKEY TESTS"

- It is really not the test CASES that are flakey – it is the test environment – and the automatic (coded) interaction with Test environment!
- Why?
  - Hardware fails, get stuck, glitches, and overheat. Or is not even turned on…
    - But we tested on Simulators "that always worked", reset it self and have clear states defined… yes, much to fit the software!
  - The so called "Chaos Monkey Solution"?
    - Plain Robustness test – Inject a fault (pull hardware, cause glitches, turn off and on very quickly etc) – make sure the system can handle it = robust/reliable
    - If live – a nice bi-product – approach kills "hanged sw"
  - The dependencies!
  - The way the test automation is done in the agile "big-bang testing" in large labs
    - to execute automatically on a large set of different configurations
  - Timing dependencies
  - Memory/buffer and "state" of system at TC start
  - Assumption on not impacting, when in fact it does (just by using some shared resources)

FUTURE OF TEST

Test Measuring the Quality

- to “Quality metrics analytics”

Test “crawlers” – find, verifies and auto-correct

Autonomous systems

IoT – Testing gadgets suits well with MBT, FV

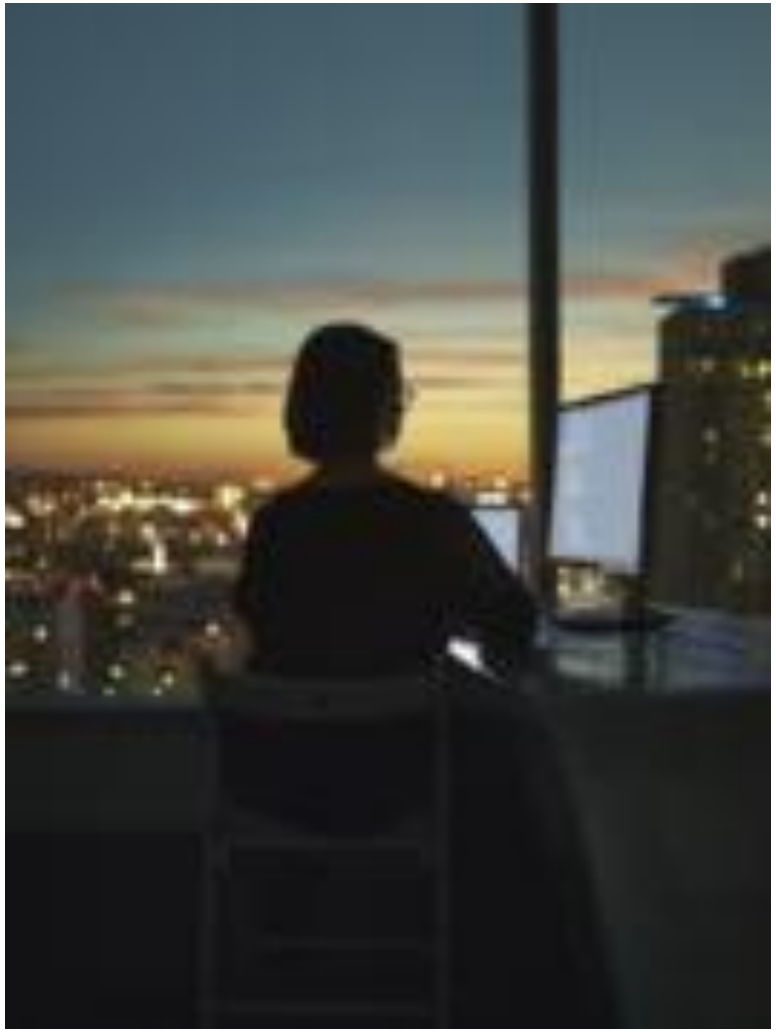
Backhaul/networks and virtual/cloud needs completely different approaches

Fault – input – gadget “tolerance”

Real-time Analytics – adaptive

Automated fault localization

From languages – to “scalable” static and dynamic analysis



SYSTEM TEST – WHY IT IS HARD!

- Installation/update of Telecom Systems – An example….
- Domain specific – Requirements fulfillment – Specification maturity
- E2E – test of complex systems – and advantage of slicing/components
- Real time components
  - Timing, concurrency, Embedded, Scheduling “in the cloud”
- Stable - Robust – Reliable Systems
  - Fault Injection, interference
- Performance Test and Automation
- Metrics support? ISO/IEC 25023? (based on ISO/IEC 9126)
- Automation of test verdicts? Why not so easy when analysis
- Troubleshooting? Fault location, traces and logs

AUTOMATION IN THE FUTURE

- Modelling maturing – Generate the test
- Great examples for e.g. GUI testing – “evolve beyond” manual…
- Design for self-healing
- Systematic approaches to negative tests....(outside the spe), random…
- Fully explore a system….
  - Now limits can be broken…
- Automation of “all aspects” in sw – self *
  - Loss of tactile know-how?
  - Fault location – automatic program repair?
- Learning in Test – You still need to know what is “correct”

TESTING IN THE FUTURE DIGITALIZATION!

- Let us Automate most of it away!
- Or – You get the quality (security, safety, reliability…) you are paying for!
  - Let us stick with this dream – that is not to far distant…
    - Modelling –
    - Automatic Program Repair
    - Self-Healing systems
    - Reliable software
- There are still a LOT to do in software and software test

Our different views

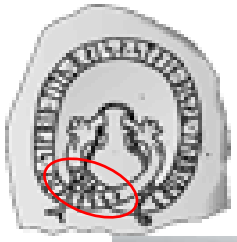
THE TESTOMAT PROJECT

- The NEXT Level of Test Autoamtion in CI/CD context
- Test Effectiveness
- Test Efficiency – Speed
- Quality Test Standards
- Test Automation Improvement (Model)
  - Basing Automatic improvements on Automatic Metrics = ML ++

Thank you for listening!

Follow me at Twitter @DrSEldh
& check out @TESTOMATPROJECT

DR. SIGRID ELDH



- Ericsson 20+ years many levels of test, 10+ years of management
  - Now Leading Ericsson Research on Software Test Quality, and Debug
  - 10+ years experience from Other business: HP, Government, Consultancy, University
  - Supervised/-ing 6 PhDs
- Adj. Professor @Carleton University Canada
- PhD “On Test Design”
- Started SAST, ISTQB, SSTB, ASTA
- Now ITEA 3 Testomat Project

Twitter @DrSEldh

ERICSSON