Automated System Level Software Testing of Networked Embedded Systems

Mälardalen University Licentiate Thesis 275

Per Erik Strandberg

1. Personal and Industrial Context

Photo: Per Erik Strandberg

Per Erik Strandberg, the Test Expert

- 2017: Industrial doctoral student
- 2006 – 16: Work in “industry”
- 2011-16: competence network in testing
- Certified tester (3xISTQB)
- Certified professional for requirements engineering (REQB CPRE)
- Employed full time at Westermo

Per Erik Strandberg at ESEM’18. Photo: Päivi Raulamo-Jurvanen

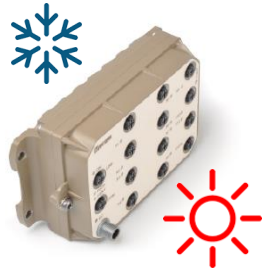
Per Erik Strandberg, the Person

Selfie. Photo: Per Erik Strandberg

Westermo

- Founded 1975
- 200+ employees
  (Most in Västerås and Stora Sundby)

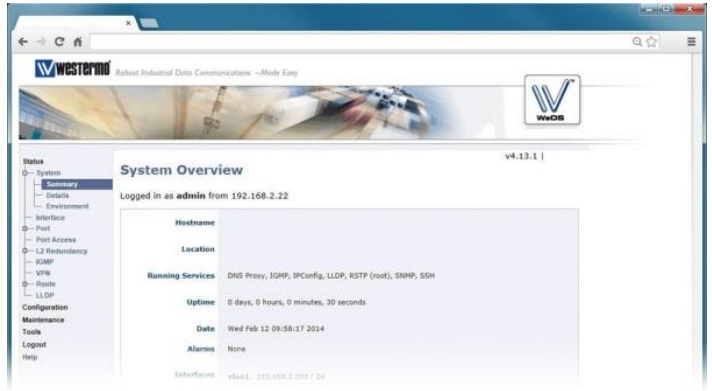
- Westermo designs and manufactures data communications products for mission-critical systems in physically demanding environments.



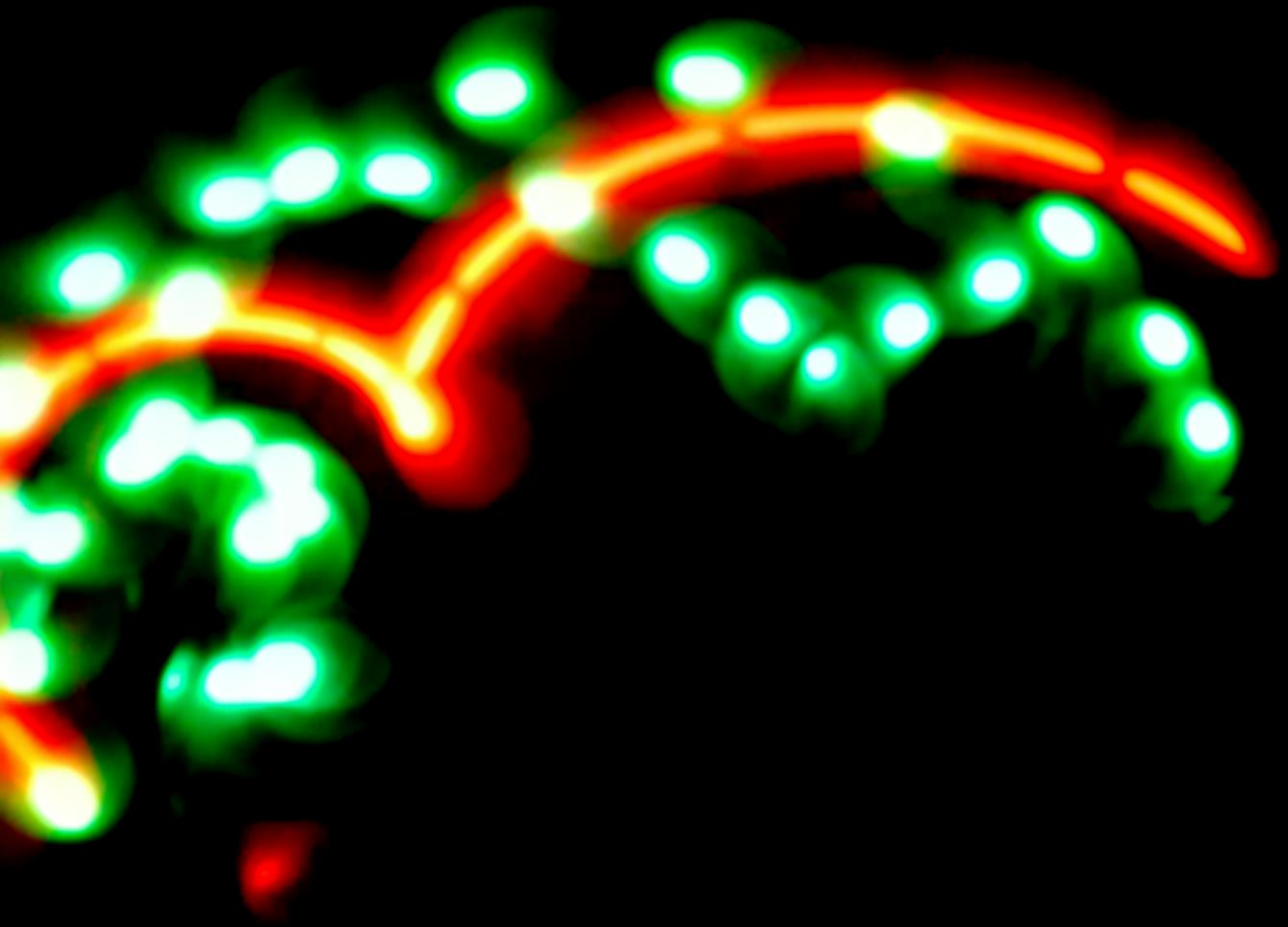
Westermo Products. Photos: Westermo.

WeOS – Westermo Operating System

- One software – Many products
  - GNU/Linux + Open Source
  - Proprietary 3’rd party libraries
  - Proprietary internal code
- Developed every day
- Tested every night



Westermo Products. Photos: Westermo



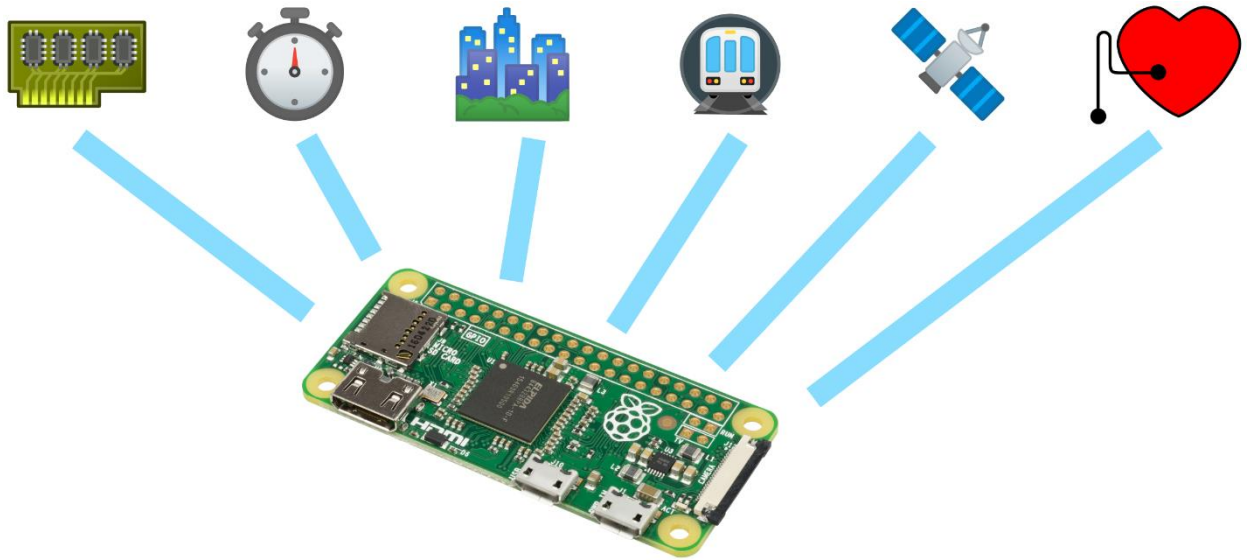
2. Networked Embedded Systems?

Photo: Per Erik Strandberg

Embedded System

- “An embedded system is a programmed controlling and operating system with
- a dedicated function
- within a larger mechanical or electrical system” –Wikipedia

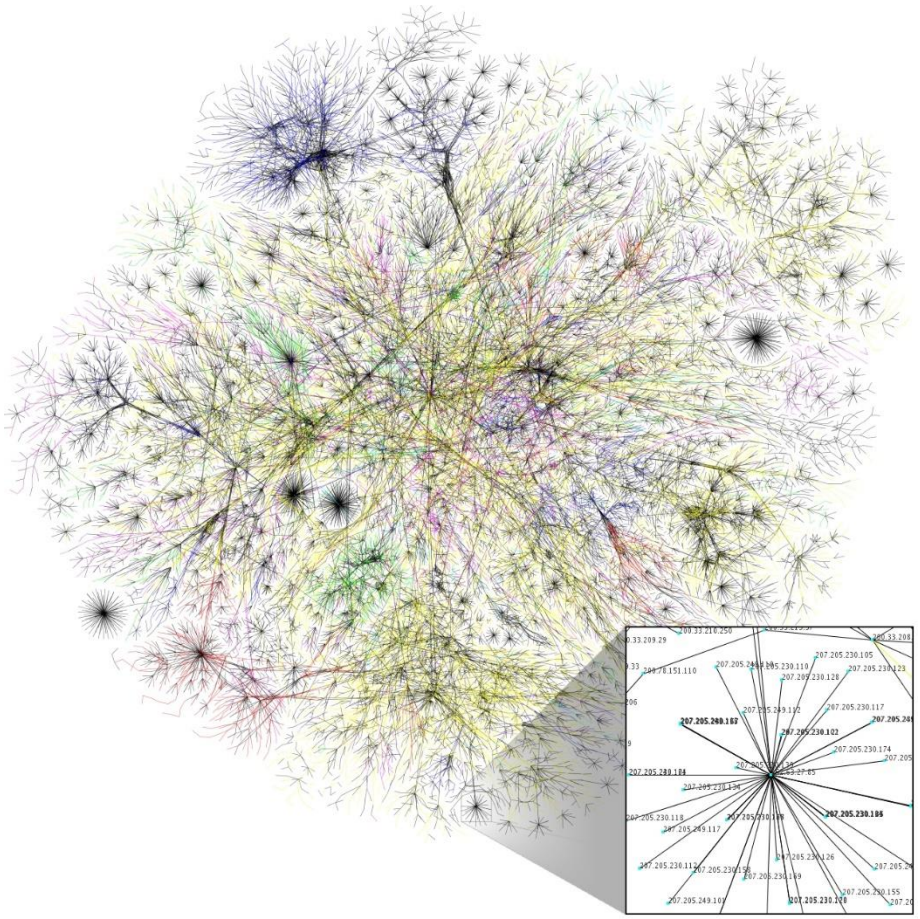
- Typically: no mouse, keyboard, monitor, …



Raspberry Pi Zero. Photo: Evan-Amos, Public Domain

Computer Network

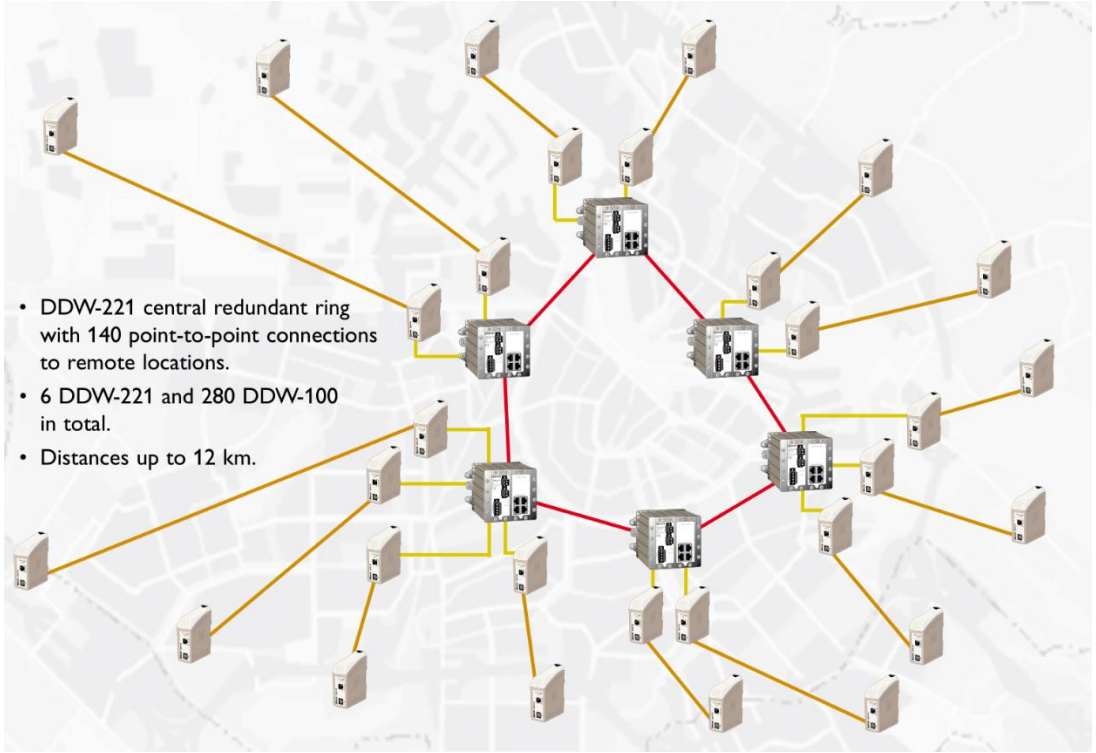
- “A computer network, or data network, is a digital telecommunications
- network which allows
- nodes to share resources” -- Wikipedia



The Internet (partial map as of 2005). Image: the Opte Project, CCA 2.5

Networked Embedded Systems

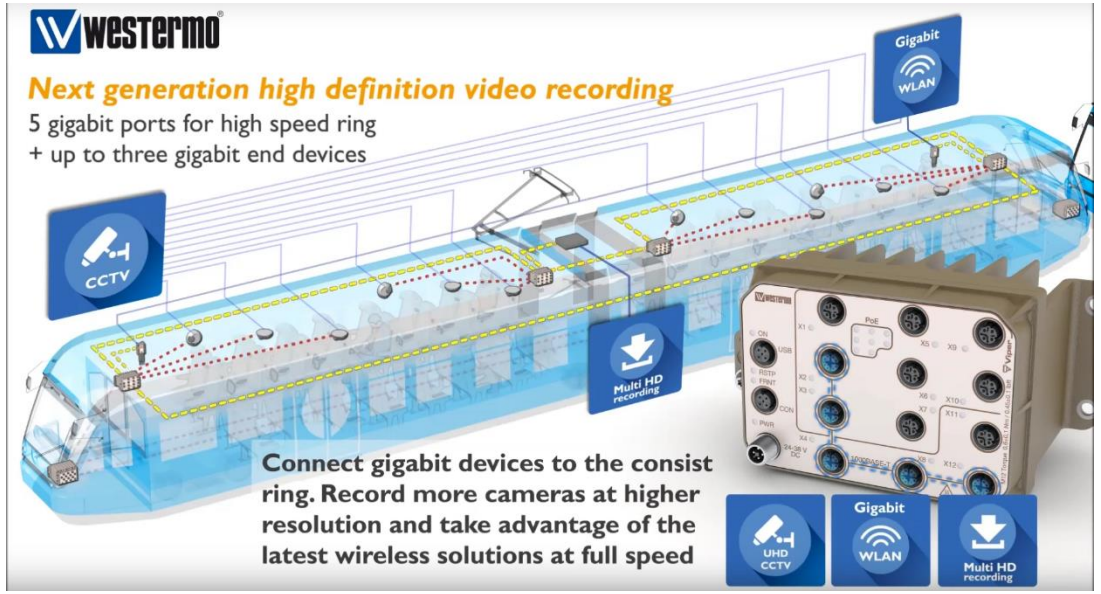
- In our cities!



Network Amsterdam gas and energy distribution. Image: Westermo.com

Networked Embedded Systems

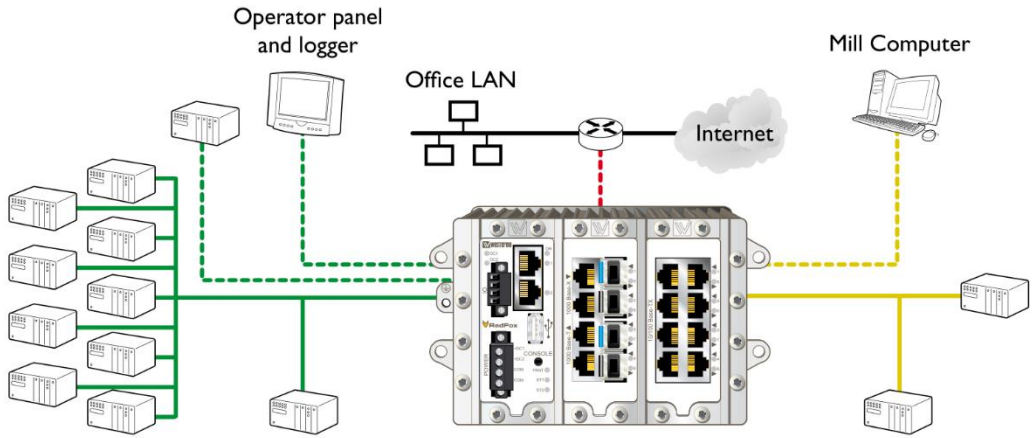
- In our cities!
- In our transportation systems!



Train Network. Image: Westermo at YouTube

Networked Embedded Systems

- In our cities!
- In our transportation systems!
- In our factories!



Industry/Factory Network. Image by Westermo.com

Networked Embedded System

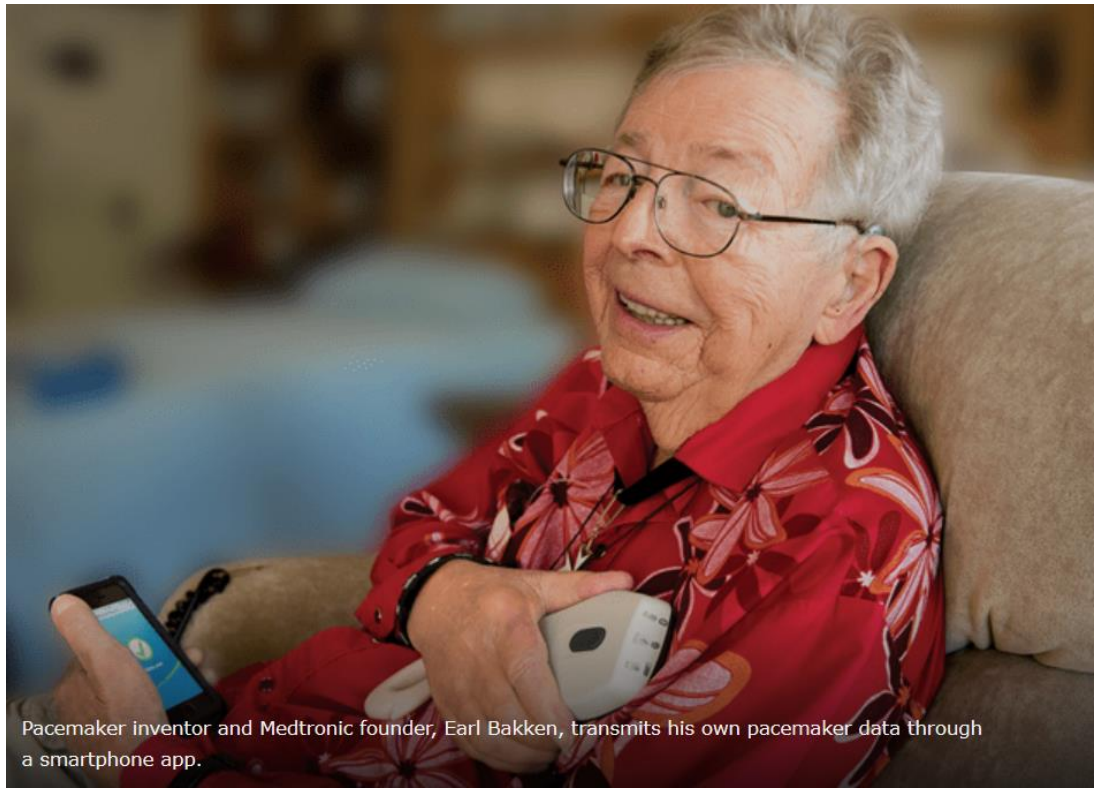
- In our cities!
- In our transportation systems!
- In our factories!
- In our homes!

Remore Lock

Pet Pal Wifi Feeder

Networked Embedded Systems

- In our cities!
- In our transportation systems!
- In our factories!
- In our homes!
- Inside our bodies!



Screenshot from hospitalnews.com

Software Quality? Bugs?

- Quality shortcomings
  - Johnson & Johnson warns diabetic patients: Insulin pump vulnerable to hacking, Jim Finkle, Reuters, 2016 Oct 4
- Software Quality is Key!
- Testing is the Standard Method

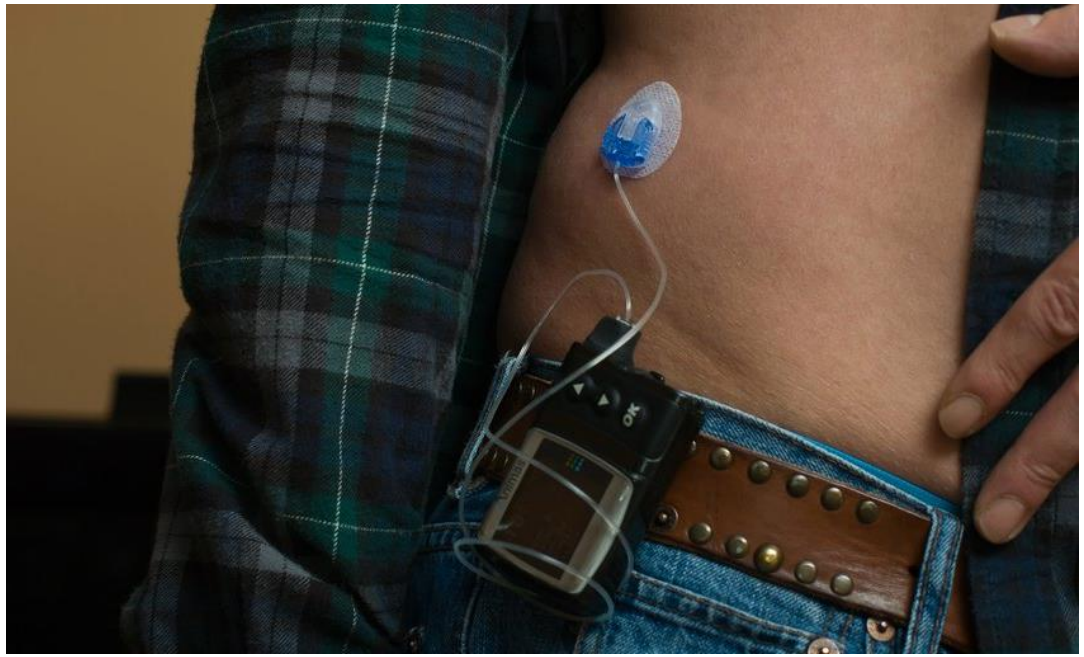


Image from businessinsider.com

Software Testing!

Software testing, or testing, can be defined as the act of:

manually or automatically

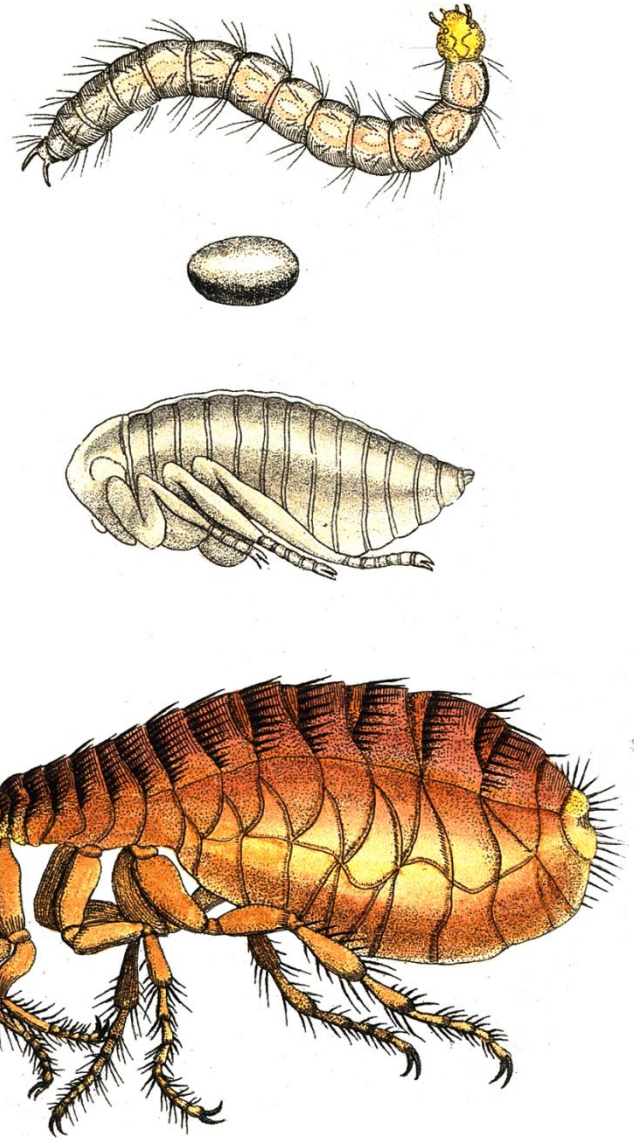
inspecting or executing software

with or without custom hardware

in order to gather information for some purpose:

feedback, quality control, finding issues (“bugs”), building trust, or other.

-- Per Erik Strandberg

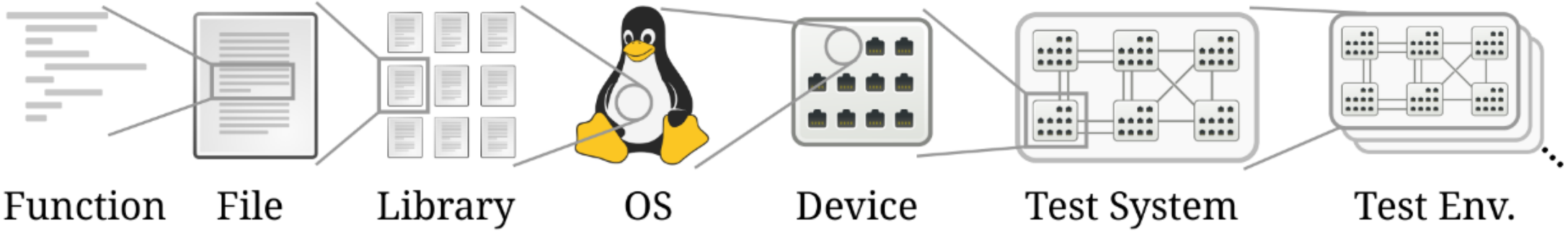


“Bug” (a dog flea). Image: WellcomeImages, CCA 4.0

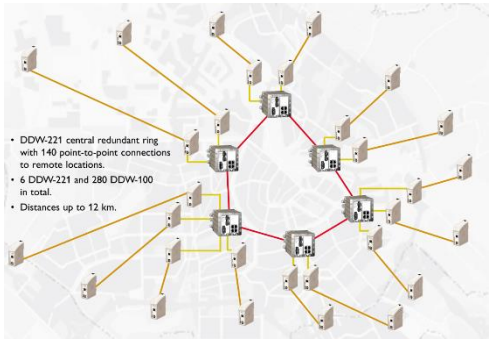
3. Testing Networked Embedded Systems

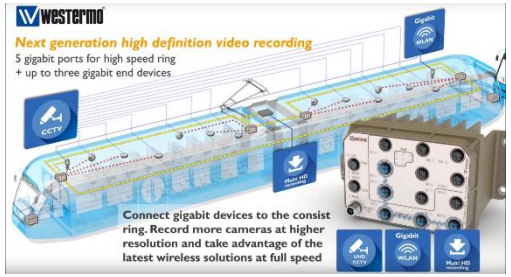
Photo: Per Erik Strandberg

Test Levels



- Left: Unit level testing
  - 0.2 ms/test case (Paper A)
- Right: System level testing
  - 2 minutes/test case (Paper A)

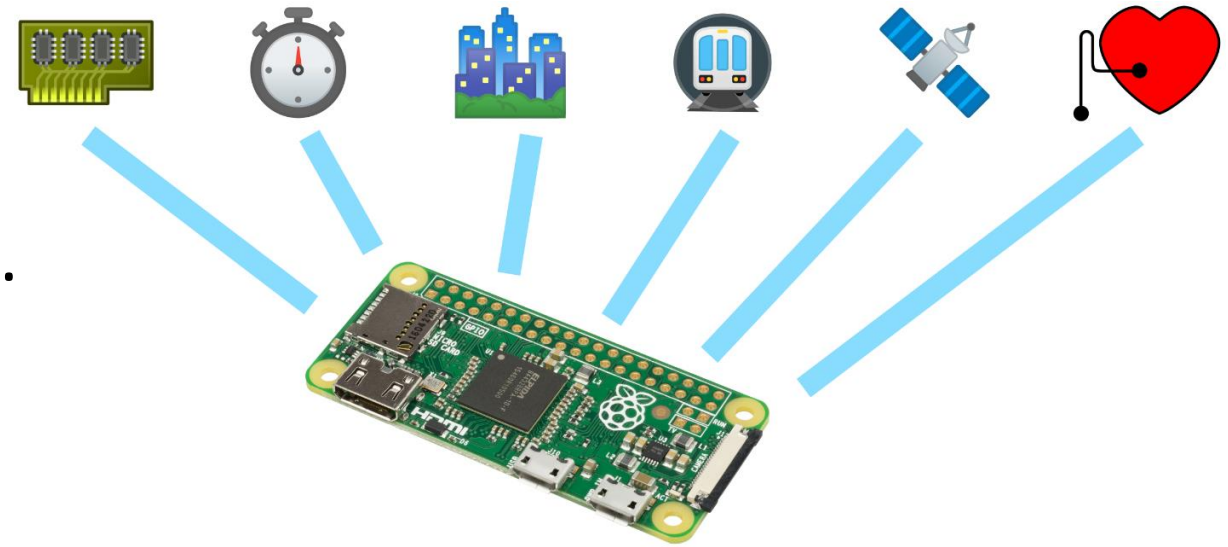




Test Levels. Illustration from thesis introduction.

Testing Networked Embedded Systems

- Devices within a larger system.

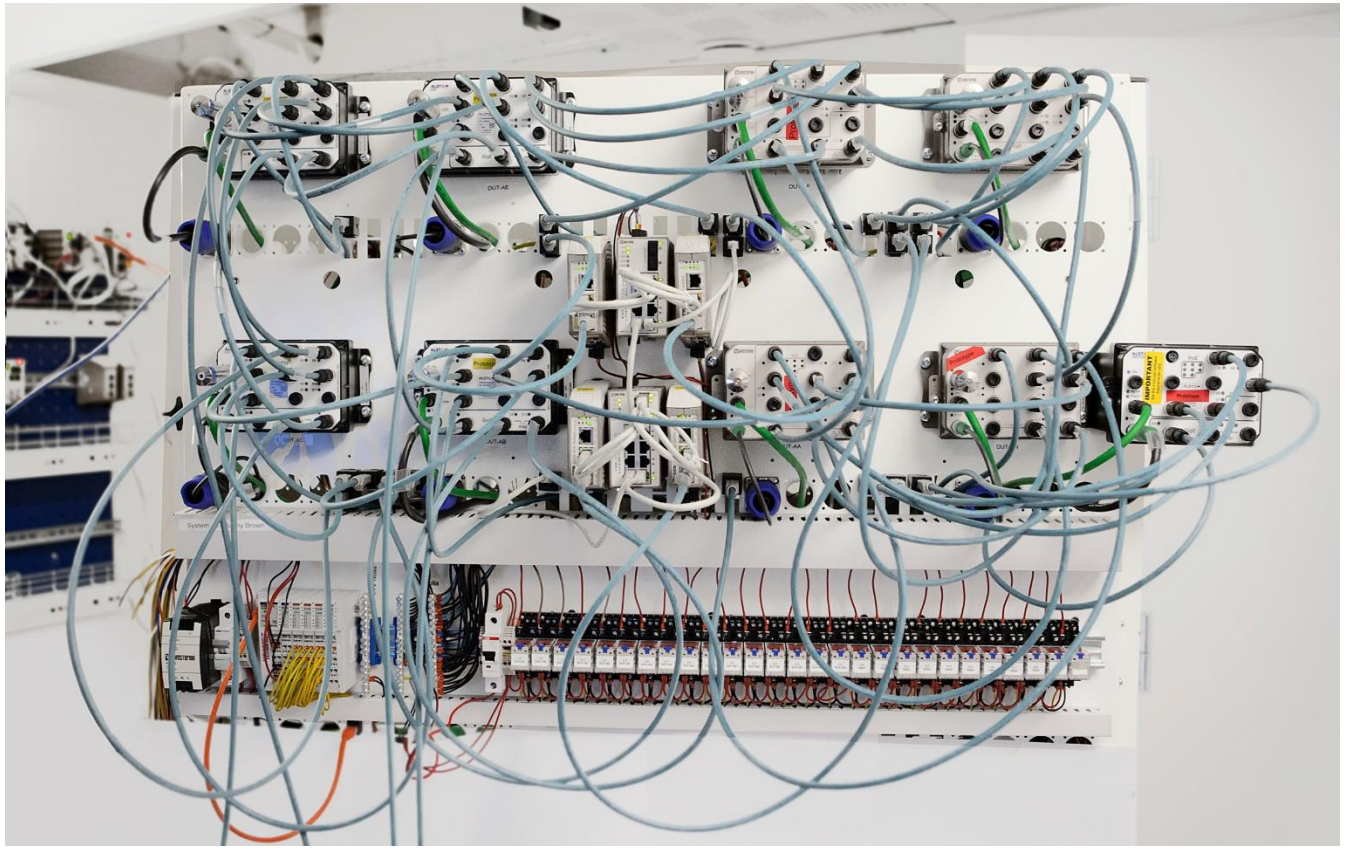


- We need to test on real hardware.
  - [3] Banerjee et al. (2016)
  - [53] Ronsenkranz et al. (2015)
  - [71] Wolf (1994)

- Build Test Systems

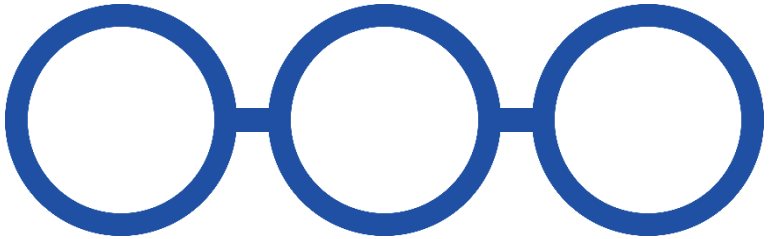
A Westermo Test System

- Devices Under Test (DUTs)
  - Run WeOS
- PC Server (not seen)
- Cables
- Linkbreakers
- IO’s
- On a wagon (with wheels)

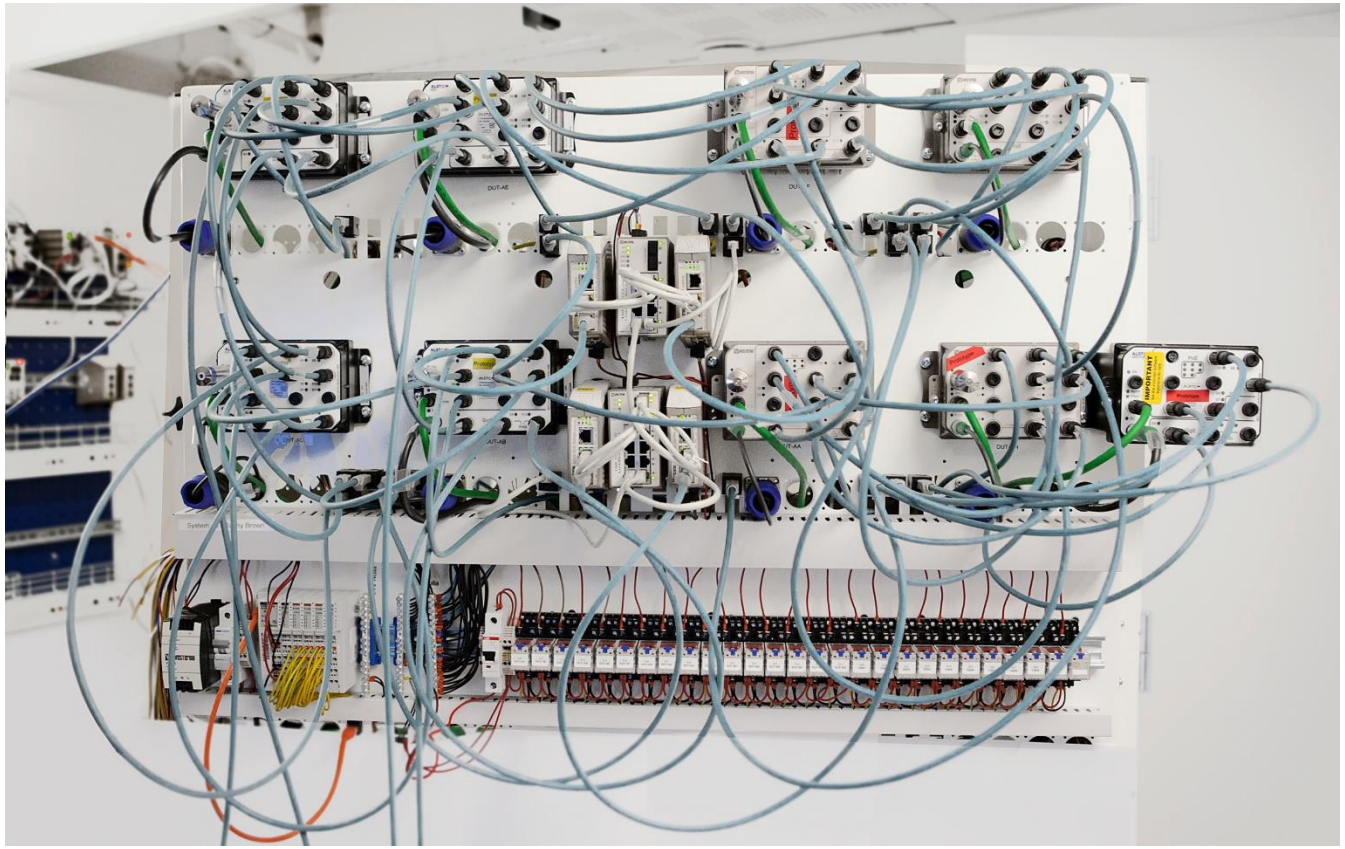


A Westermo Test System. Image from Thesis Introduction.

Firewall: A Typical Test Case



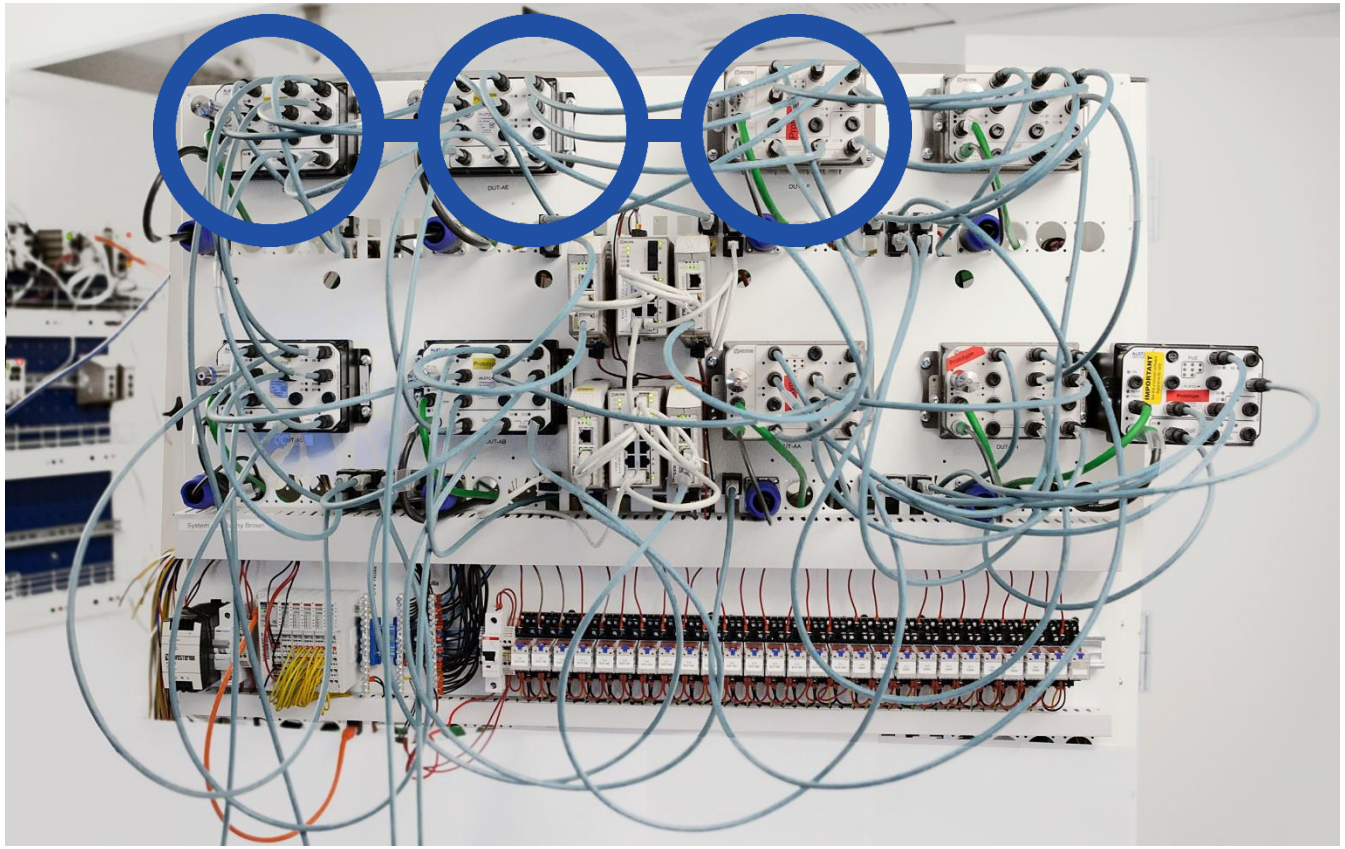
- DUT 1: Inside
- DUT 2: Firewall
- DUT 3: Outside



Firewall: A Typical Test Case

- DUT 1: Inside
- DUT 2: Firewall
- DUT 3: Outside

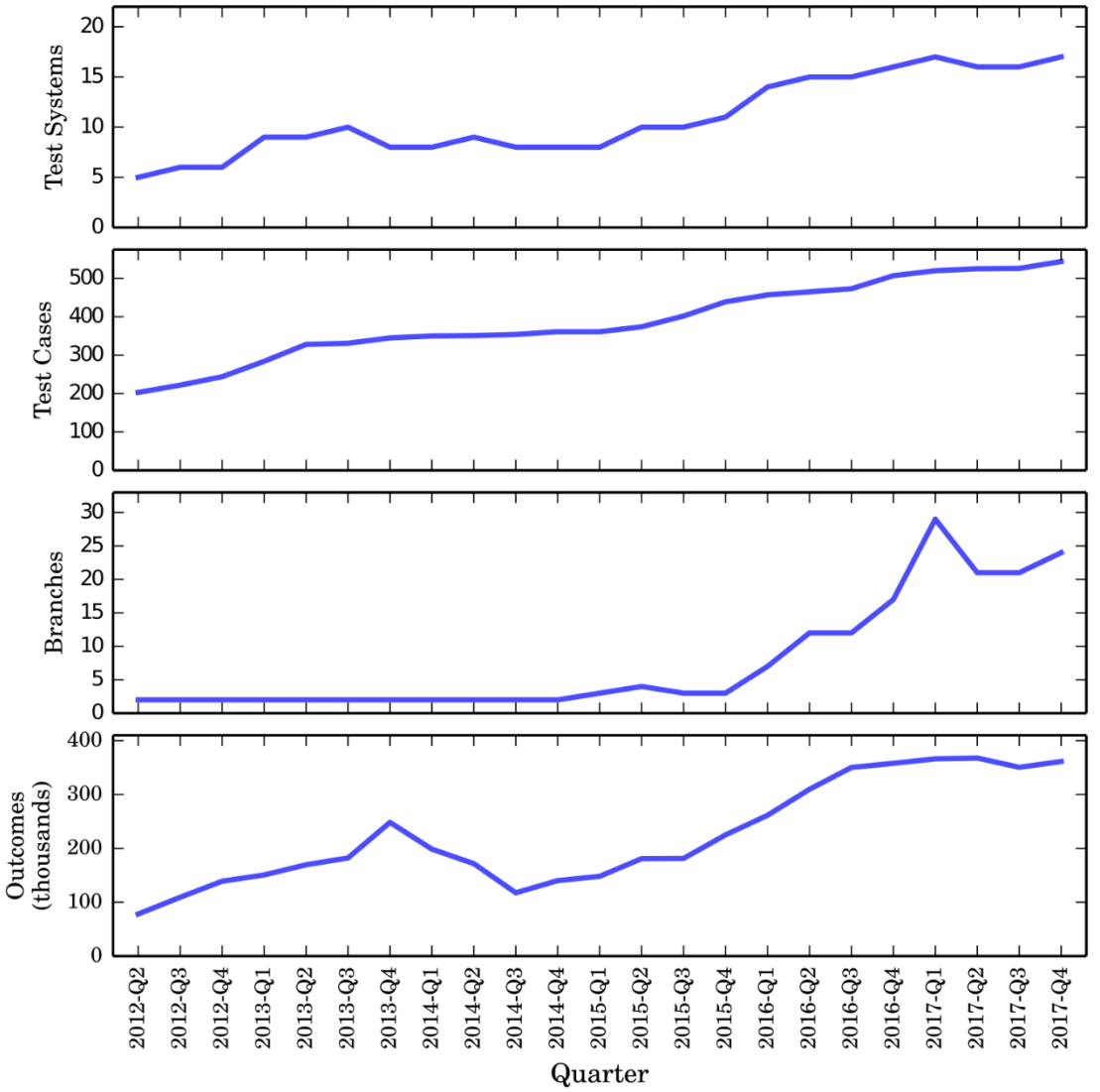
- Example of one “Mapping”, on one test system
  - 15 s/mapping (Paper B)



Automation Challenge

- Westermo is doing well
- New HW products ☺
- New SW features ☺
- New SW development model ☺
- Million outcomes/year ☺

- Test all combinations? ☹
- Who receives the information? ☹



Increasing Combinatorial Complexity. Image from Paper C

Motivation

- Test Automation Challenges, [69] Wiklund et al. (2017)
  - Lack of time
  - Test systems not available
- Automated/Continuous Practices, [56] Shahin et al. (2016)
  - Exponential growth of information
  - Lack of awareness and transparency
- This matches what we see at Westermo

4. Research Questions

Photo: Per Erik Strandberg

Research Goal and Questions

Describe and improve upon industrial automated system level software testing of networked embedded systems.

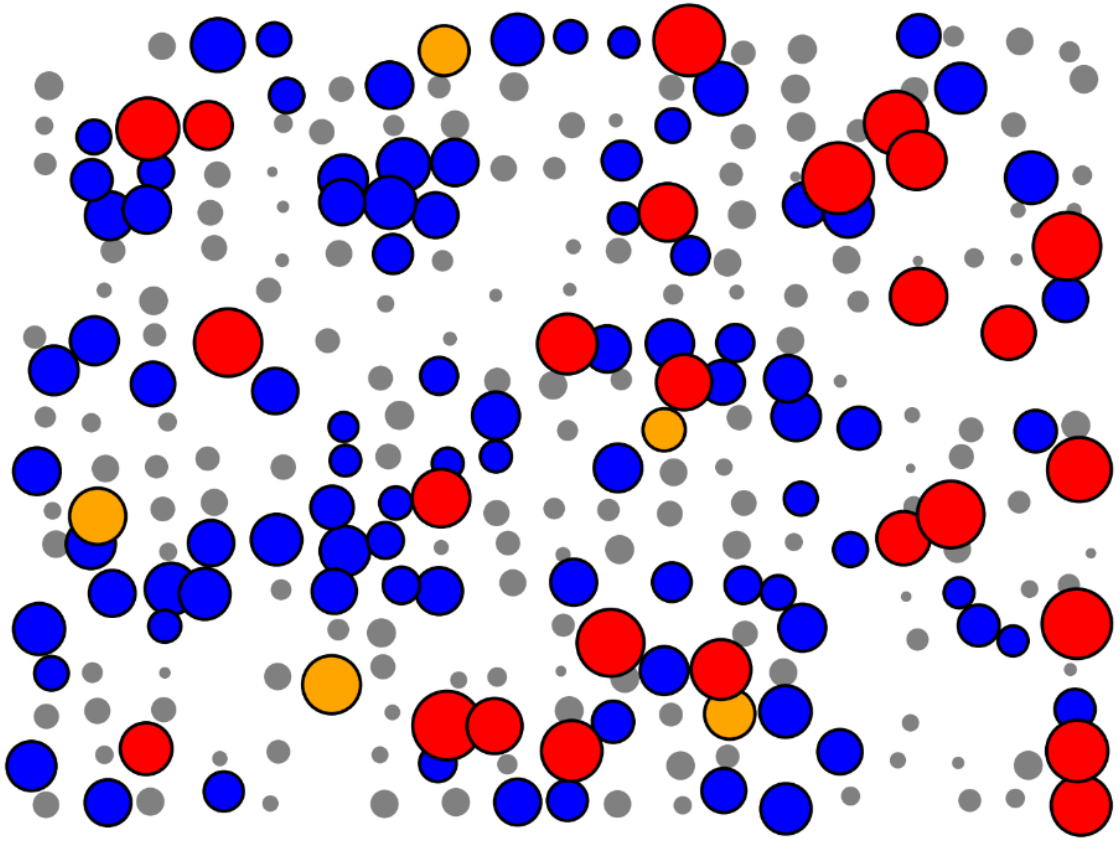
1. Test selection
2. Test Env. Assignment
3. Information Flow

The Goal. Photo: Per Erik Strandberg

Research Goal and Questions

Describe and improve upon industrial automated system level software testing of networked embedded systems.

1. **Test selection**
2. Test Env. Assignment
3. Information Flow

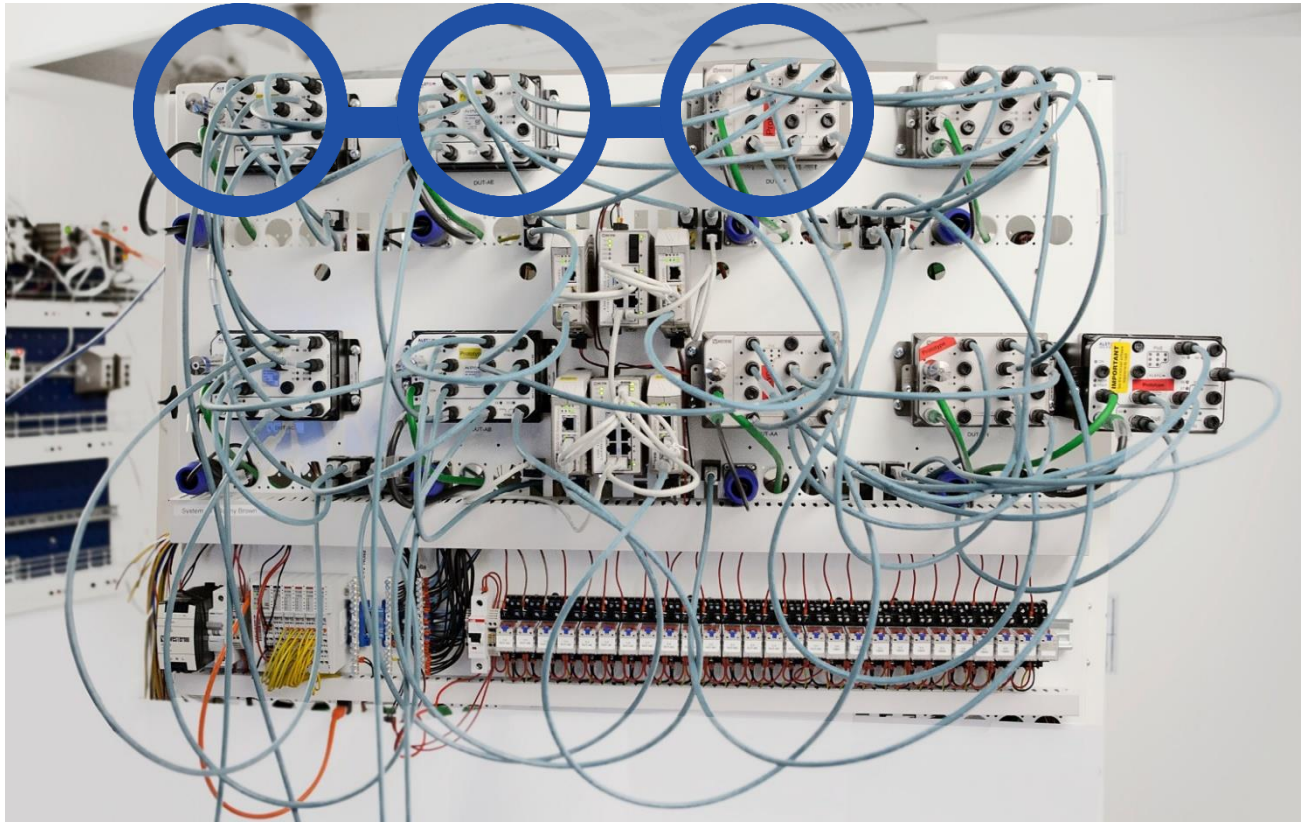


Select from 275 dots. Illustration: Per Erik Strandberg

Research Goal and Questions

Describe and improve upon industrial automated system level software testing of networked embedded systems.

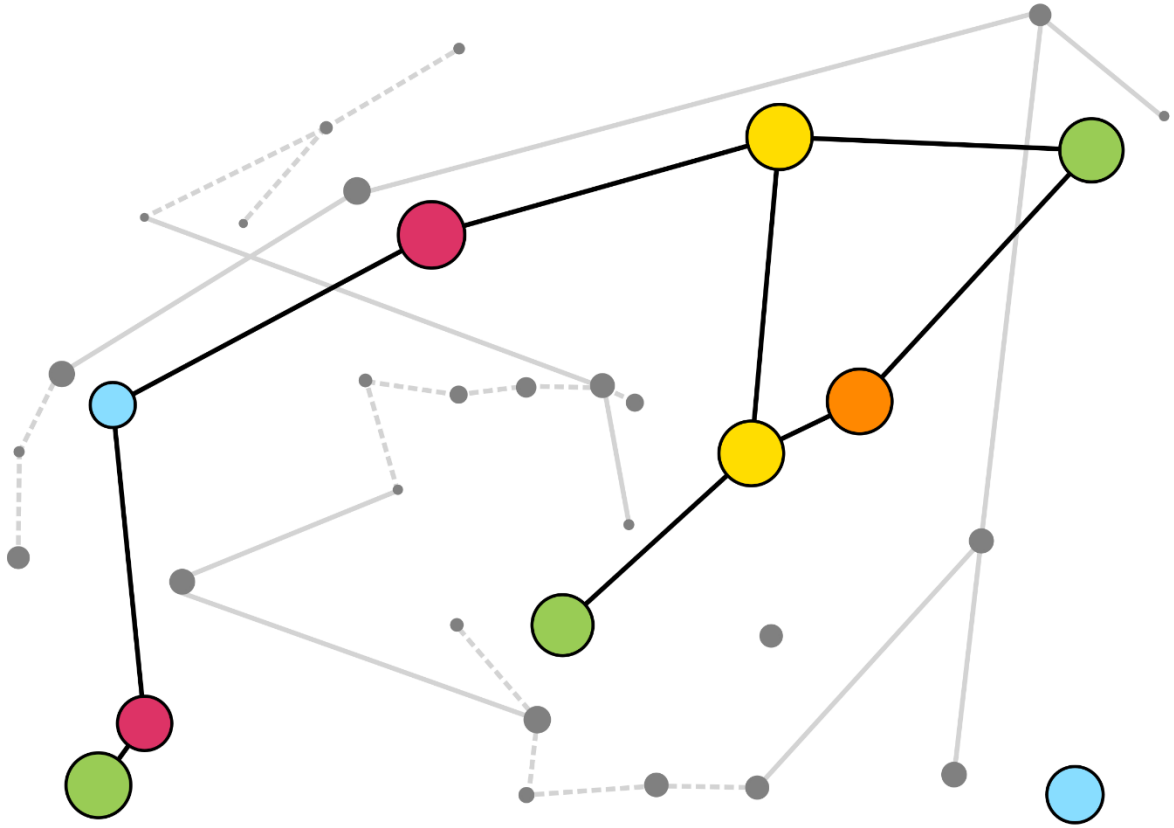
1. Test selection
2. **Test Env. Assignment**
3. Information Flow



Research Goal and Questions

Describe and improve upon industrial automated system level software testing of networked embedded systems.

1. Test selection
2. Test Env. Assignment
3. **Information Flow**



Abstract Information Flow. Illustration: Per Erik Strandberg

5. Research Methods

Photo: Per Erik Strandberg

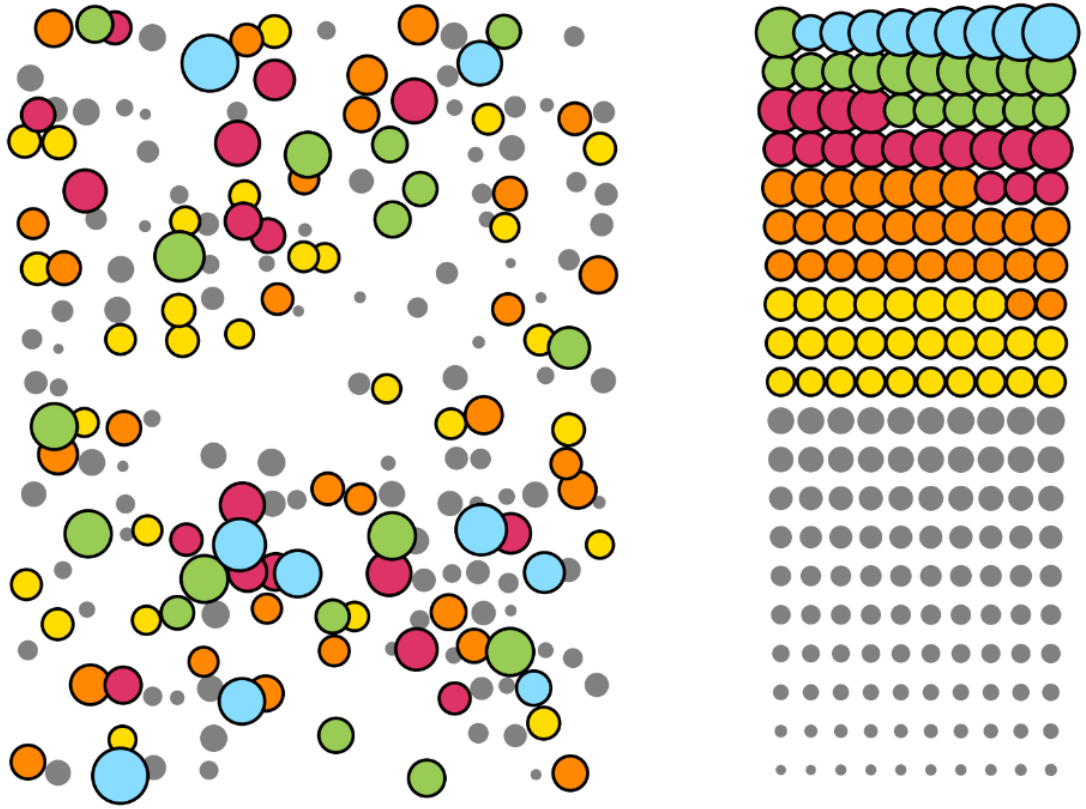
Methods/Algorithms/Tools

- Identify a problem
  - No time for testing
  - Shortcomings in Mapping
- Implement a tool
  - SuiteBuilder
  - Mapper

Victorinox Huntsman in red. Photo: Victorinox.com

Quantitative Empirical Studies

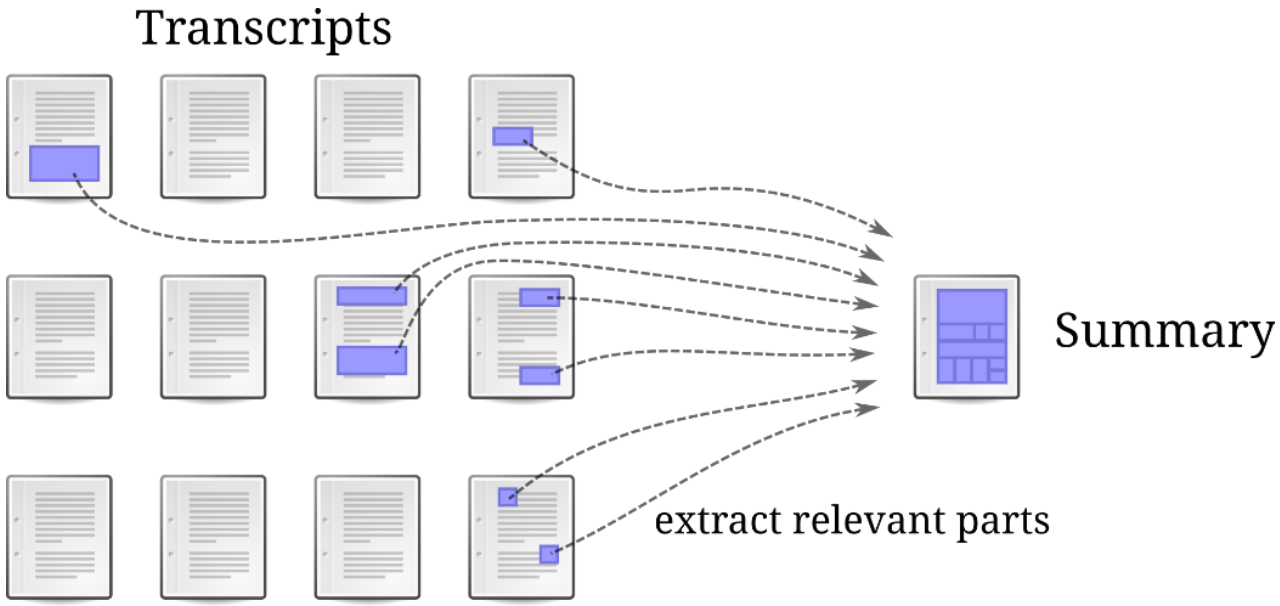
- Compare Before/After
- Measure Improvement
- 4 years of nightly test data
- 10000 mappings



Before and After. Illustration: Per Erik Strandberg

Qualitative Empirical Studies

- Interviews
- Transcription
- Thematic Analysis
  - [7] Braun & Clarke (2006)
  - Also used scripts as aid
- Overall guideline:
  - [40] Linåker et al. (2015)



Scripted Thematic Analysis. Illustration: Per Erik Strandberg

6. Research Contributions

Photo: Per Erik Strandberg

Four Papers in Thesis

A: System Level Regression Test Selection

P. E. Strandberg, D. Sundmark, W. Afzal, T. J. Ostrand, and E. J. Weyuker. Experience Report: Automated System Level Regression Test Prioritization Using Multiple Factors. In ISSRE'16, 2016. Best research paper.

Experience Report: Automated System Level Regression Test Prioritization Using Multiple Factors

B: Test Environment Assignment

P. E. Strandberg, T. J. Ostrand, E. J. Weyuker, D. Sundmark, and W. Afzal. Automated Test Mapping and Coverage for Network Topologies. In ISSTA'18, 2018.

Automated Test Mapping and Coverage for Network Topologies

C: Decision-making and Visualizations

P. E. Strandberg, W. Afzal and D. Sundmark. Decision Making and Visualizations Based on Test Results. In ESEM'18, 2018.

Decision Making and Visualizations Based on Test Results

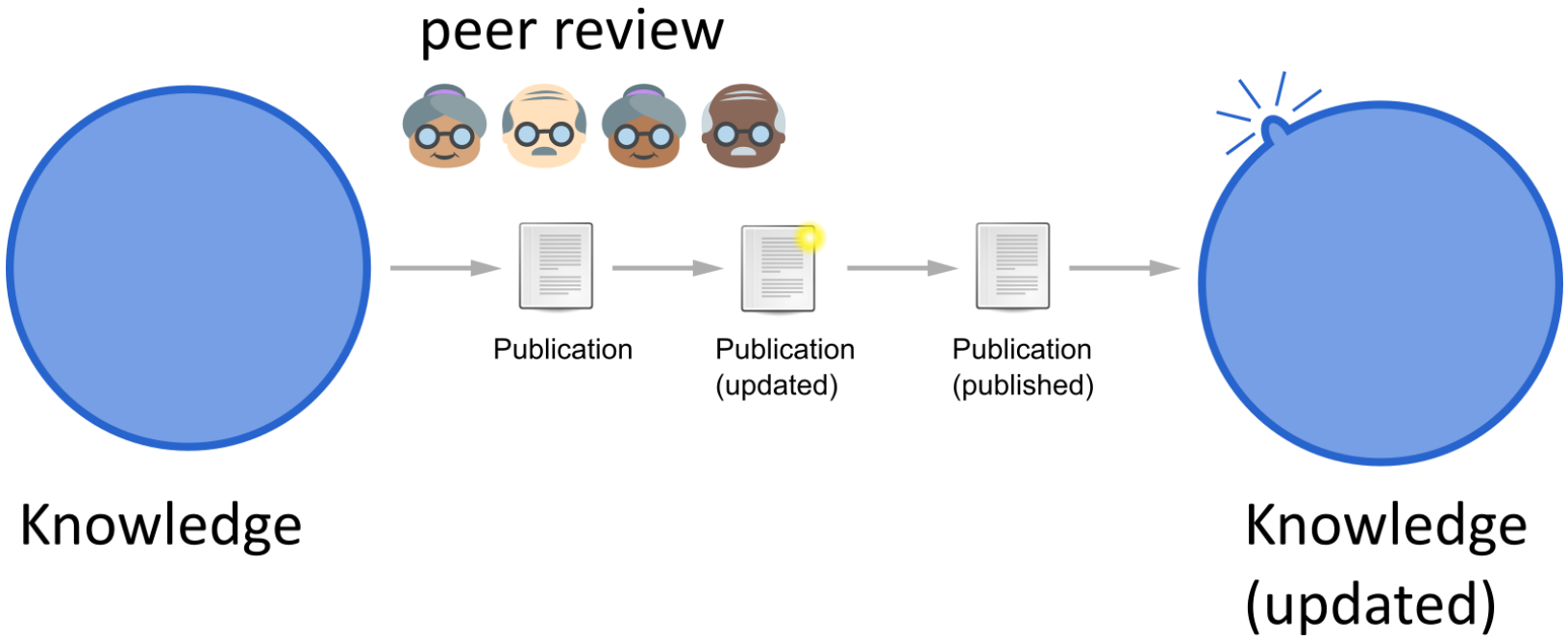
D: Information Flow

P. E. Strandberg, E. P. Enoiu, W. Afzal, D. Sundmark, and R. Feldt. Information Flow in Software Testing – An Interview Study with Embedded Software Engineering Practitioners. In revision.

Information Flow in Software Testing – An Interview Study with Embedded Software Engineering Practitioners

Why Publications?

- One publication
- One kilogram
- One man-month
- One line of code
- One bug
- One “like”

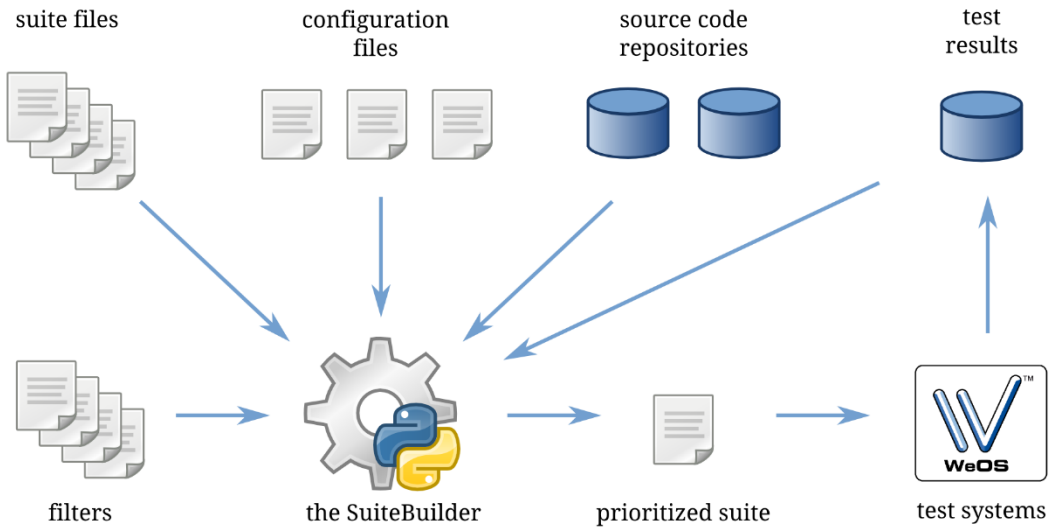


Increasing Collective Knowledge. Illustration: Per Erik Strandberg

westermo

Paper A:
System Level Regression Test Selection

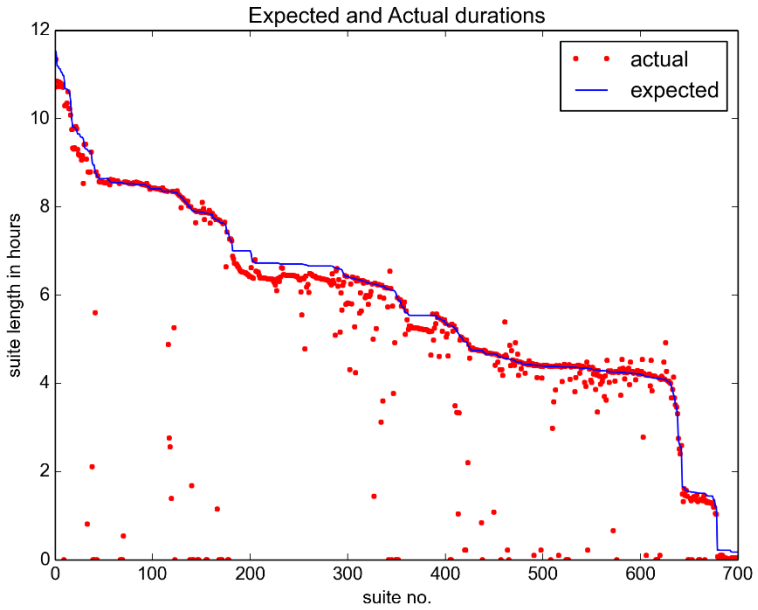
- Problem
  - Nightly testing does not finish on time
  - Manual work (and omitted test cases)
  - No priority
- Approach
  - SuiteBuilder tool
  - Prioritize tests
  - Estimate time and stop



Overall Data Flow in SuiteBuilder. Figure from Paper A

Paper A:
System Level Regression Test Selection

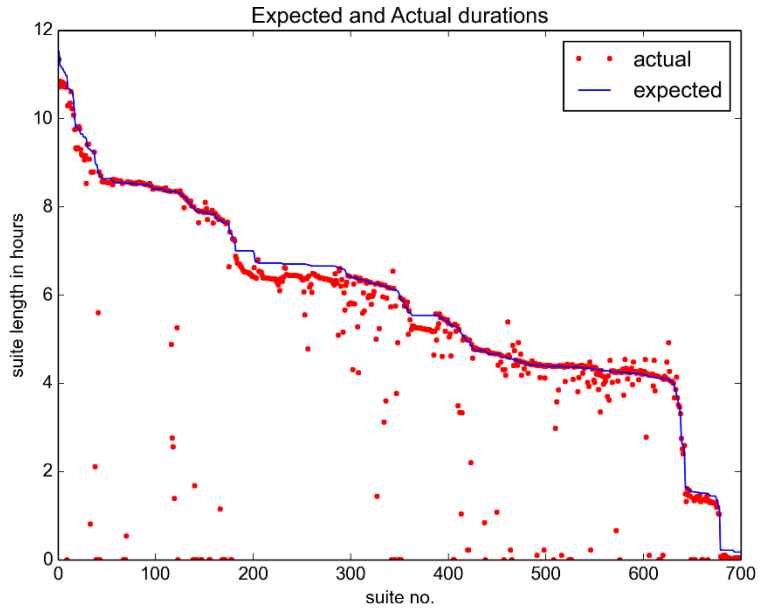
- Results
  - Suites finish on time



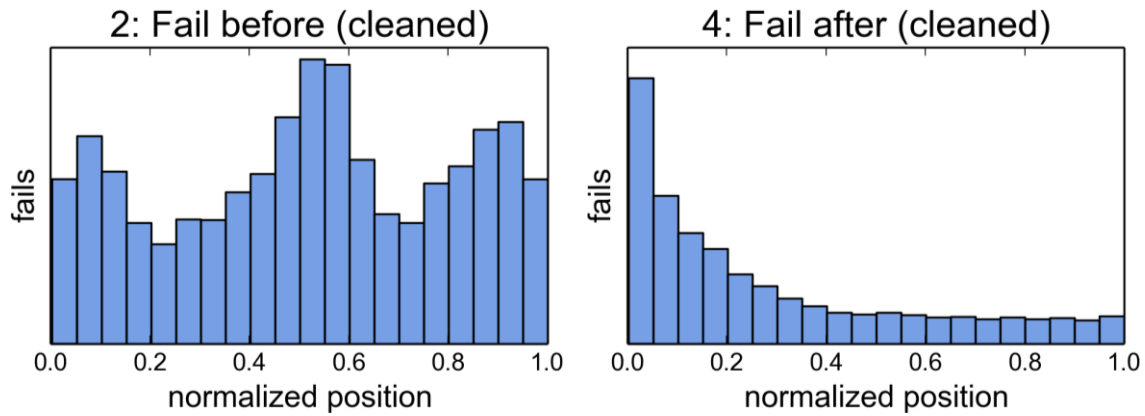
Figures from Paper A

Paper A:
System Level Regression Test Selection

- Results
  - Suites finish on time
  - Find faults early

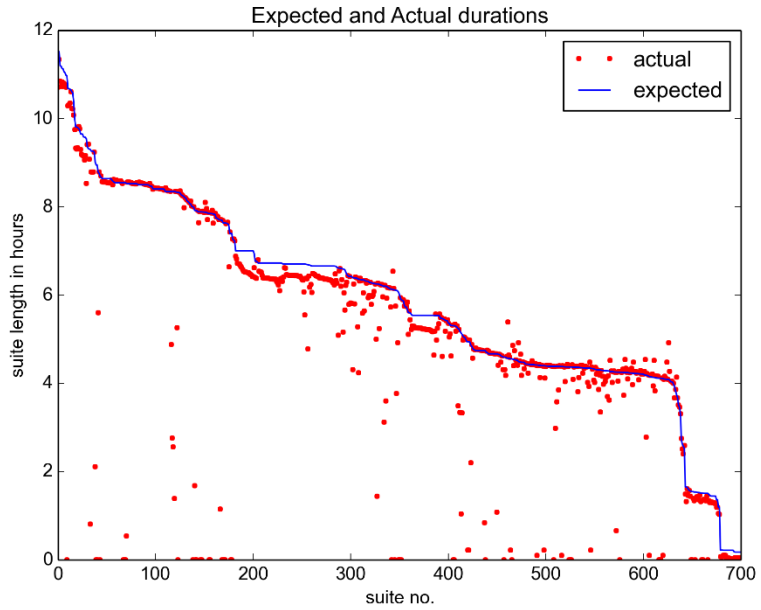


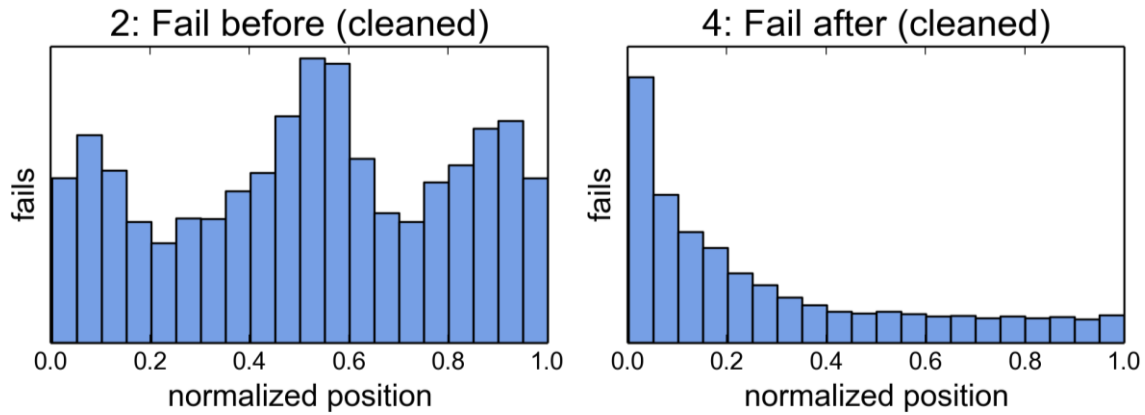
Figures from Paper A



Paper A:
System Level Regression Test Selection

- Results
  - Suites finish on time
  - Find faults early
  - Less manual work
- Contributions
  - Automated regression test selection on a *system-level*
  - Solution integrated in nightly testing
  - Industry-grade implementation and evaluation

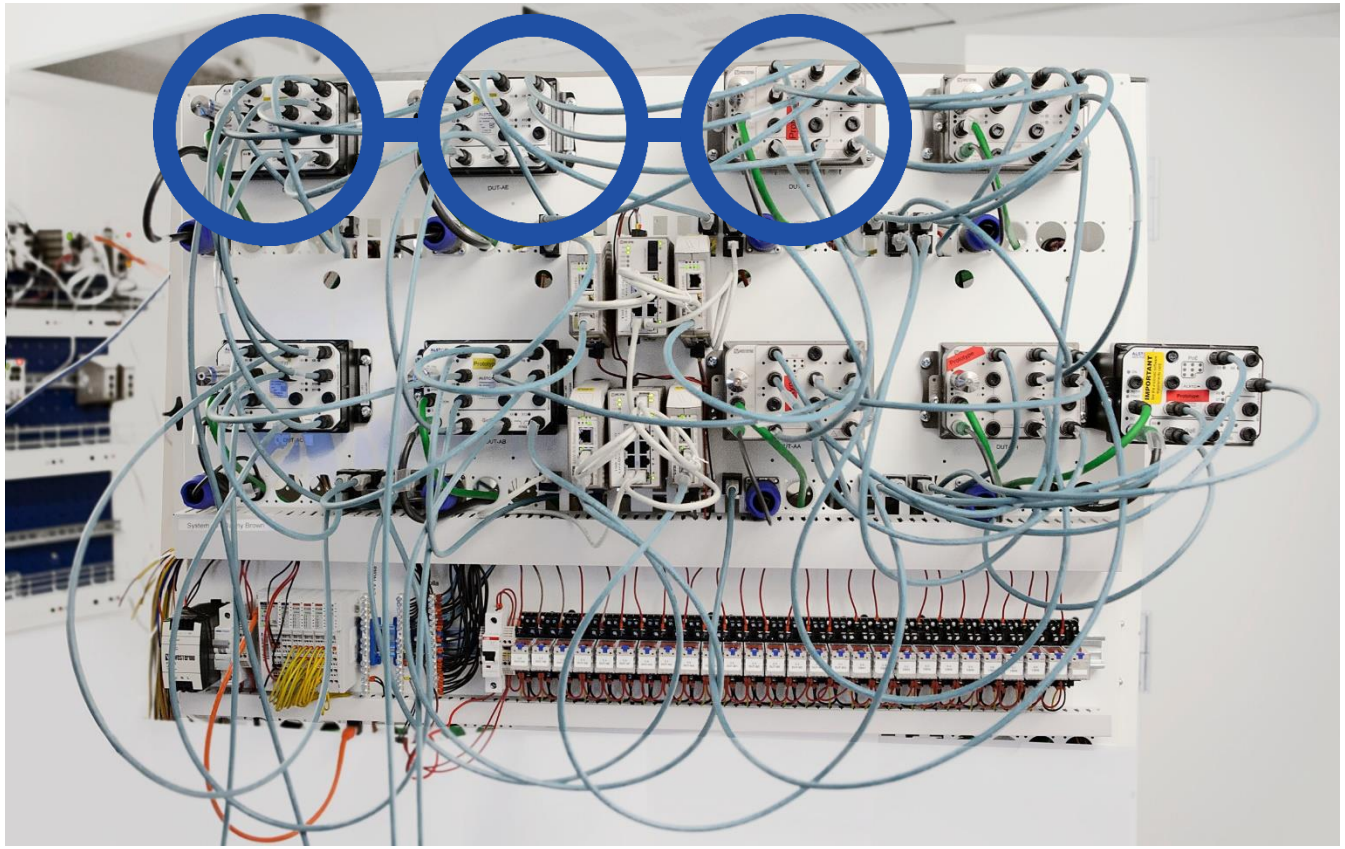




Figures from Paper A

Paper B: Test Environment Assignment

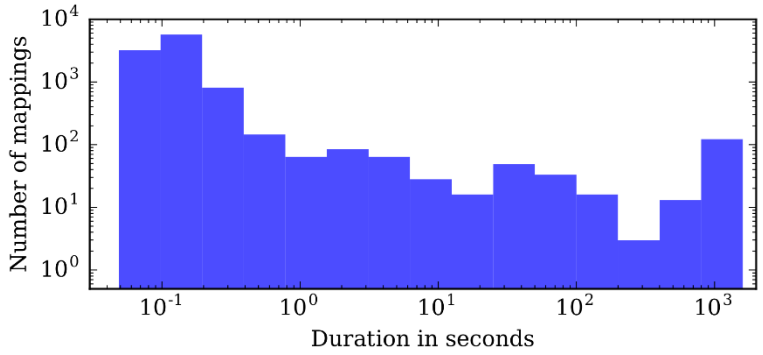
- Problem
  - Performance (old solution)
  - Always the same mapping
- Approach
  - Graph Theory: Subgraph isomorphism problem
  - Search for a mapping
  - Reduce size of search space
  - Remember old mappings and search for unused parts
  - Evaluate with 10000 pairs (17 test systems, 607 test cases)

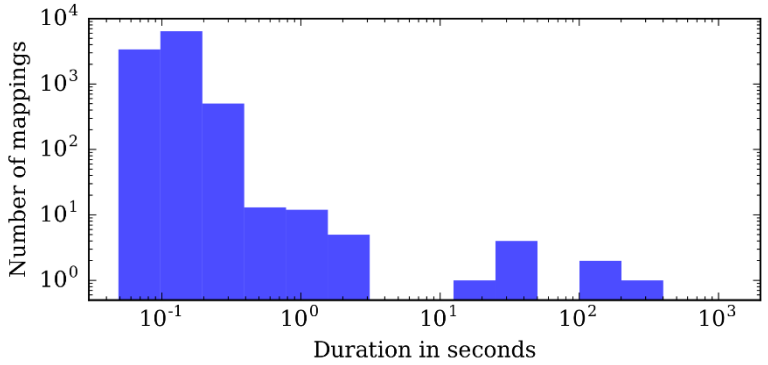


westermo

Paper B: Test Environment Assignment

- Results
  - Speedup x 80



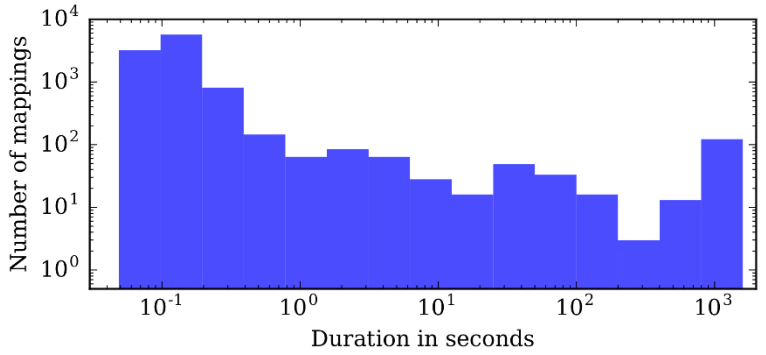


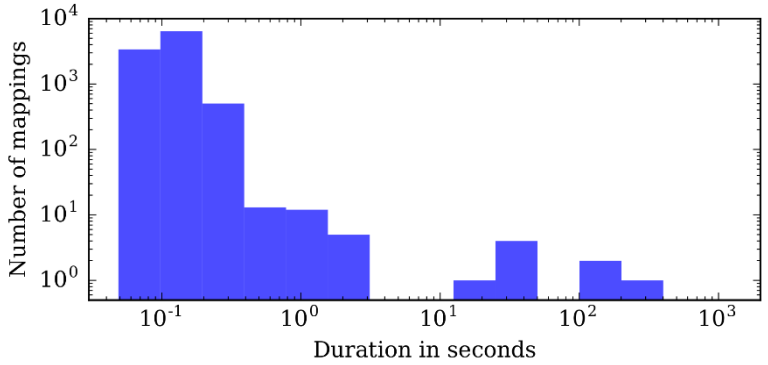
Figures from Paper B

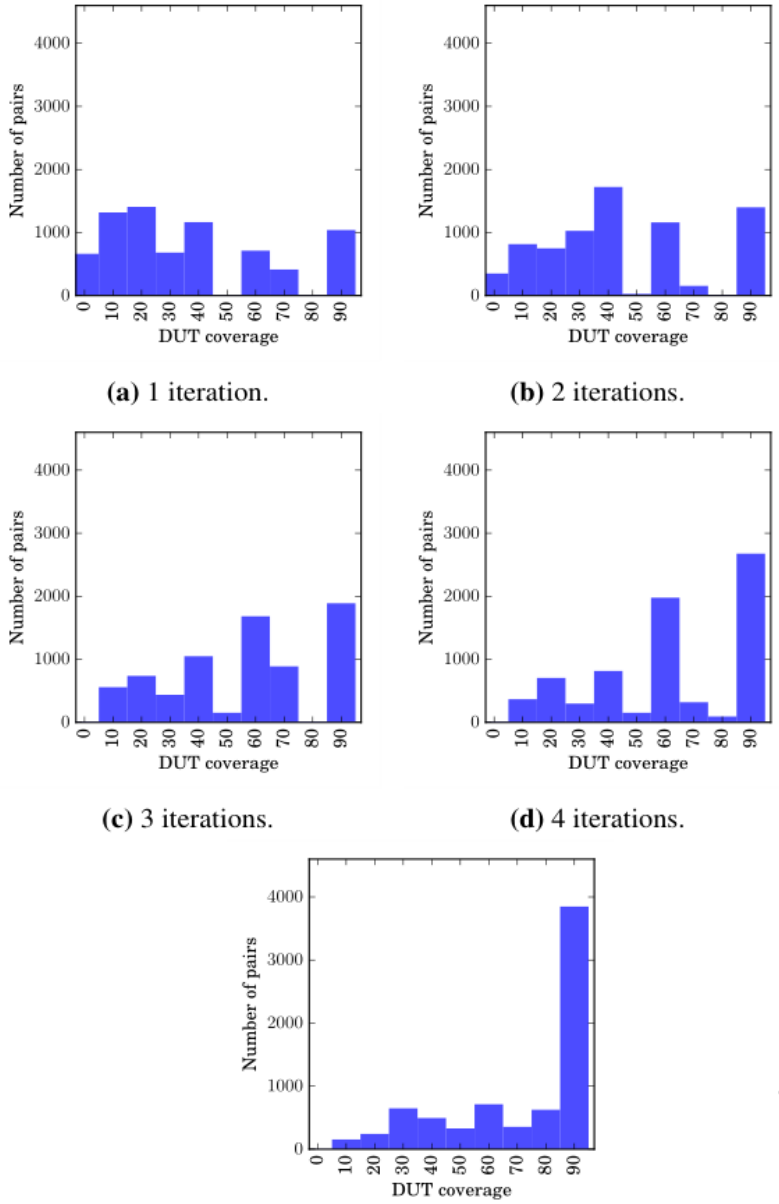
westermo

Paper B: Test Env. Ass.

- Results
  - Speedup x 80
  - Coverage:
    from median of 33%
    to 100%
    in 5 iterations



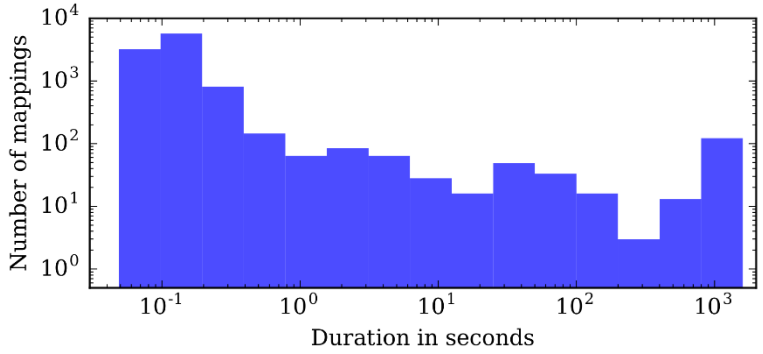


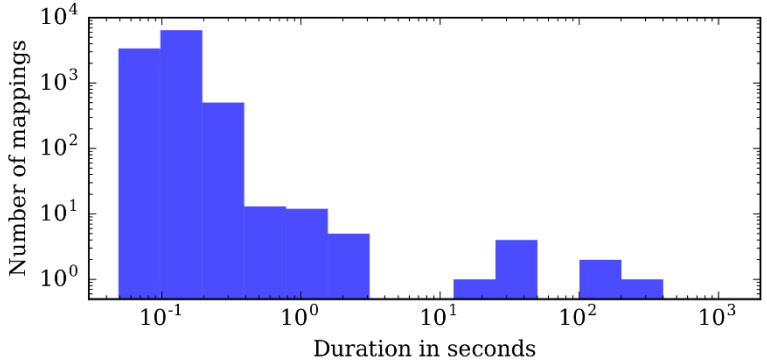


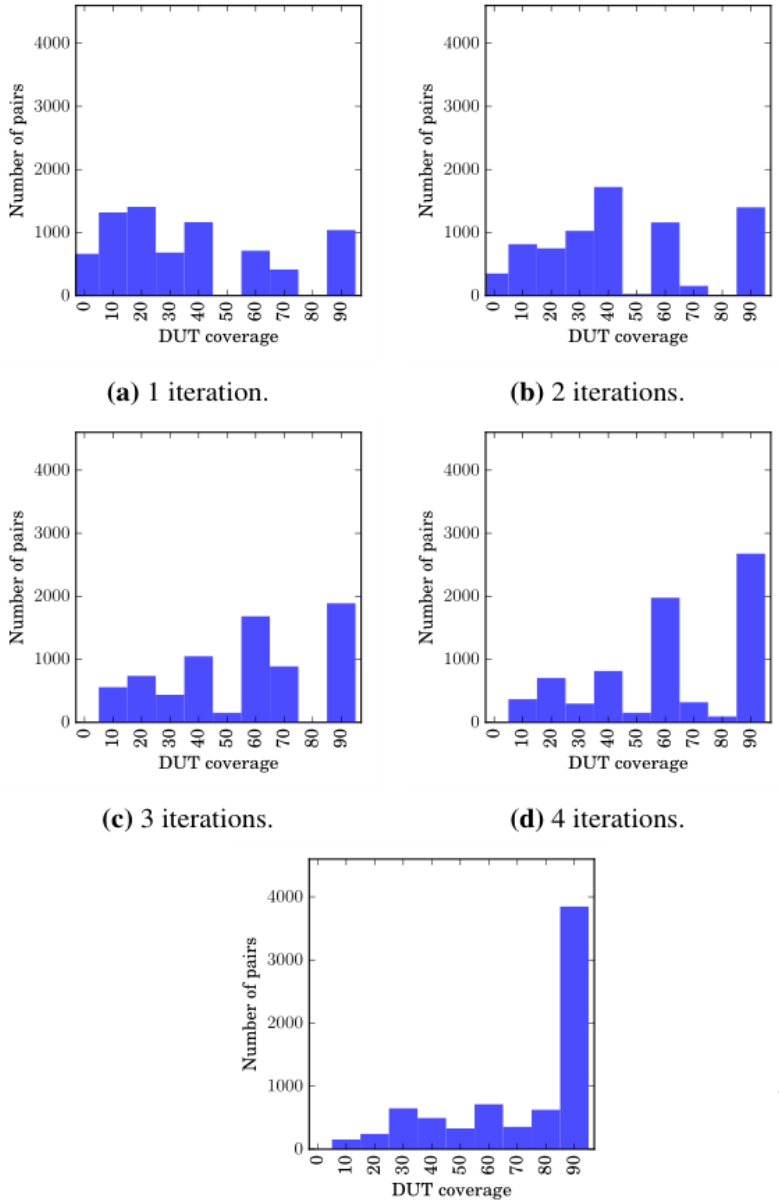
Figures from Paper B

Paper B: Test Env. Ass.

- Results
  - Speedup x 80
  - Coverage:
    from median of 33%
    to 100%
    in 5 iterations
- Contributions
  - Industry-grade implementation and evaluation
  - First (?) publication with working solution for test environment assignment



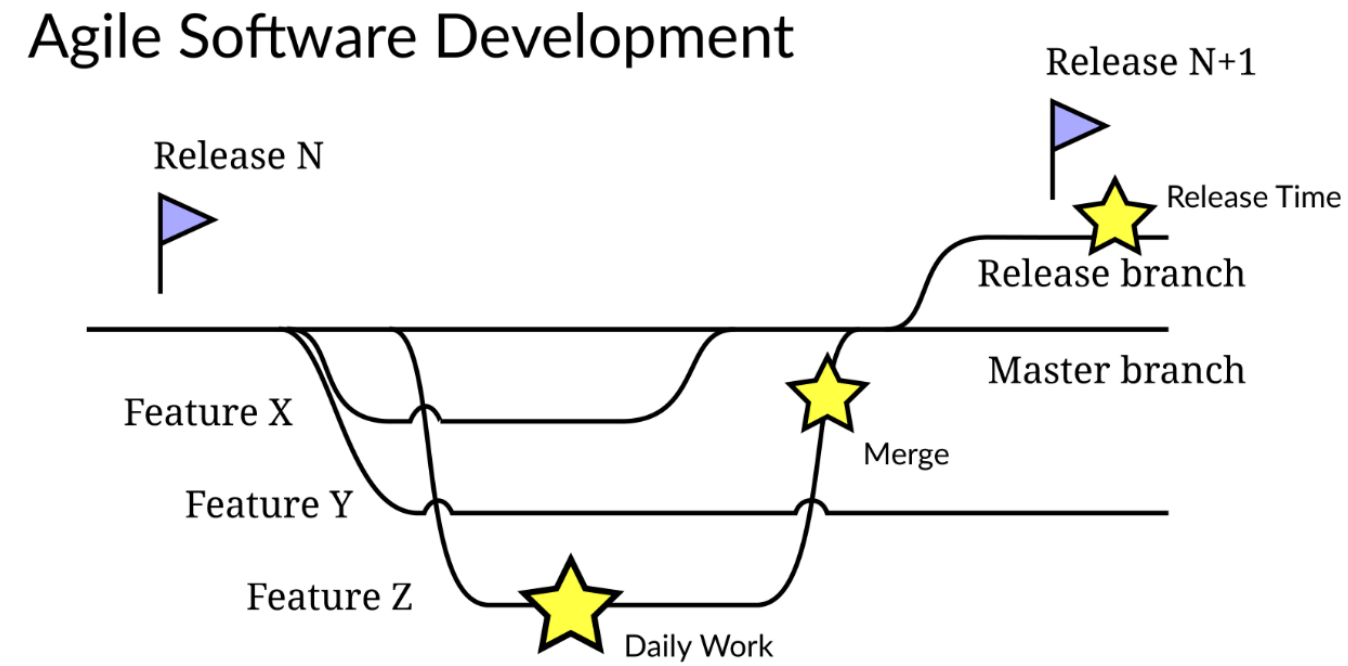




Figures from Paper B

Paper C: Decision-making and Visualizations

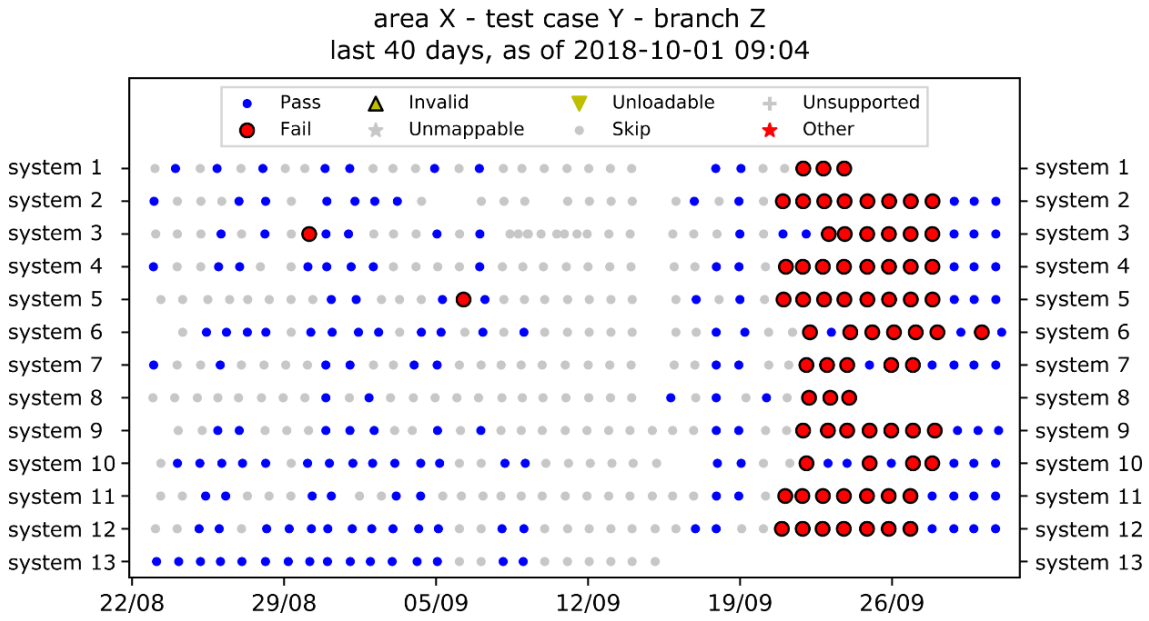
- Focus
  - How are visualizations made?
  - What is the perceived value?
  - (In daily work, at merge time, and at release time)
- Approach
  - Exploratory
  - Descriptive
  - Embedded case study



Software Development in Branches. Modified from figure in Paper C

Paper C: Decision-making and Visualizations

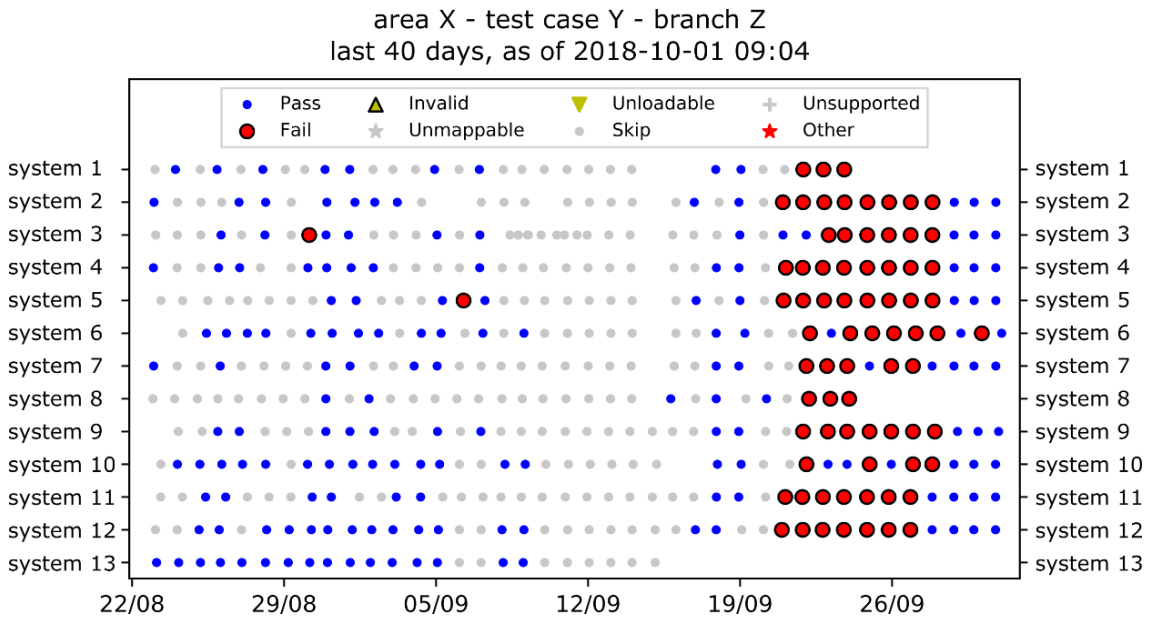
- Results and Contributions
  - Visualizations and scripts for exploring test results.

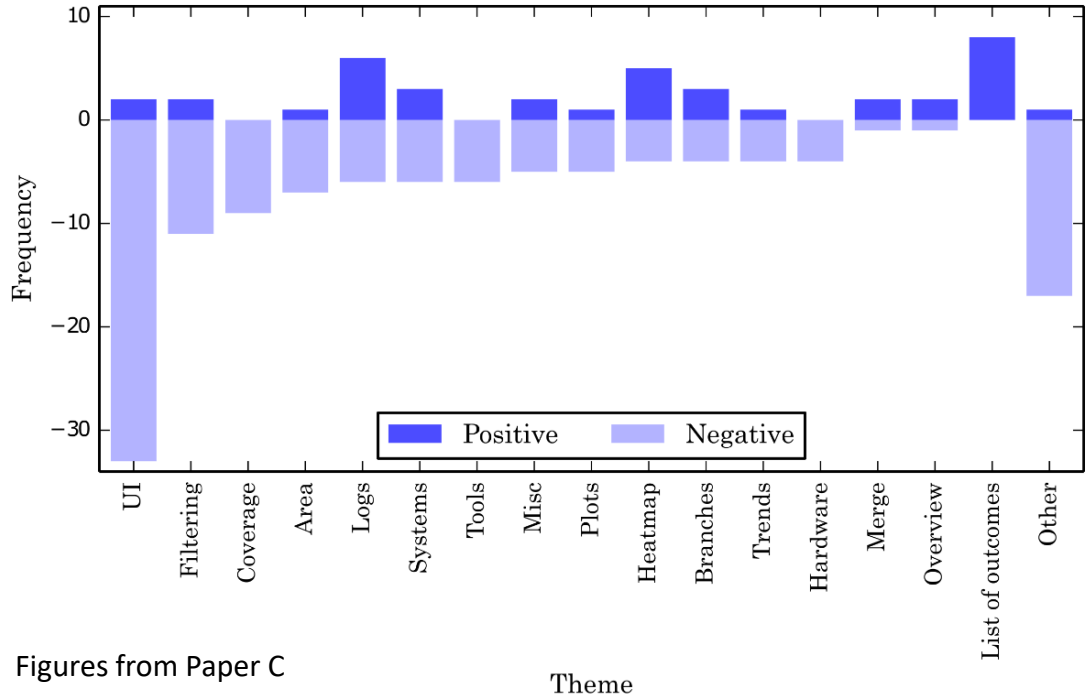


Figures from Paper C

Paper C: Decision-making and Visualizations

- Results and Contributions
  - Visualizations and scripts for exploring test results.
  - Positive/Negative

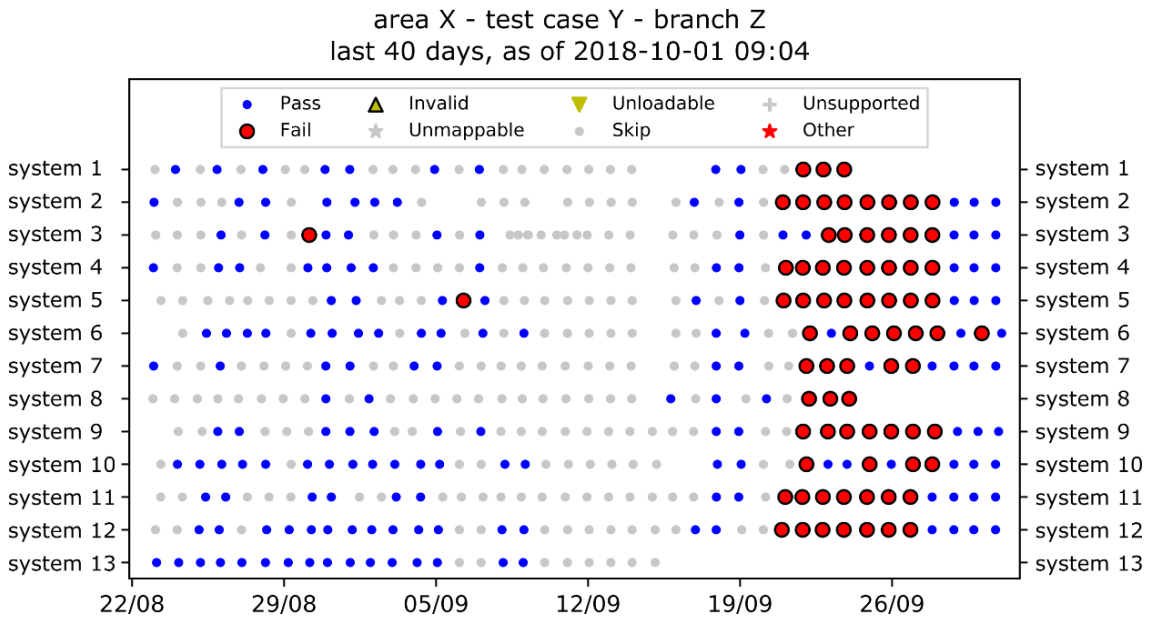


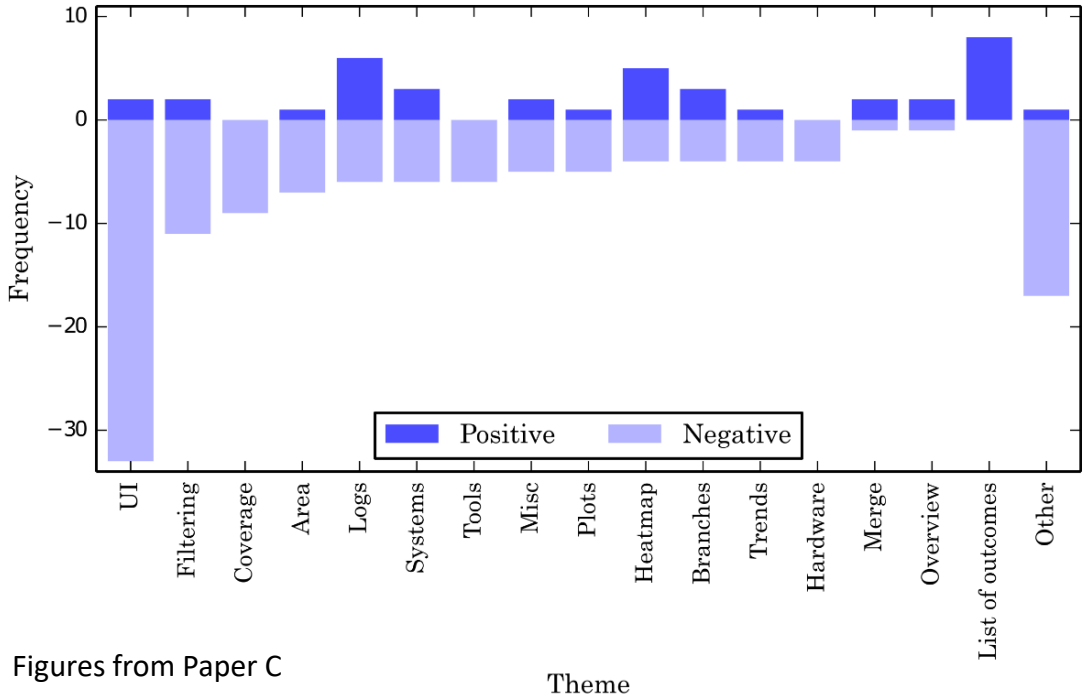


Figures from Paper C

Paper C: Decision-making and Visualizations

- Results and Contributions
  - Visualizations and scripts for exploring test results.
  - Positive/Negative
  - Experiences and Evaluation at Westermo
    - Six years of usage.
    - User Stories (Appendix C)





Figures from Paper C

Paper D: Flow of Information SW Testing

- Question:
  - What is the overall flow of information in software testing?

Paper D: Flow of Information SW Testing

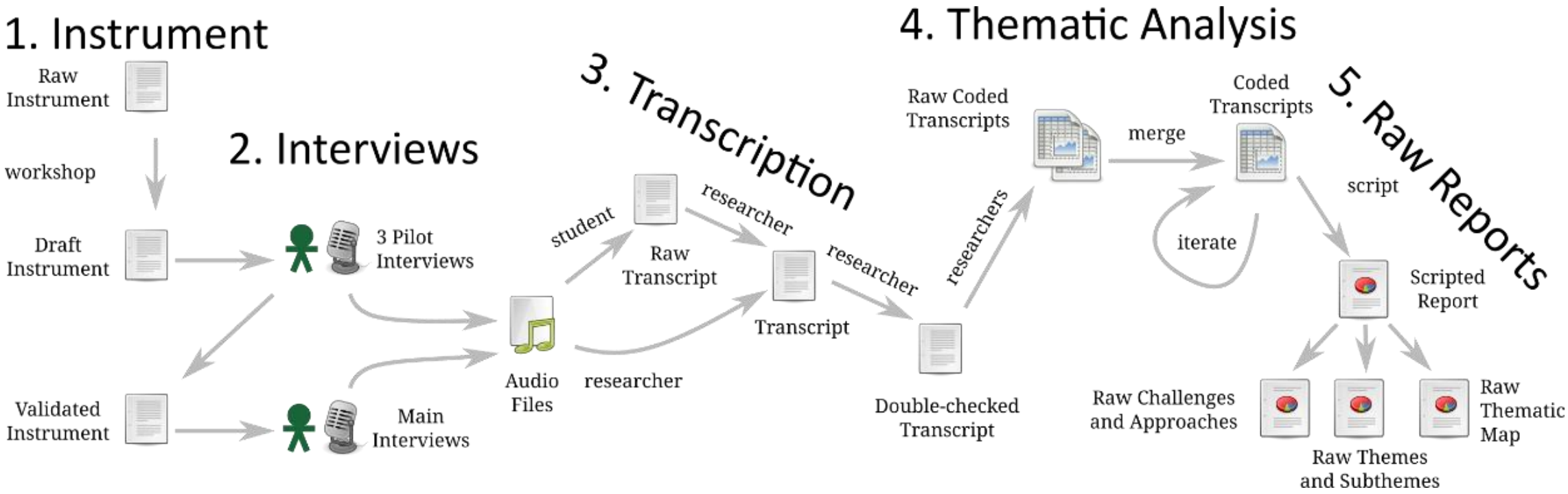
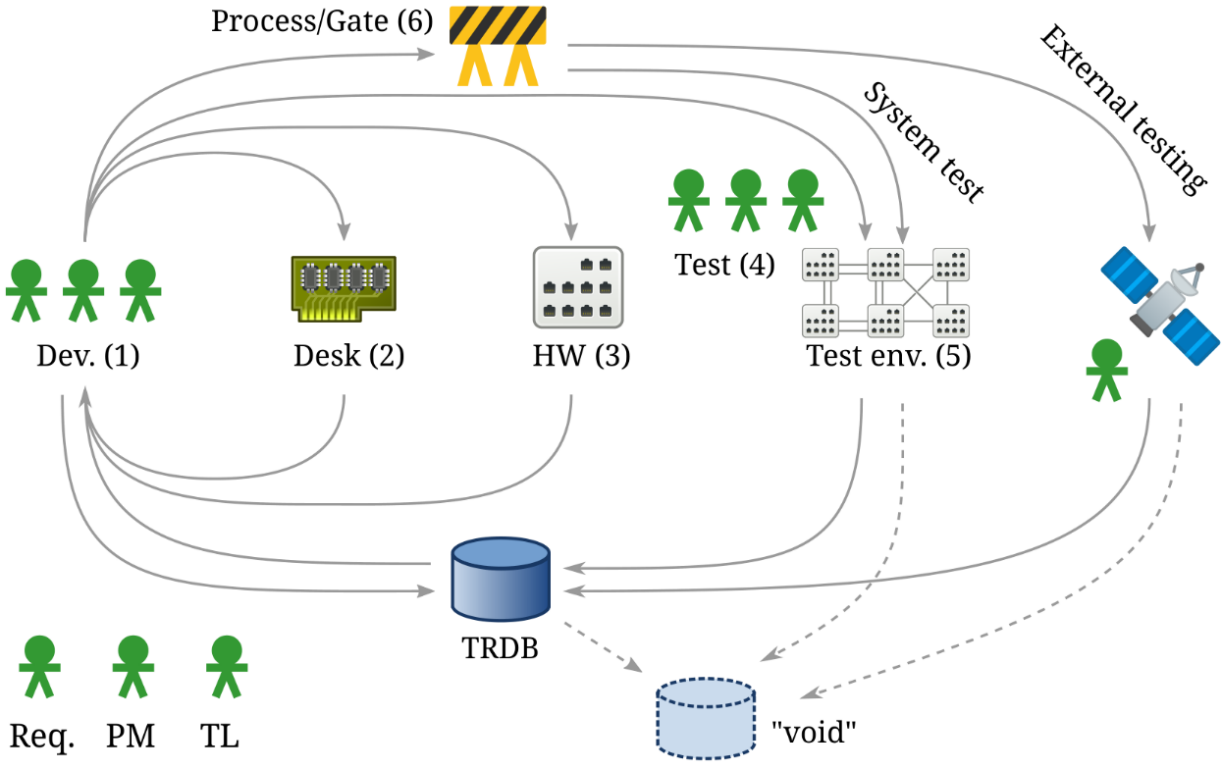


Figure modified from Paper D

- Question:
  - What is the overall flow of information in software testing?
- Approach:
  - Ambitious Interview Study
  - *5 Companies*, 12 Practitioners

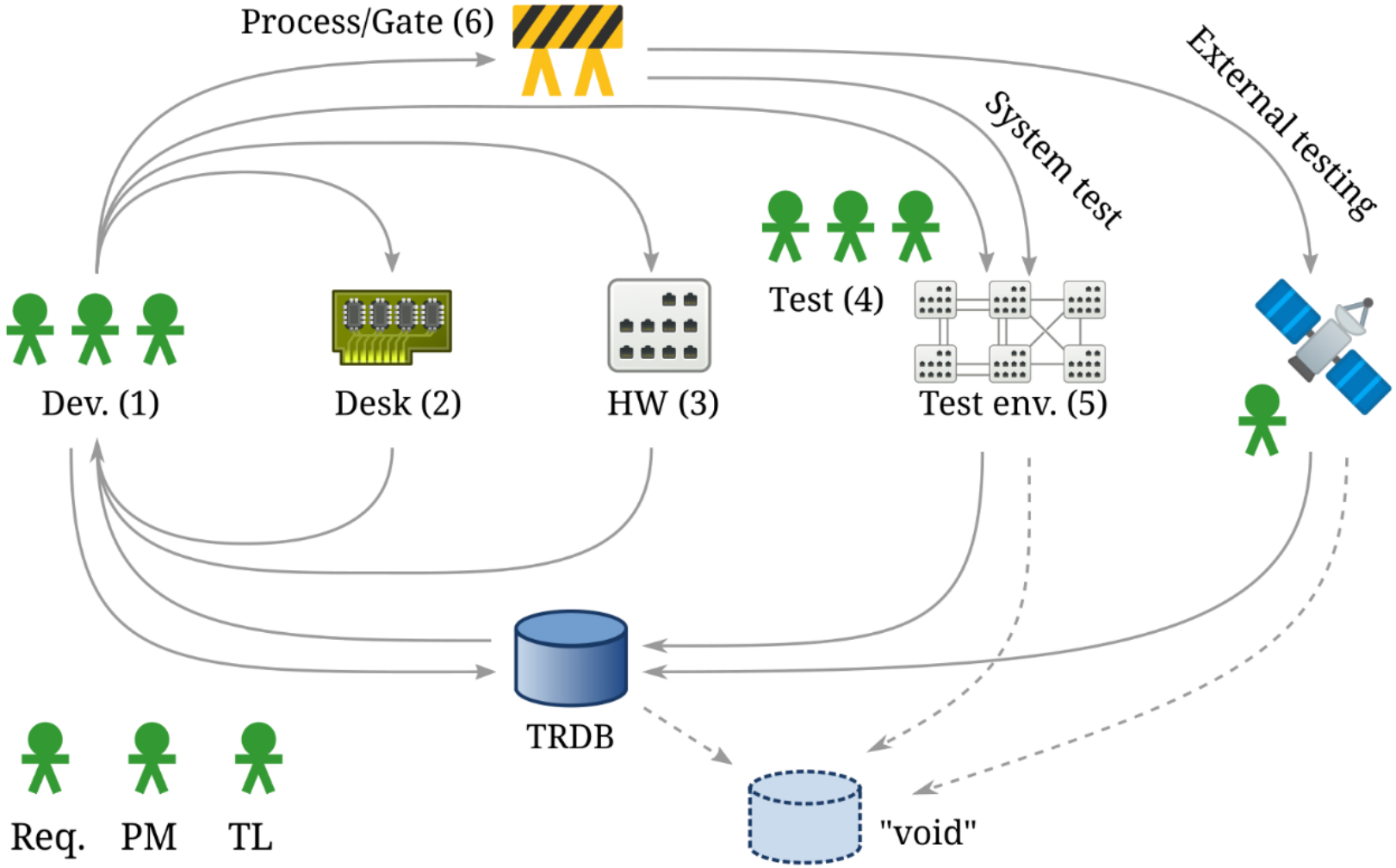
Paper D: Flow of Information SW Testing

- Results and Contributions:
  - Overall Information Flow Model
  - Challenges
  - Approaches
  - Themes



Flow of Information in Software Testing. Image from Thesis Introduction, modified from figure in Paper D

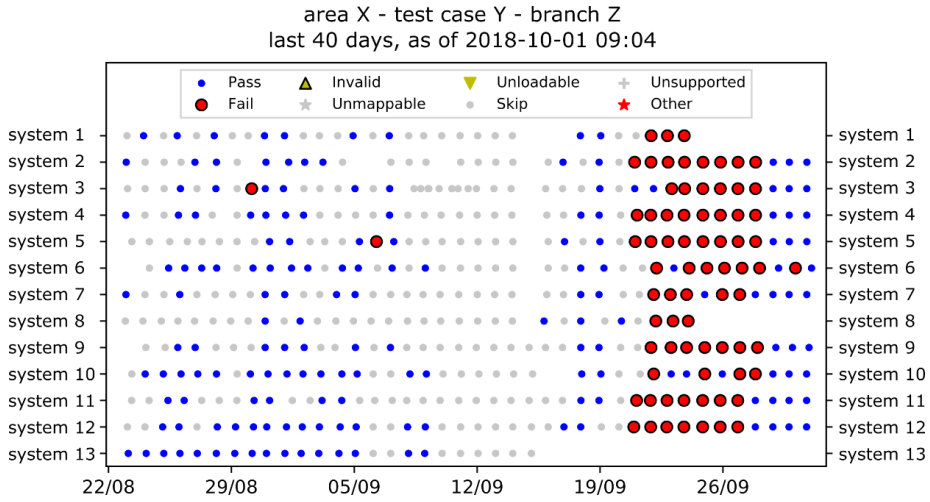
Paper D: Flow of Information SW Testing



7. Future Research

Photo: Per Erik Strandberg

Future Research



- Improve Test Results Exploration
  - Learn from Paper C and D
  - Learn from others [6, 15, 18, 26, 42, 46, 47, 61]
  - Combine implementation and evaluation

- Will it be harder to understand the results when…
  - Paper A: Not all tests run (and not in the same order)?
  - Paper B: Test cases run in different ways over time?

- Flaky Tests

westermo

8. Summary

Photo: Per Erik Strandberg

Recap

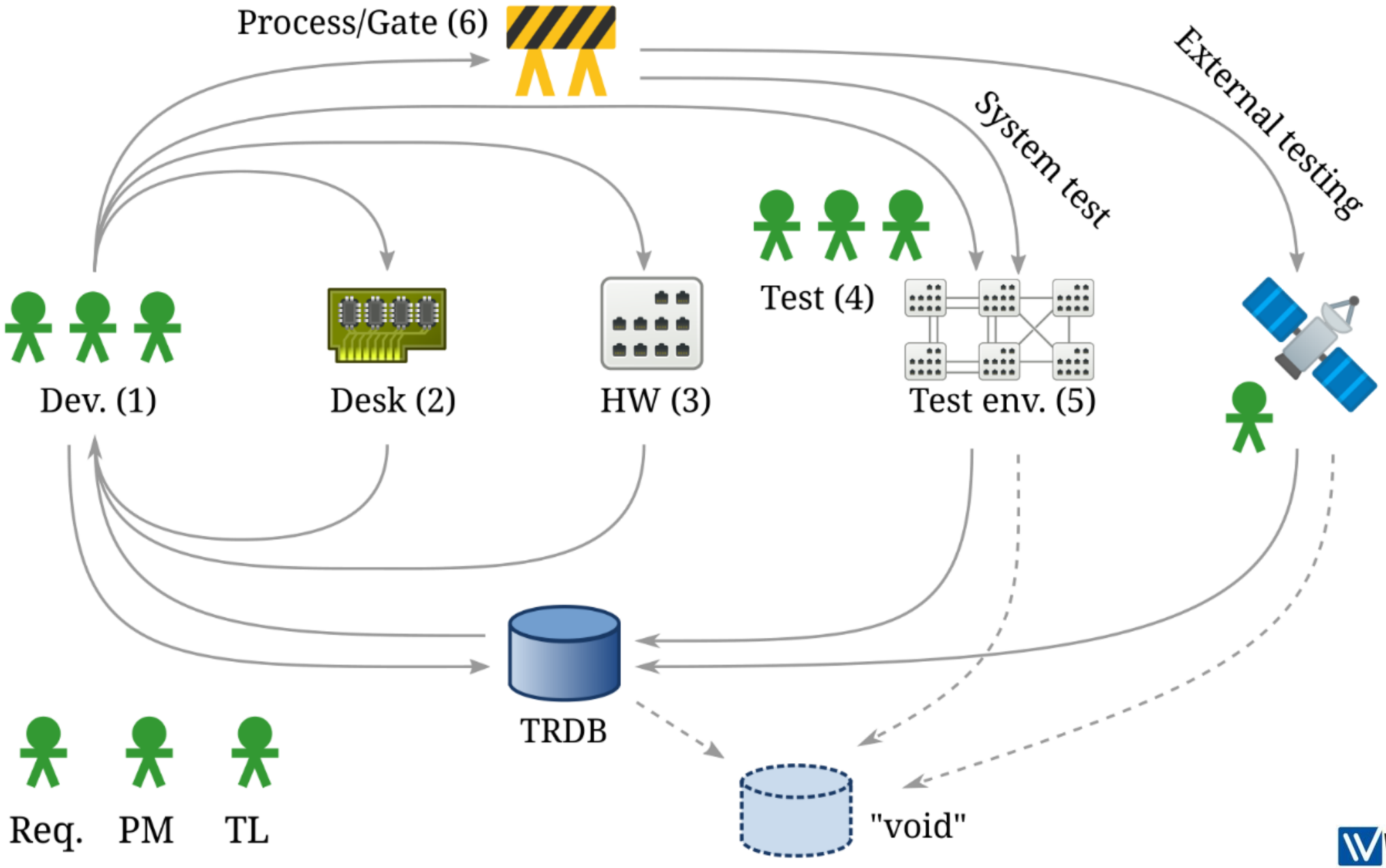
1. Context
2. What are networked embedded systems?
3. Testing networked embedded systems
4. Research Questions
5. Research Methods
6. Research Contributions
7. Future Research
8. Summary

Looking back. Photo: Per Erik Strandberg

Acknowledgments

- This research financed by
  - Westermo Research and Development AB
  - Swedish Knowledge Foundation
    - 20150277 (ITS ESS-H)
    - 20160139 (TESTMINE)

- Co-authors been financed by:
  - Swedish Knowledge Foundation
    - 20130085 (TOCSYC)
    - 20130285 (Volvo Chair)
  - Swedish Research Council
    - 621-2014-4925 (EXACT)
  - European Union's Horizon 2020
    - 737494 (MegaMART2)


Process/Gate (6)
External testing
System test
Test (4)
Dev. (1)
Desk (2)
HW (3)
Test env. (5)
TRDB
"void"
Req.
PM
TL

Automated System Level Software Testing of Networked Embedded Systems

Mälardalen University Licentiate Thesis 275

Per Erik Strandberg