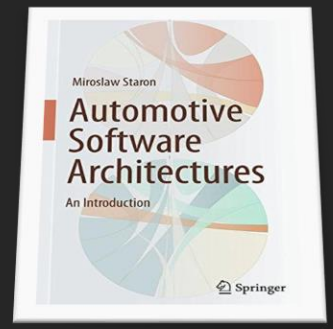
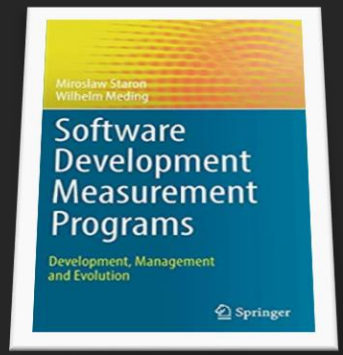
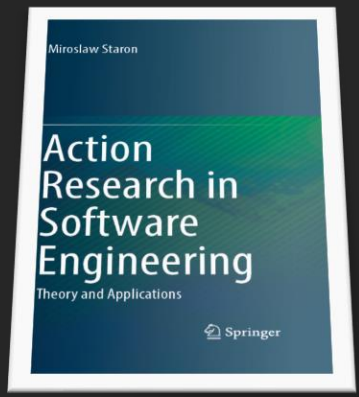
UNIVERSITY OF GOTHENBURG



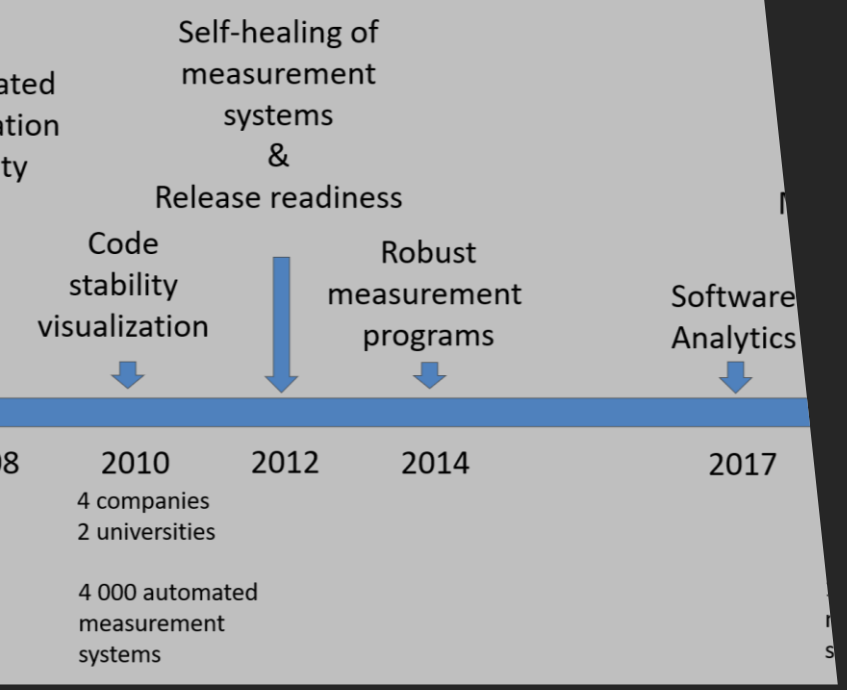




# INFORMATION NEEDS FOR SAFE TEAMS AND RELEASE TRAIN MANAGEMENT:
## A DESIGN SCIENCE RESEARCH STUDY

MIROSLAW STARON, CHALMERS | UNIVERSITY OF GOTHENBURG
WILHELM MEDING, ERICSSON AB
POUPAK BANIASAD, SOFTWARE CENTER

# Who am I?

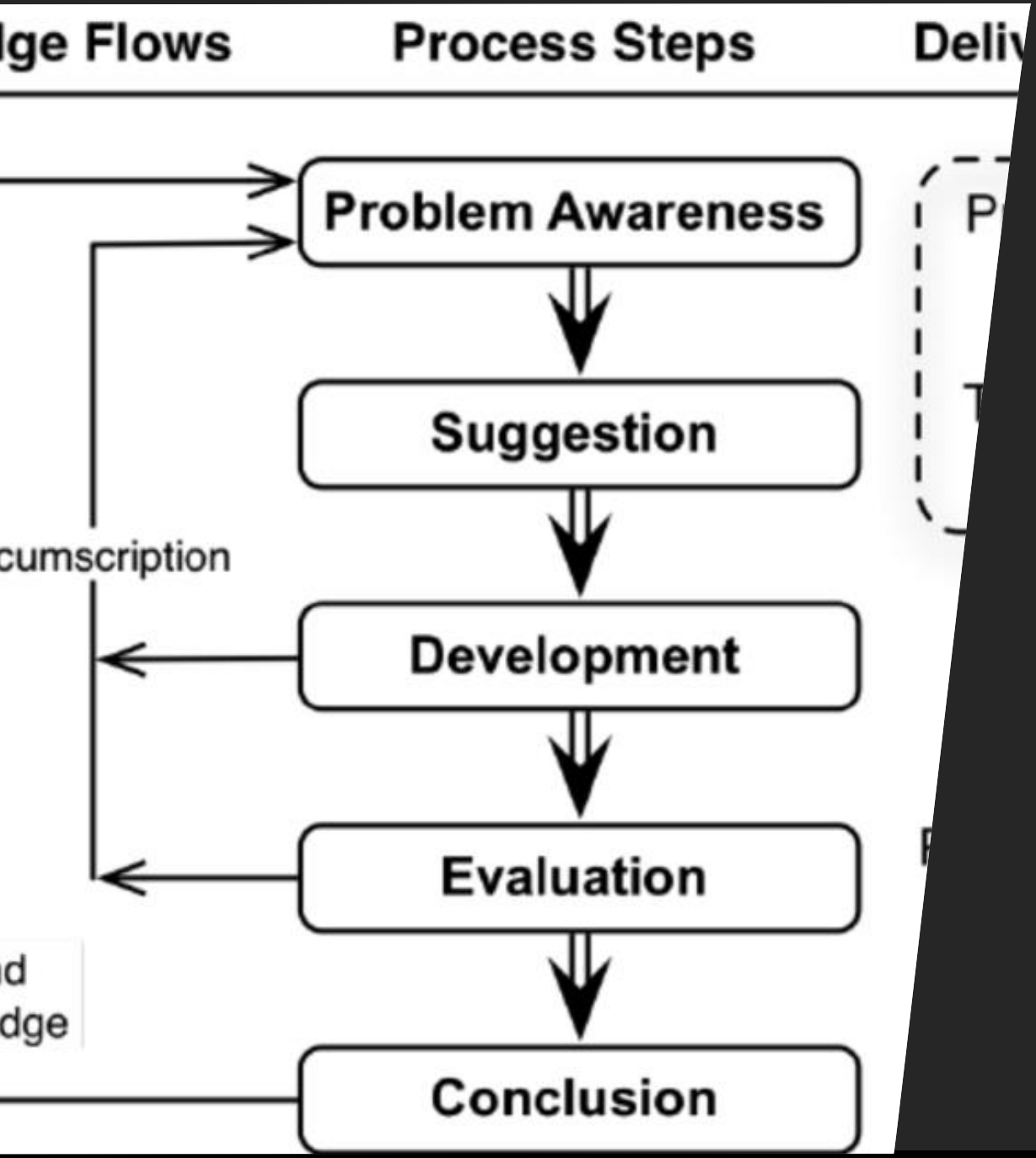
- Professor of Software Engineering at Chalmers | University of Gothenburg
- Specialization in software measurement
  - Machine learning in software engineering
  - Autonomous artificial intelligence based measurement
  - Measurement knowledge discovery
  - Simulation of outcome before decision formulation
  - Metrological foundations of measurement reference etalons
- Actively working with the standards
  - ISO/IEC 15939 - Software and Systems Engineering - Measurement Processes
  - ISO/IEC 25000 (series) - Software Quality Requirements and Evaluation (SQuaRE)
  - ISO/IEC 14598 - Information Technology - Software Product Evaluation
- Software Center – a collaboration between 13 companies and 5 universities





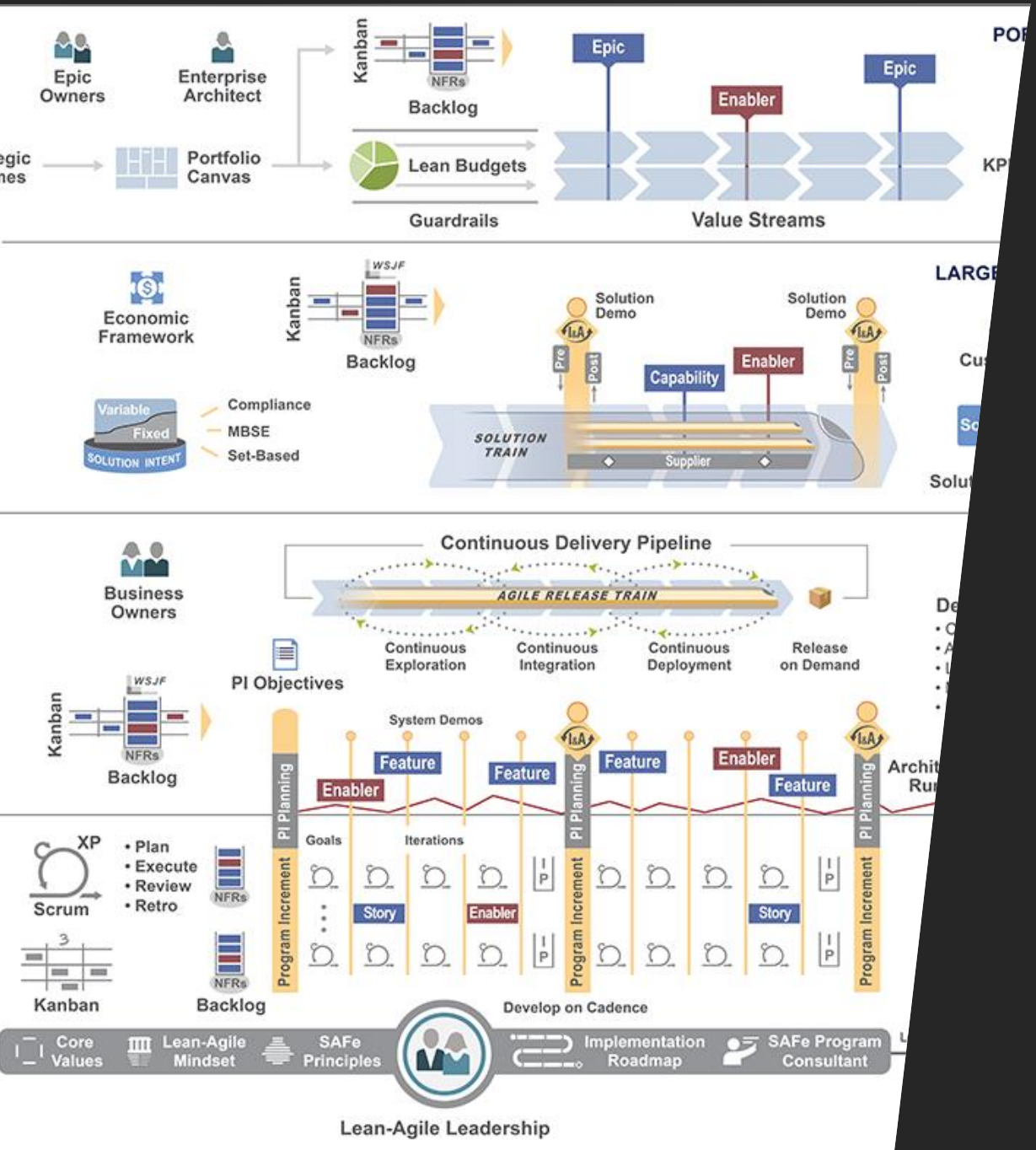
## Software Center – a *collaboration* between 12 companies and 5 universities

- We work together to accelerate the adoption of novel approaches to software engineering
- Our mission with the Software Center is to contribute to maintaining – and strengthen – Sweden's leading position in engineering industrial software-intensive products
- The metrics theme is led by Miroslaw and Wilhelm




# Problem formulation

- What are the information needs of SAFe teams and train management in the automotive domain?
- Context
  - Company in embedded systems domain
  - Mature software product (over 10 years on the market)
  - Mature development organization
    - In a transition between project-based and Agile-based (SAFe) software development

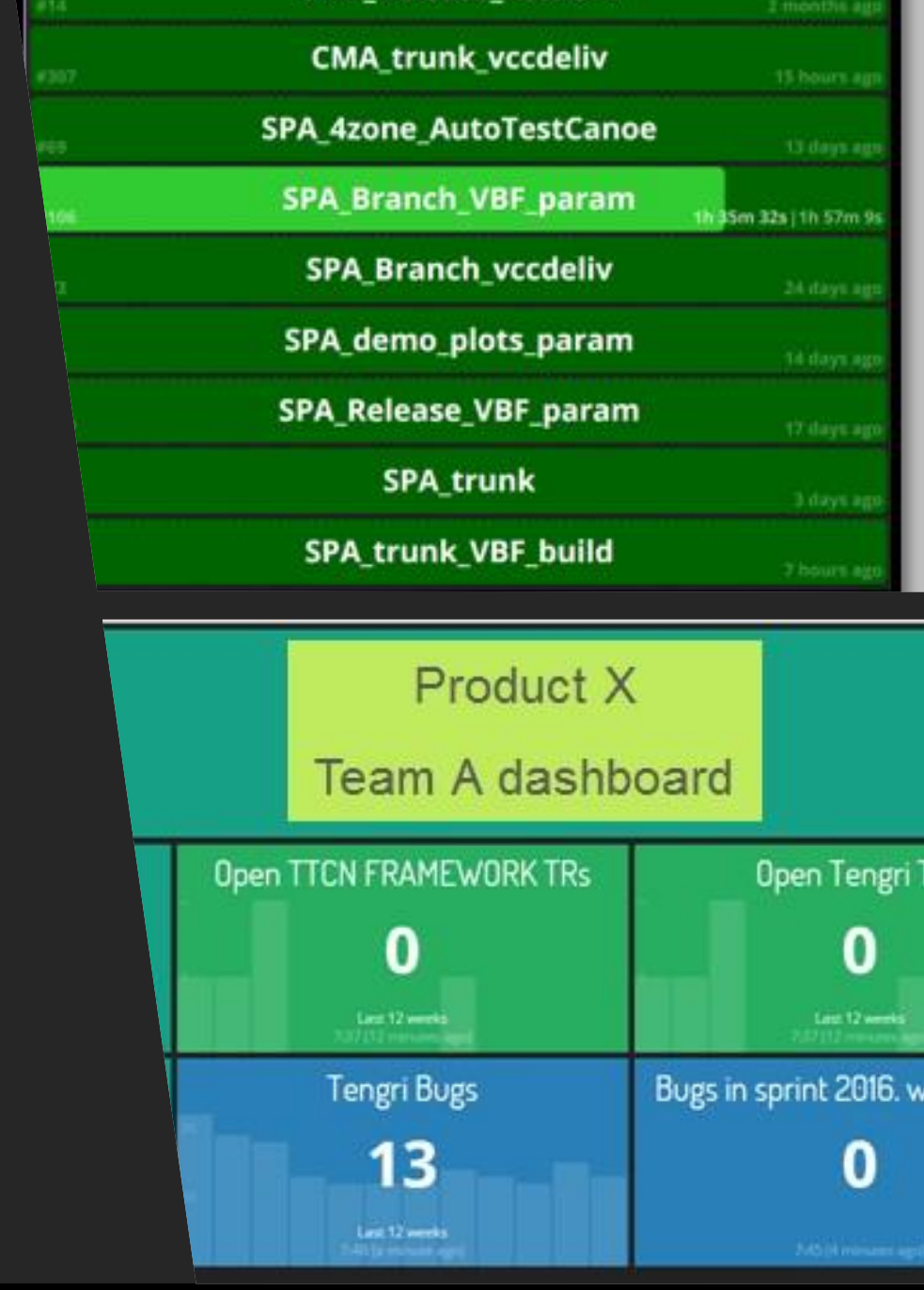



# SAFe

- Software development paradigm based on Agile principles
- Suited for companies in embedded systems domains
- Fits with the needs of ISO 262626
- Comes with prescribed sets of measures to monitor
- Brings in empowerement into large scale software development

# Our previous research

- Empowered teams have specific needs (not only the ones prescribed by processes)
- Product increments are equally important as project increments
- Speed and quality trumps all other aspects



# Measures and information needs prescribed by SAFe

**Table 1.** Relevant Portfolio measures

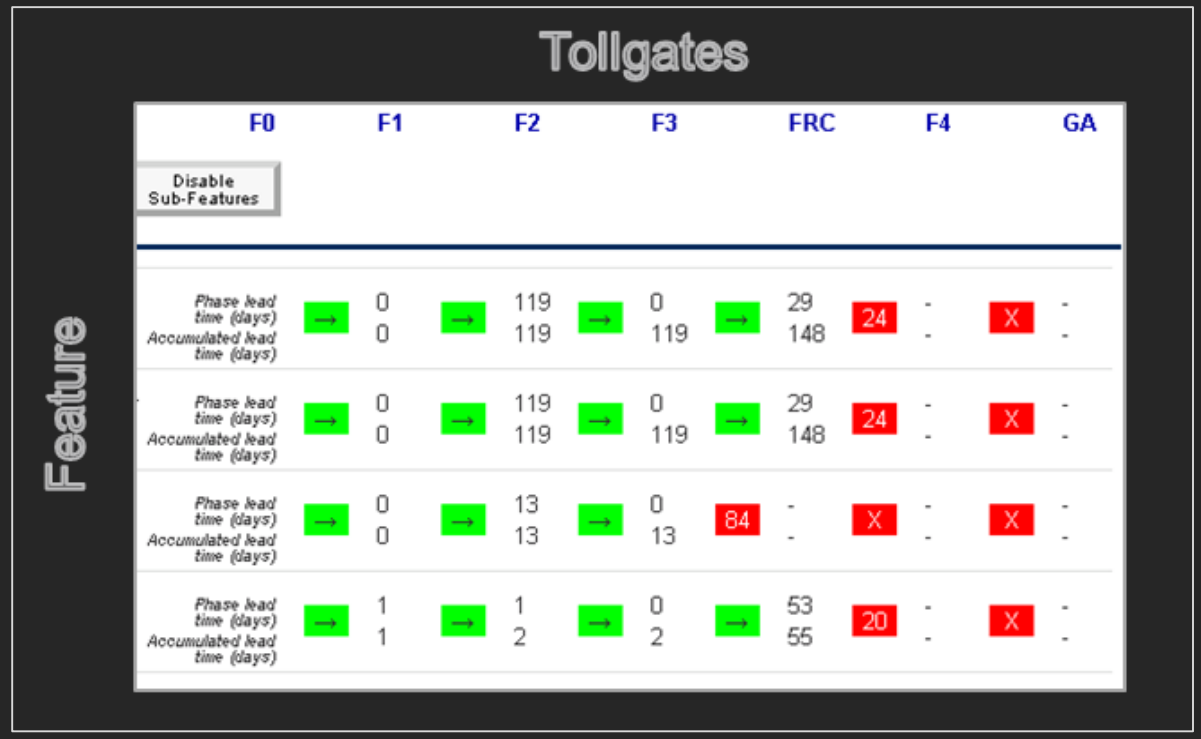
| Area | Measure | Measurable concept |
|---|---|---|
| Lean portfolio | HR Statistics | Employee satisfaction |
| Lean portfolio | Net promoter score | Customer satisfaction |
| Lean portfolio | Feature cycle time | Productivity |
| Lean portfolio | Team, program, large solution and portfolio self-assessment | Improvement |
| Lean portfolio | Release predicability | Improvement |
| Lean portfolio | Support call volume | Quality |
| Lean portfolio and Enterprise balance scorecard | Number of releases per year | Time to market |
| Lean portfolio and Enterprise balance scorecard | Number of defects | Quality |
| Enterprise balance scorecard | Team velocity vs. capacity | Efficiency |
| Enterprise balance scorecard | Teamwork | Agility |
| Enterprise balance scorecard | Value feature point delivered | Value delivery |

**Table 2.** Relevant team and large solution measures

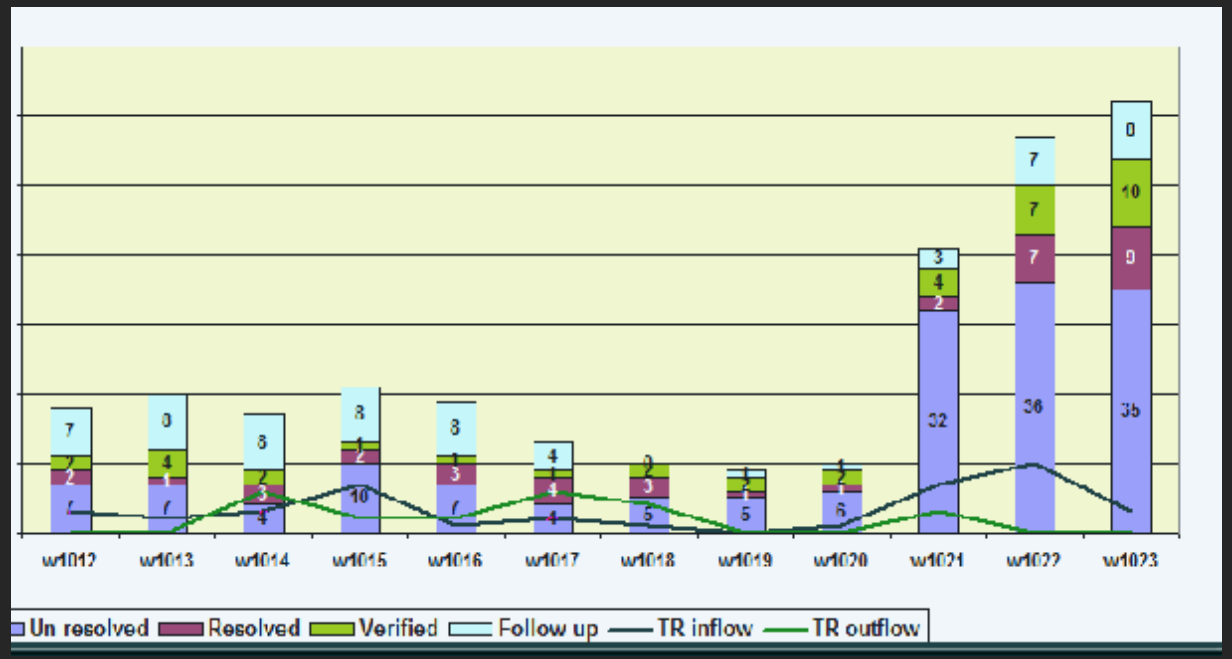
| Area | Measure | Measurable concept |
|---|---|---|
| STP | Program velocity | Functionality |
| STP | Predicability | Functionality |
| STP | Number of features planned | Functionality |
| STP | Number of features accepted | Functionality |
| STP | Number of enabler features planned | Functionality |
| STP | Number of enabler features accepted | Functionality |
| STP | Number of non-functional tests | Quality |
| STP and team | Number of stories planned | Functionality |
| STP and team | Number of stories accepted | Functionality |
| STP and team | Unit test coverage | Quality |
| STP and team | Number of defects | Quality |
| STP and team | Number of total tests | Quality |
| STP and team | Percent of automated test | Quality |
| Team | Perecent of stories accepted | Functionality |
| Team | Velocity planned | Functionality |
| Team | Velocity actual | Functionality |
| Team | Number of new test cases | Quality |
| Team | Number of new test cases automated | Quality |
| Team | Number of refactors | Quality |

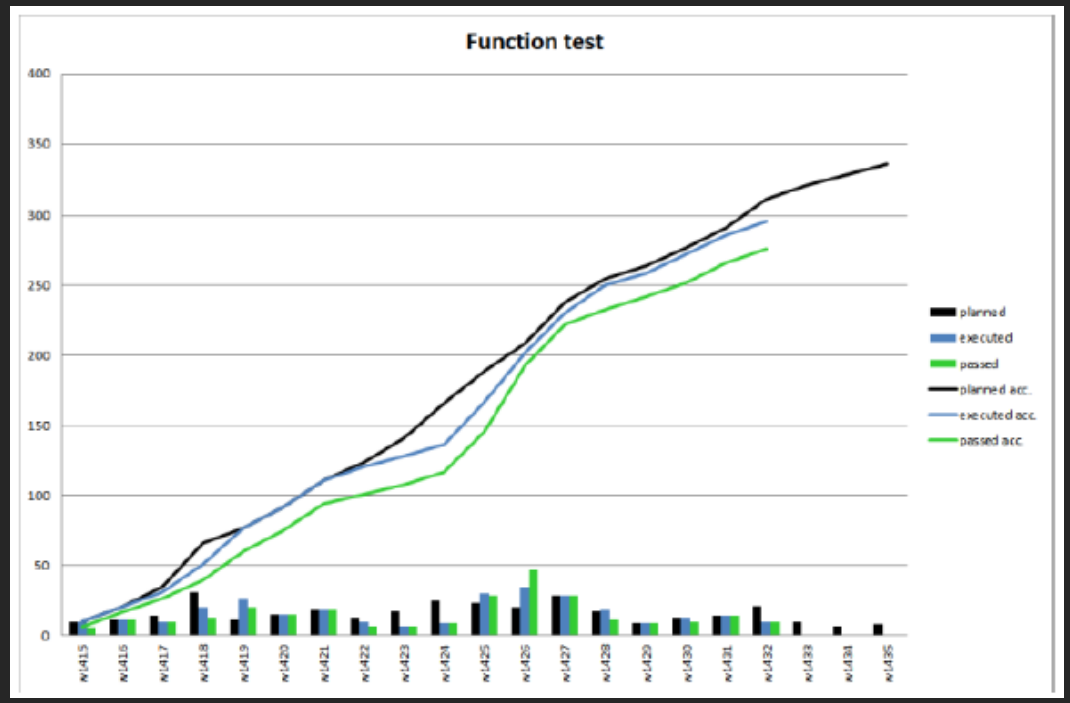
# Standard SAFe measures (examples)

- Program velocity, program predicability
- Number of features planned, Number of features accepted
- Number of enabler features planned, Number of enabler features accepted
- Number of non-functional tests
- Number of stories planned, Number of stories accepted
- Unit test coverage
- Number of defects
- Number of total tests
- Percent of automated test
- Velocity planned, Velocity actual
- Number of refactorings



# Theory: Monitoring Defect Backlog and test progress



# Results: Information needs for Scope creep

- Stakeholder: Product Manager
- Measures:
  - Problems from field tests: number of Problem Reports defects reports
  - Problems from the previous scope (from customer): number of Problem Reports
  - Old defects: number of PRs
  - Missed estimations: estimation + actual
  - Supporting others:  issues + burn-up + fluctuations in velocity
  - New features: JIRA
  - Changed external context

- Stakeholder: Development team
- Measures:
  - Changes in the environment (e.g. tooling)
  - New tasks
  - Problems from field tests: number of Problem Reports defects reports
  - Problems from the previous scope (from customer): number of Problem Reports
  - Old defects: number of PRs
  - Missed estimations: estimation + actual
  - Supporting others:  issues + burn-up + fluctuations in velocity
  - New features: JIRA
  - Dependency on other teams

# Results: relevant information needs for team and solution management

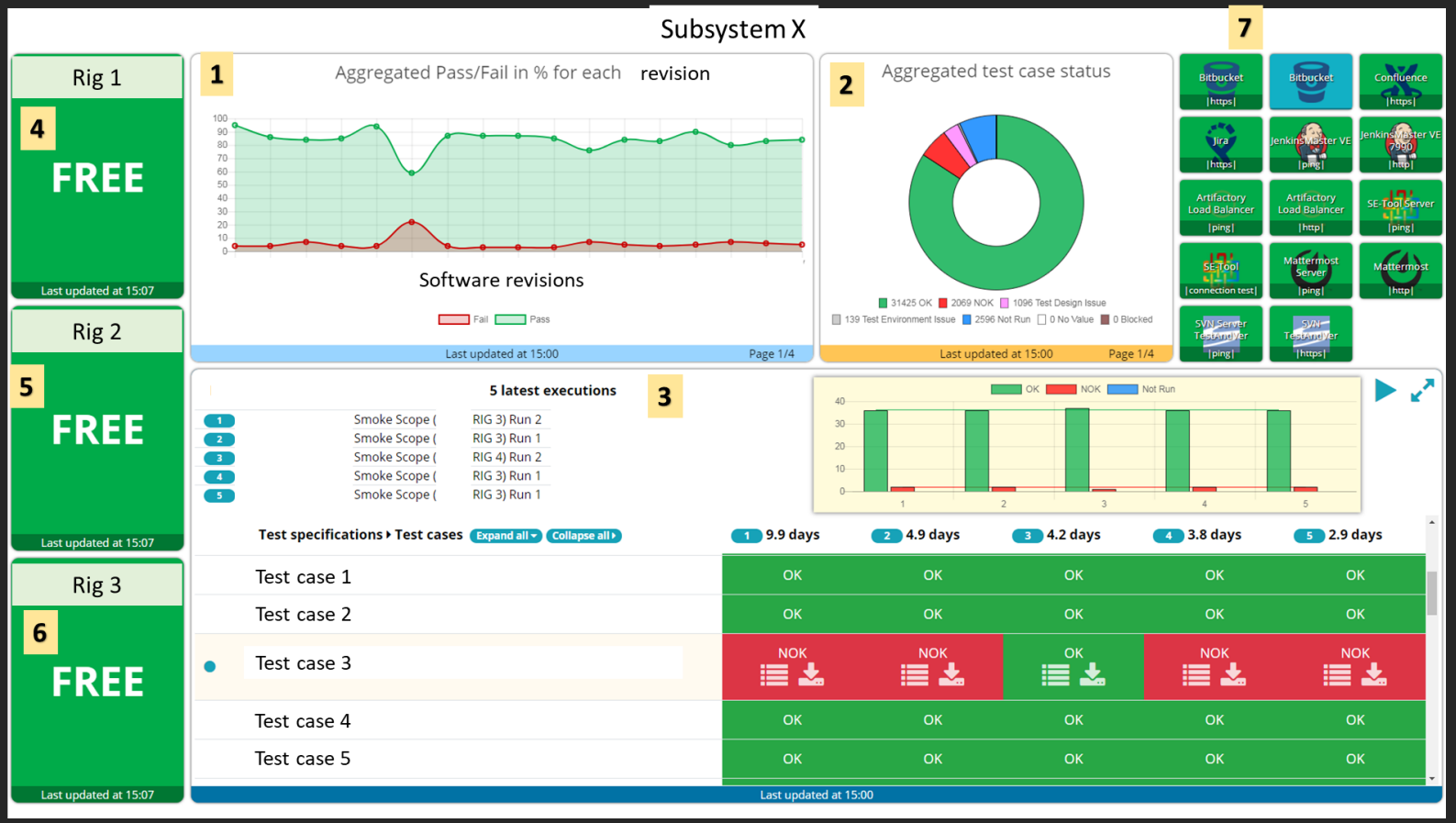



**Table 3.** Relevant team and large solution measures

| Stakeholder | Information Need | Measure(s) |
|---|---|---|
| ProdMan, DevTeam | What is the status of our field tests? | Number of Problem Reports (PR) from the field tests |
| ProdMan, DevTeam | What is our status of problems from the previous scope (from customer) | Number of PRs from customer |
| ProdMan, DevTeam | What is our status of legacy defects? | Number of internal PRs (previous release defects) |
| ProdMan, DevTeam | What is our planning accuracy? | Difference between estimated development time and actual development time |
| ProdMan, DevTeam | How much do we support other teams? | Number of resolved issues (internal defects) + burn-up + fluctuations in velocity |
| ProdMan, DevTeam | How much do we support other teams? | Number of new issues in internal defect reporting tool |
| DevTeam | How many changes in the development environment do we experience? | Number of changes introduced to tooling |
| DevTeam | What is the status of our quality journal? | Number of open problem reports |
| DevTeam | How dependent are we on other teams? | Expert assessment |

# Results: Problem reports from the field

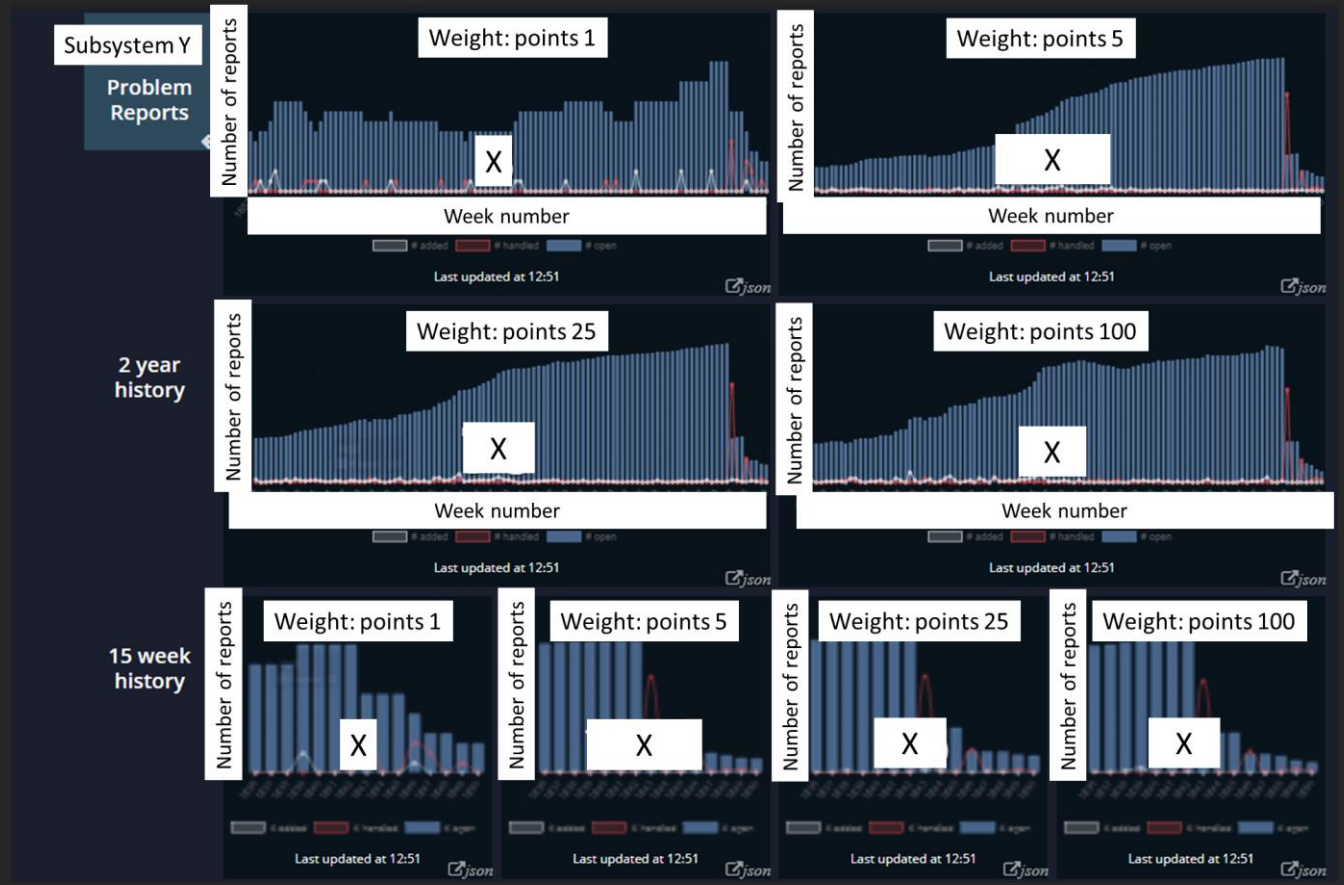
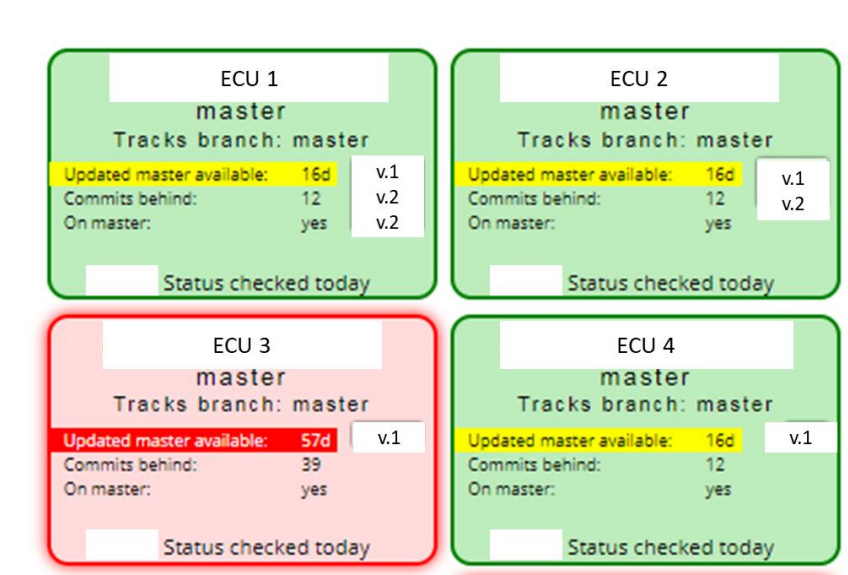
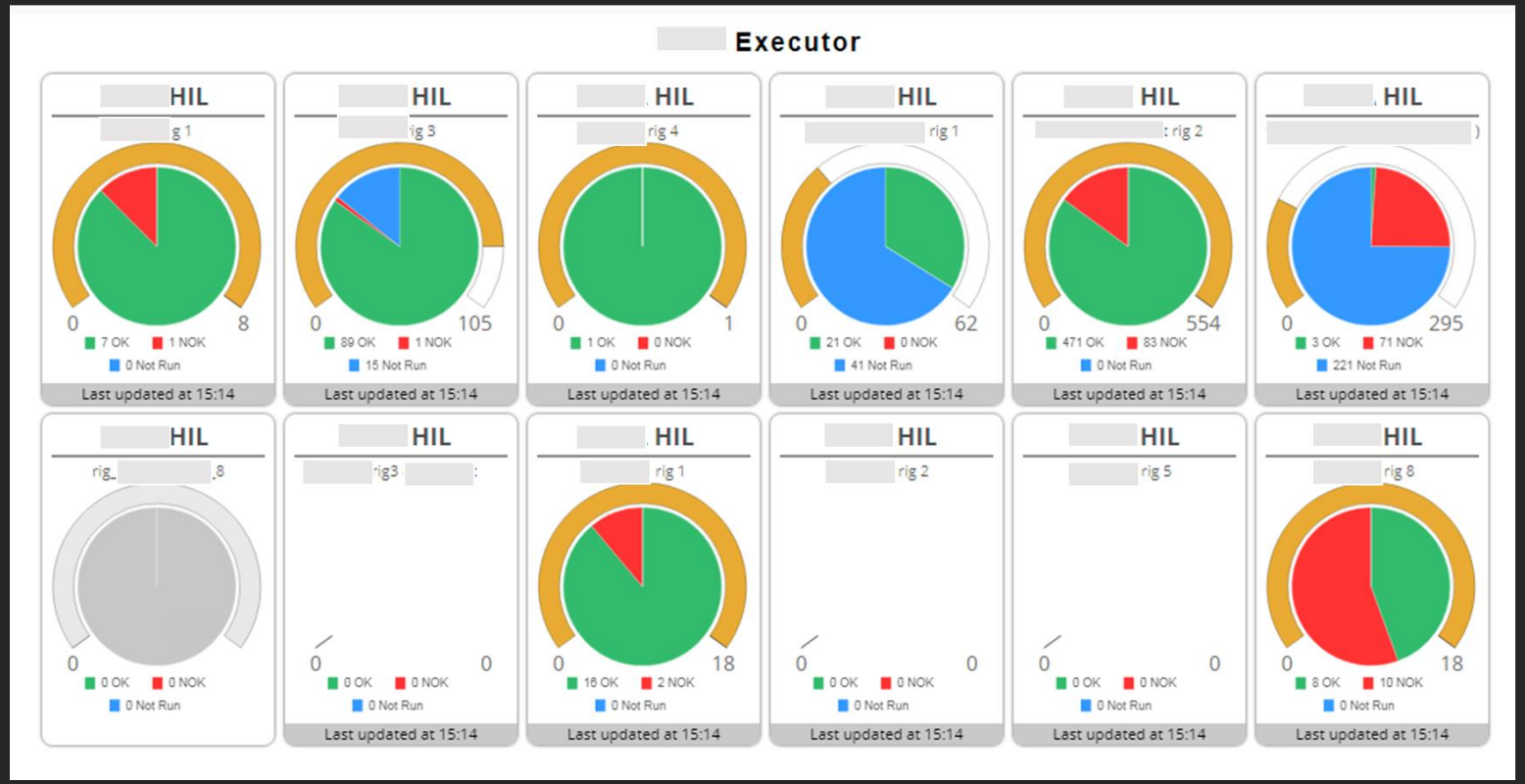



**Table 4.** Measures for defects into integration

| Stakeholder | Information Need | Measure(s) |
|---|---|---|
| Integrator | What is our integration status? | Days to integrate |
| Integrator | What are the versions of the used tools? | N/A |
| Integrator, DevTeam | What is our integration speed? | Build time |
| Integrator, DevTeam | What is our integration speed? | Compilation time |
| Integrator, DevTeam | What is our integration speed? | Generation time |
| Integrator, DevTeam | What is our integration speed? | Time to create the description file |
| Integrator | What is the size of our software? | Artifact size (binary file size) |
| Tester | What is the availability of our tools? | Binary status of tool availability |
| Tester | What is our test status? | Uptime / average time between fails |
| Tester | What is our test status? | Response time for server |
| Tester | What is our test status? | Test results over time (per sw. revision) |
| Tester | What is our test status? | Requirements coverage over time (per sw. revision) |
| Tester | What is our backlog? | Average number of sw. defects over time |
| Tester | What is the extra workload in our sprints? | Burn-up over time (work items not planned) |
| Tester | What is our planning accuracy? | Schedule slippage |
| Tester | What is our release speed? | Time between the designer's readiness of model and model's release |
| Tester | What is our test status? | Number of smoke tests executed |
| Tester | What is our test status? | Number of scope tests executed |
| DevTeam | What is the status of our test rigs? | Test rig availability |
| DevTeam | What is the quality of our integration? | Code commits/broken builds |
| DevTeam | What is the status of our release? | Status of release steps per sprint |
| DevTeam | What is our backlog? | Number of changed Electronic Control Units (to be integrated) |
| DevTeam | What is our integration speed? | Total lead time from model to integrated code |
| DevTeam | What is our test status? | Number of passed regression test cases |
| DevTeam | What is our test speed? | Execution time per test case |
| DevTeam | What is our defect resolution speed? | Defect resolution time |

# Results: tool set-up and CI progress



# Theory vs Companies' need (excerpt from our study)

| Measure | Theory | Company A | Company B |
|---|---|---|---|
| Velocity | ++ | -- | -- |
| Speed | -- | ++ | ++ |
| Number of releases per year | ++ | -- | -- |
| Release readiness | -- | ++ | -- |
| Team velocity vs. Capacity | ++ | -- | -- |
| Scope creep | -- | -- | ++ |
| Burn-up | -- | ++ | ++ |
| Number of *-tests | ++ | ++ | ++ |
| Number of defects | -- | ++ | ++ |
| Tool status (Up-time, ISP) | -- | ++ | ++ |
| Integration status (commits/broken builds) | -- | ++ | ++ |

# Conclusions & Further work

- Empowered software development teams' information needs are very different from the prescribed ones
  - Prescribed: based on management's needs
  - Actual: focus on product increments and team impediments
- New aspects
  - Speed
  - Availability of tools and infrastructure
  - Defects as feedback and improvement
- Next steps
  - Extended studies in more organizations
  - Longitudinal study on the evolution of the organization's information needs


# VISIT OUR WEBSITE
# www.software-center.se