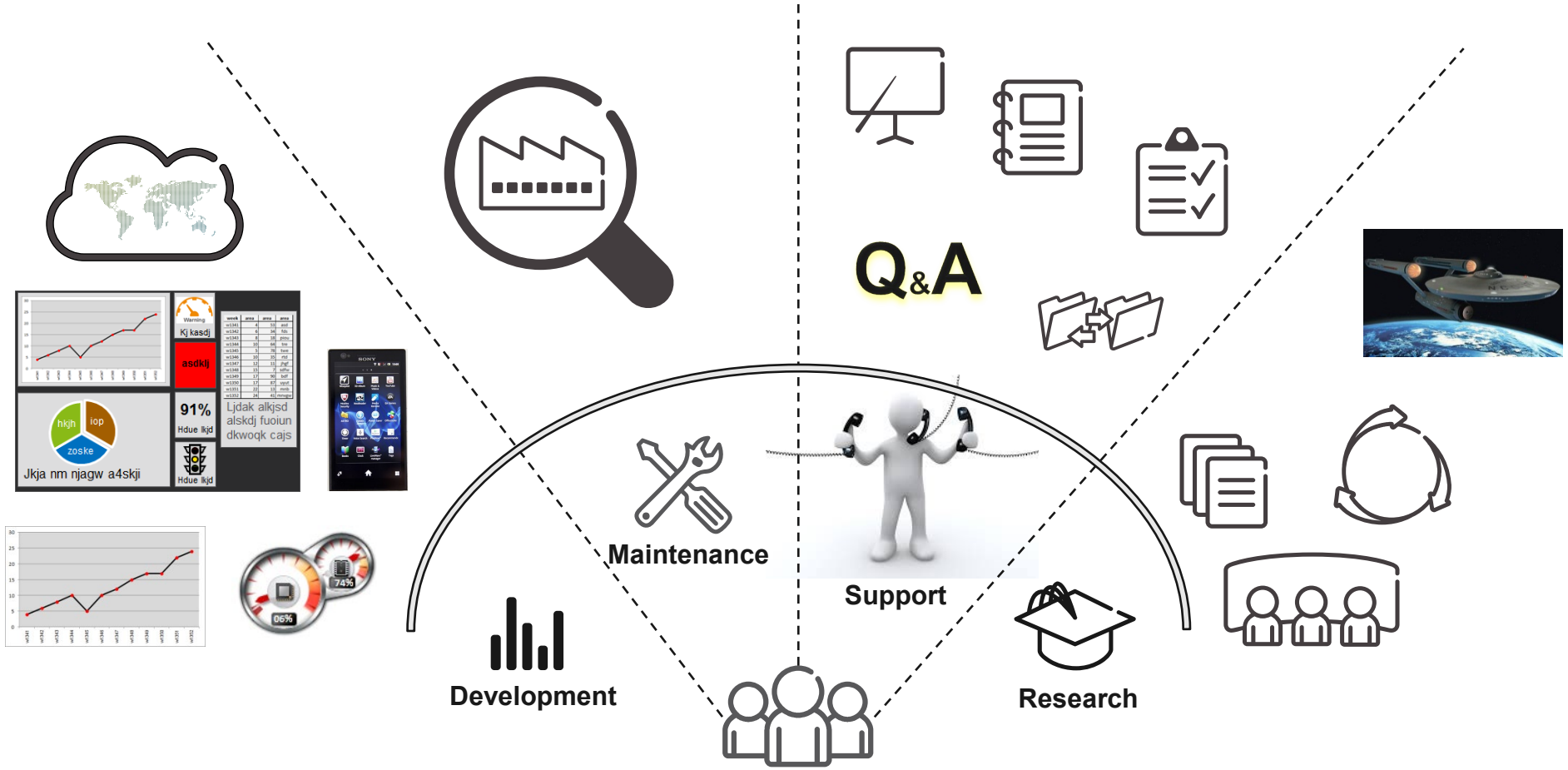
Experiences with metrics

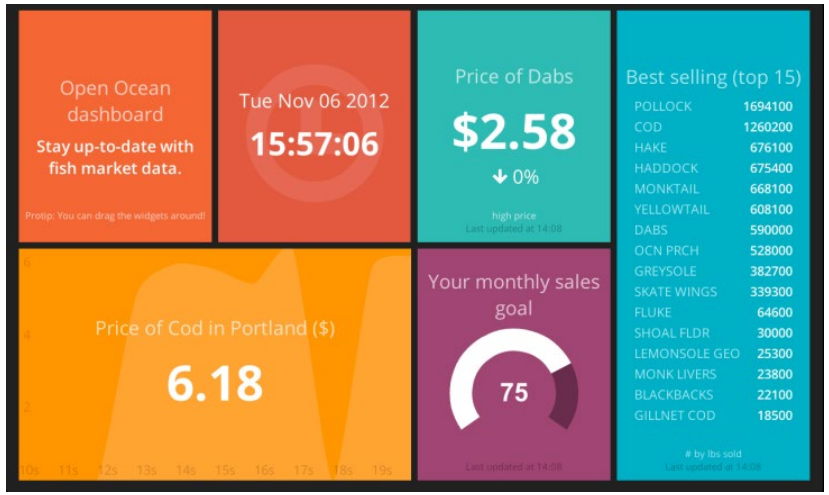
A webinar for the Linköping’s University,
by Wilhelm Meding, System Manager
Ericsson

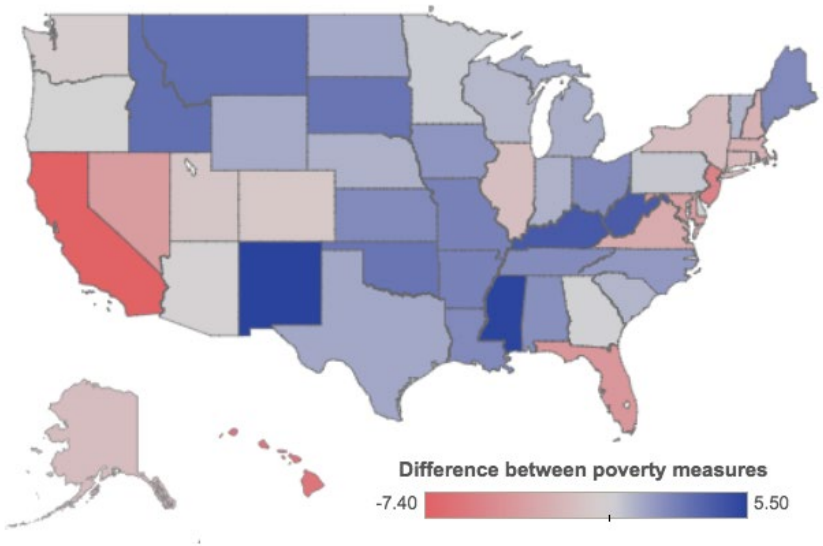
2021-04-06

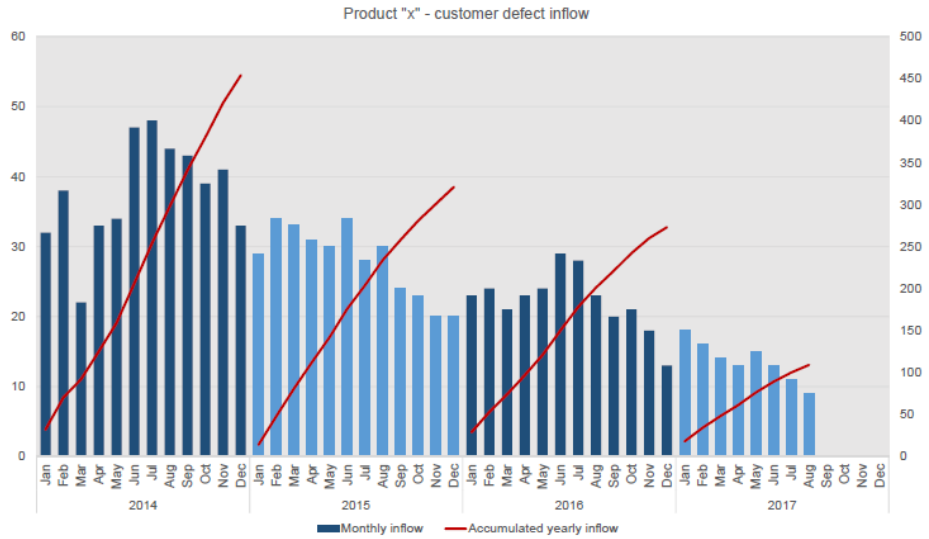
Metrics Team



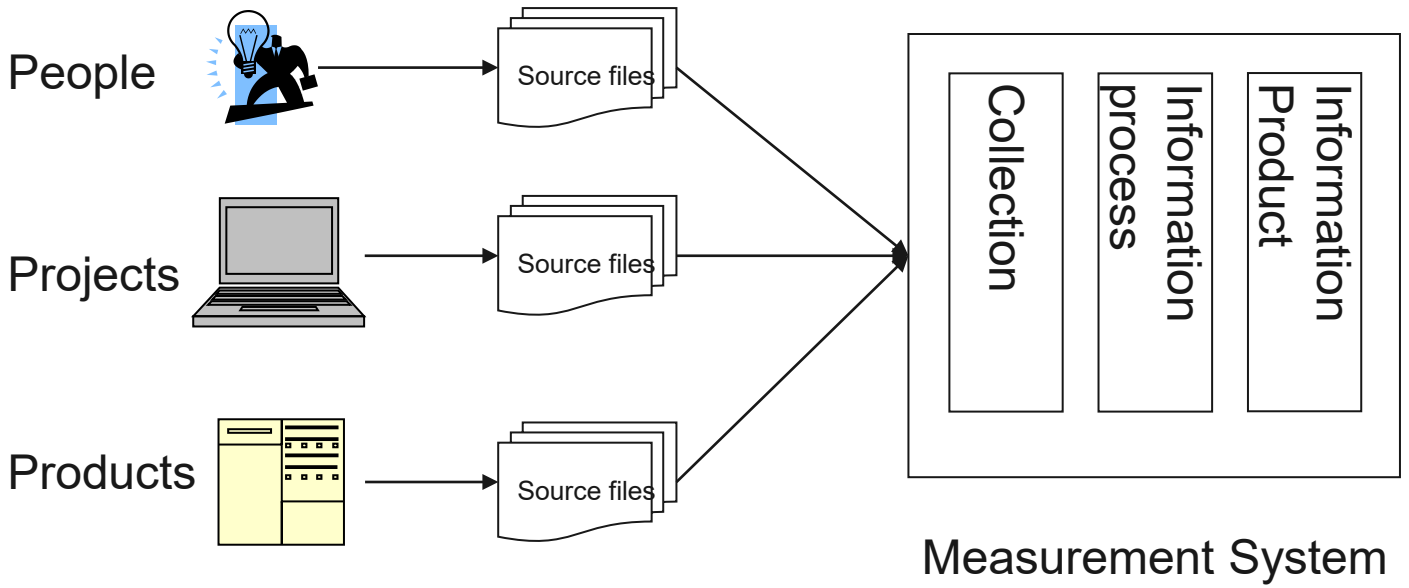
Measurement systems – examples

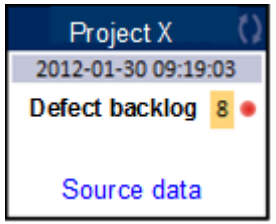






Measurement System - overview





Many *data points*

Few *data points*

Measurement Program

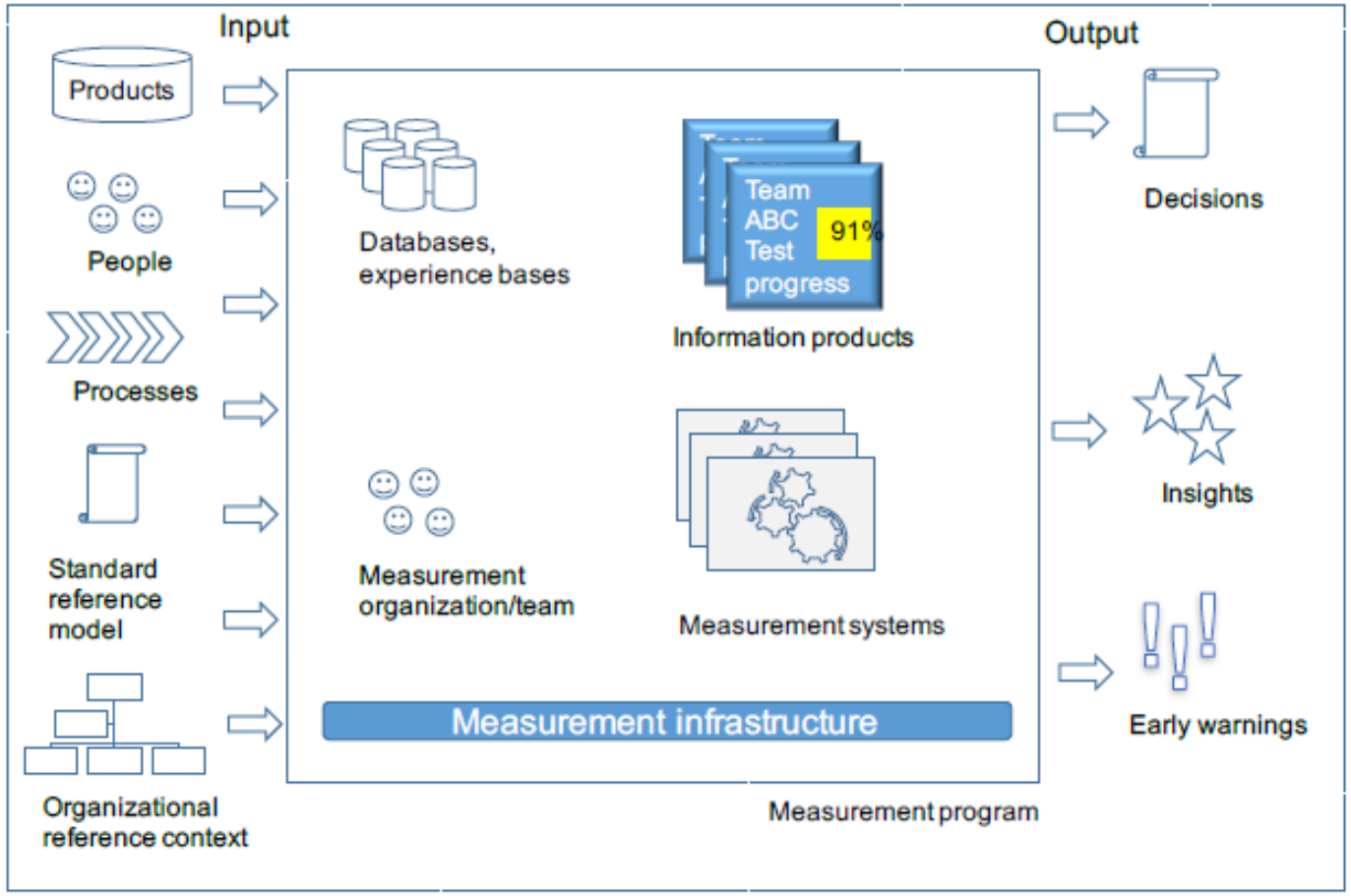


Fig. 1.4 Conceptual model of a measurement program.

Picture taken from book:
"Software Development Measurement Programs. Development, management and evolution"
Publisher: Springer
ISBN: 978-3-319-91835-8

Example of a server-based software measurement infrastructure

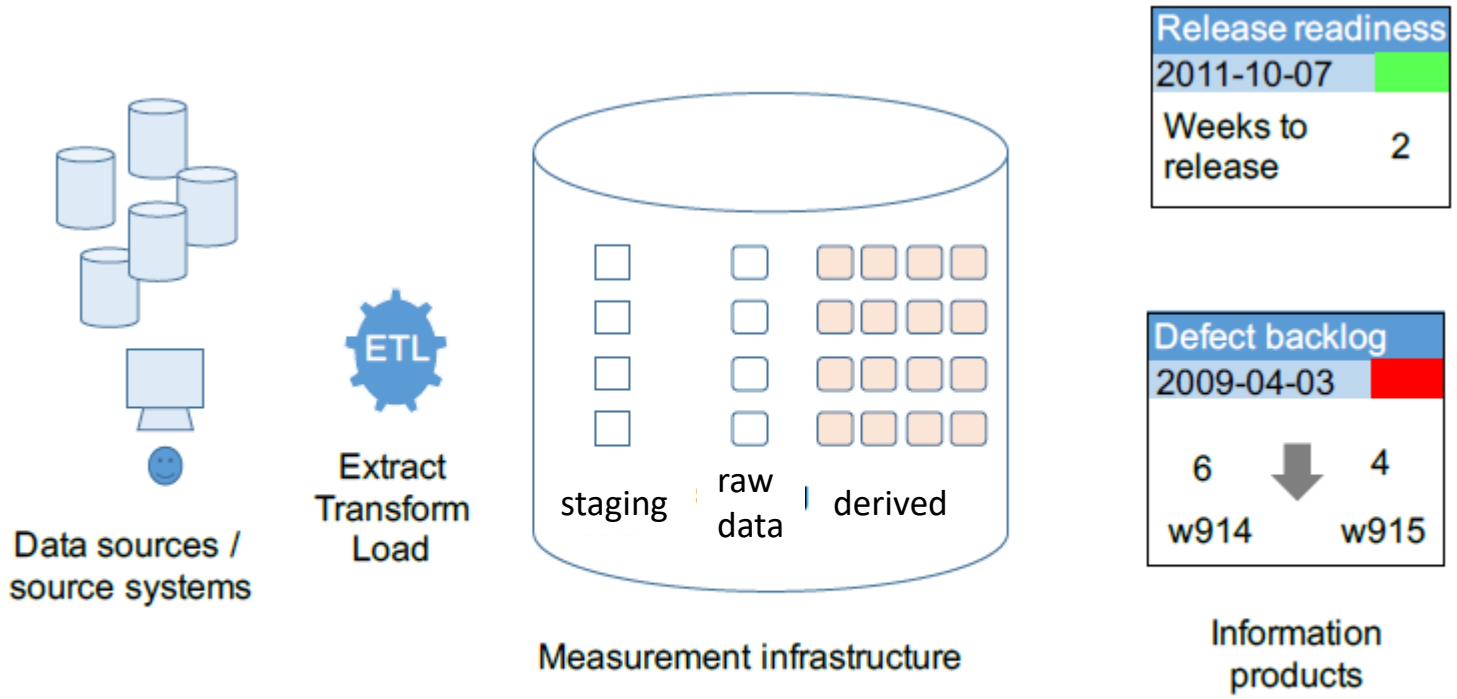


Fig. 3.5 An example of a server data storage solution.

Picture taken from book:
"Software Development Measurement Programs. Development, management and evolution"
Publisher: Springer
ISBN: 978-3-319-91835-8

Measurement Information Model*

The Measurement Information Model lists the main elements that build up a measurement system, and how they relate to one another.

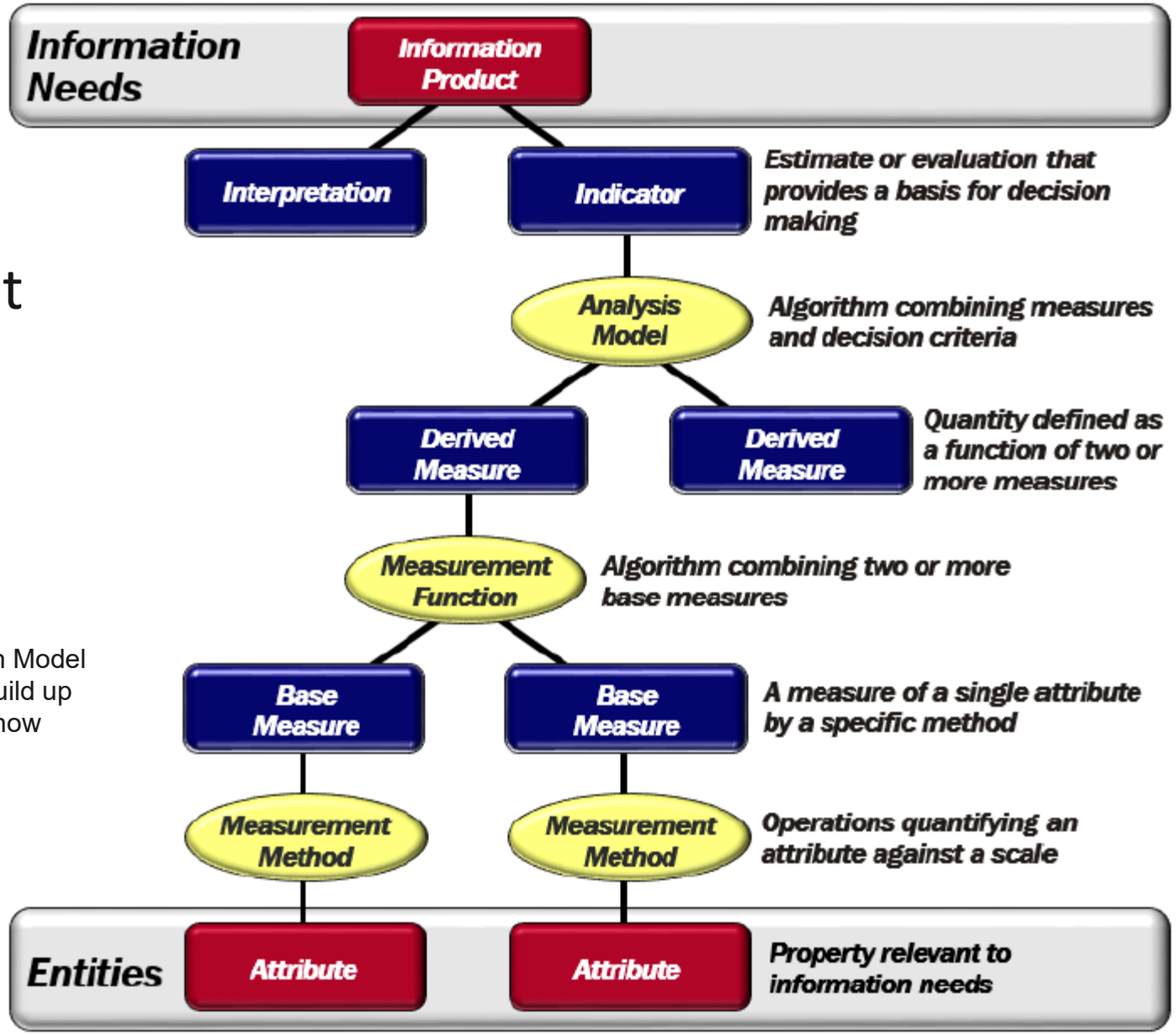
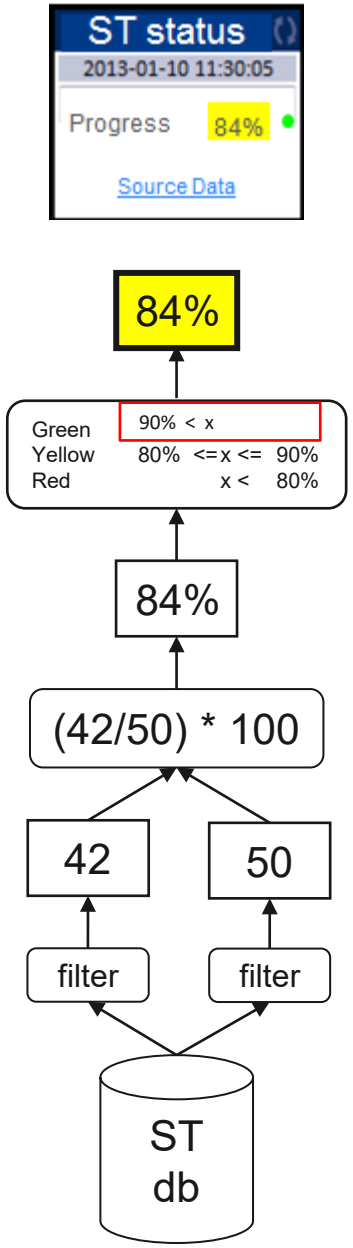


Figure A.1 — Key relationships in the measurement information model

Picture from
ISO/IEC/IEEE 15939:2017

An example



Information Product
A gadget that is automatically updated on a daily basis. Raw data and details can be accessed from the gadget.

Interpretation
- Red ask line for resources
- Yellow order overtime
- Green enjoy life

Indicator: Status of ST progress expressed in %

Analysis model
Result should be above 90% to conclude the progress is on schedule.

Derived Measure
System test progress

Measurement Function
Divide number of executed STC over planned STC, and multiply the result with 100.

Base Measures
- Number of executed System Test Cases
- Number of planned System Test Cases

Measurement Method
- Count number of executed System Test Cases
- Count number of planned System Test Cases

Attributes
- Executed System Test Cases
- Planned System Test Cases

Entity: System test case

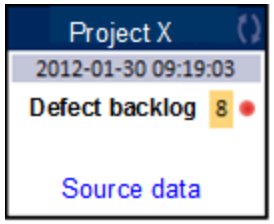
Measurable concept: System test progress

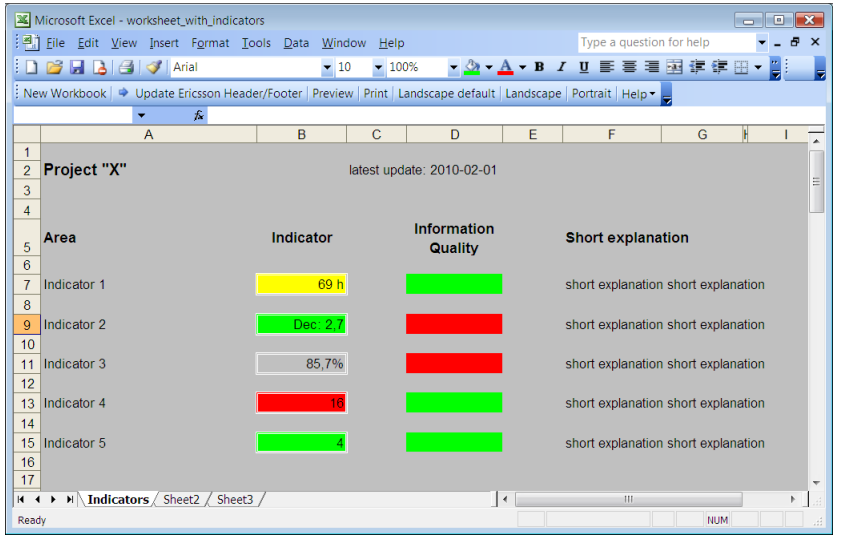
Information Need: What is the status of the progress of system testing?

Stakeholder: Leader of the System Test team

Trusting the numbers
How do we know that “green” is “green”?

- What can go wrong?
  - Data access (e.g. file/-s not accessible on the network)
  - Data update (e.g. database has not been updated)
  - Calculations (e.g. division by 0)
  - Decision criteria (e.g. value out of range)
  - Not possible to save (e.g. disk full)
  - …
- How do we know that nothing went wrong?
- How do we notify the stakeholder?
  - Either ”YES” or ”NO”

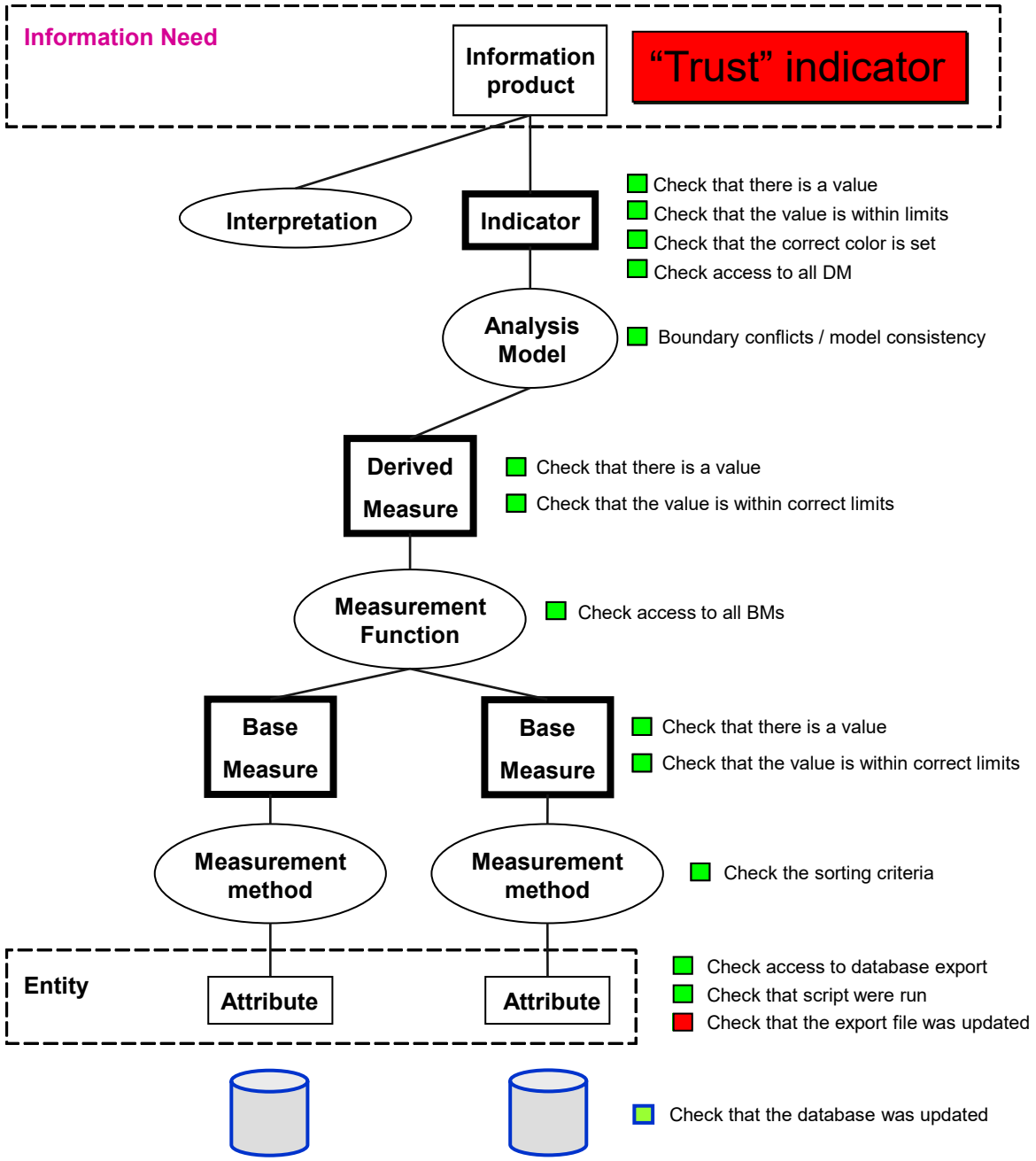


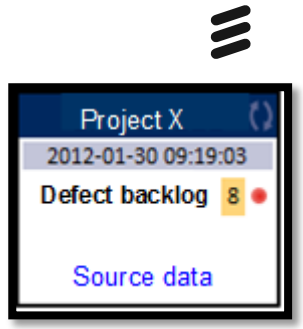


How do we know that green is green?

We check “states” and “transitions”

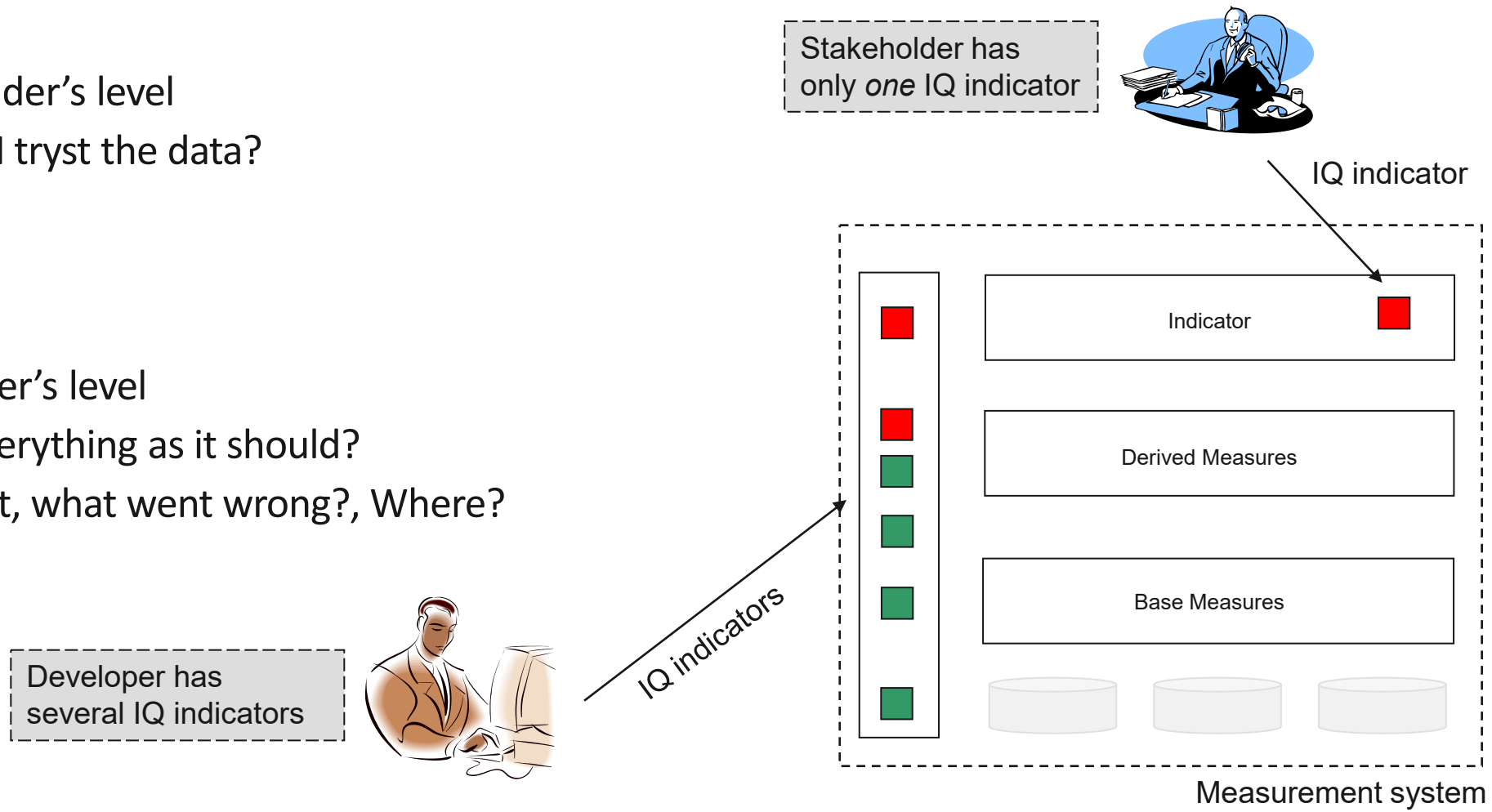
We summarize the “reliability” in one binary indicator





Two levels of Information Quality

- Stakeholder's level
  - Can I tryst the data?
- Developer's level
  - Is everything as it should?
  - If not, what went wrong?, Where?



Measuring Information Quality

External IQ
How the information is perceived by the stakeholder.

Examples:

- Believability
- Ease of operation
- Understandability

Assessment

Stakeholder

Information “consumption” – i.e. interpretation

External IQ
is non-measurable in automated way – can only be measured empirically (experiments, interviews, case studies, etc).

Indicator

Derived Measures

Base Measures

Internal IQ
How the information is obtained and composed from components.

Examples:

- Timeliness
- Free of error
- Completeness

Internal IQ
can be measured because it is related to how the information is “produced”.

Information “production” – i.e. providing information

Data sources

What we measure is:
How good is this arrow?

AIMQ* framework - Quality Attributes

- Free of error
  - the information is correct
- Understandability
  - the information is easy to comprehend
- Interpretability
  - it is easy to interpret what this information means
- Objectivity
  - the information was objectively collected
- Relevancy
  - the information is useful for our work
- Reputation
  - this information has a good reputation of quality
- Security
  - the information is protected from unauthorized access
- Timeliness
  - the information is sufficiently current for our work
- Accessibility
  - the information is easily retrievable
- Appropriate amount
  - the information is of sufficient volume for our needs
- Believability
  - the information is believable (non-doubtful credibility)
- Completeness
  - the information includes all necessary values
- Concise representation
  - the information is formatted compactly
- Consistent representation
  - the information is consistently presented in the same format
- Ease of operation
  - the information is easy to handle for our needs

* **AIMQ:** a methodology for information quality assessment
http://portal.acm.org/citation.cfm?id=637474.637478&preflayout=tabs

Example of a server-based software measurement infrastructure

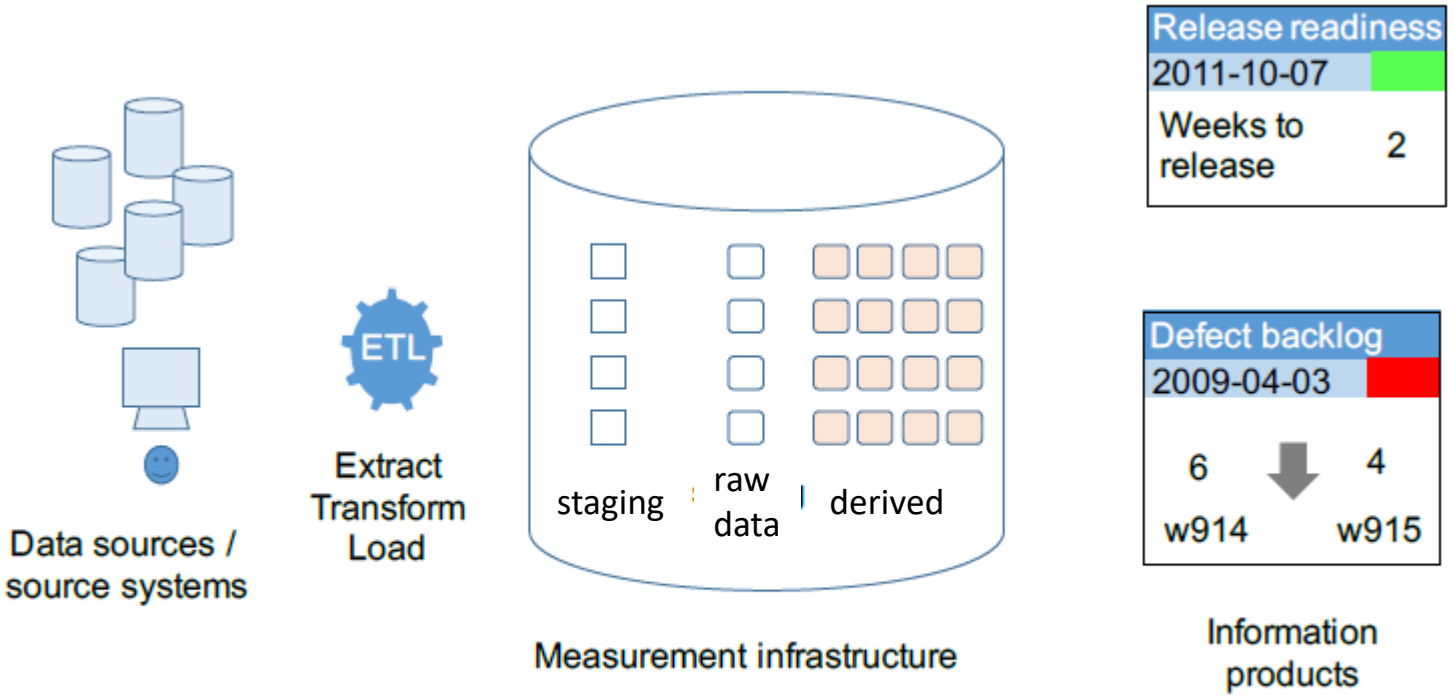


Fig. 3.5 An example of a server data storage solution.

Feedback to data source owners

Performance and data quality monitoring

Measures in Lean and Agile Software Developmnet Organizations

Development speed

Prediction of defect backlog (short term)

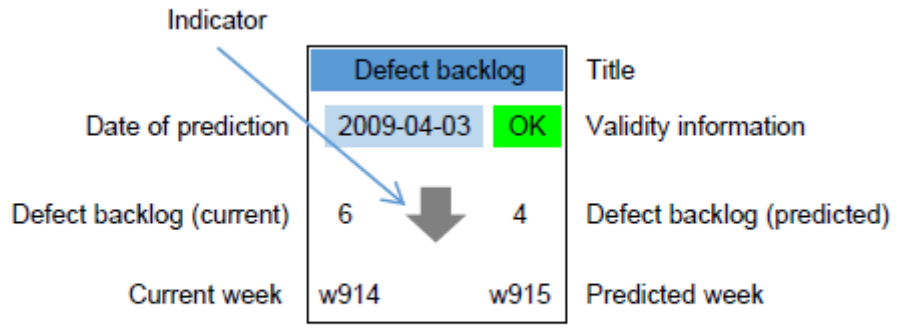


Fig. 6.3 MS Vista-like gadget, presenting the defect backlog indicator.

$$db(i) = db(i-1) + \frac{db(i-1) + db(i-2) + db(i-3)}{3} - \frac{do(i-1) + do(i-2) + do(i-3)}{3} \quad (6.2)$$

Picture taken from book:
"Software Development Measurement Programs. Development, management and evolution"
Publisher: Springer
ISBN: 978-3-319-91835-8

Release readiness

Fig. 6.1 MS Vista-like gadget, with predicted week of release.

$$\text{Release readiness} = \frac{\text{No of defects}}{\text{Defect removal rate} - (\text{Test execution rate} - \text{Test pass rate})}$$

Picture taken from book:
"Software Development Measurement Programs. Development, management and evolution"
Publisher: Springer
ISBN: 978-3-319-91835-8

Progress of software development teams

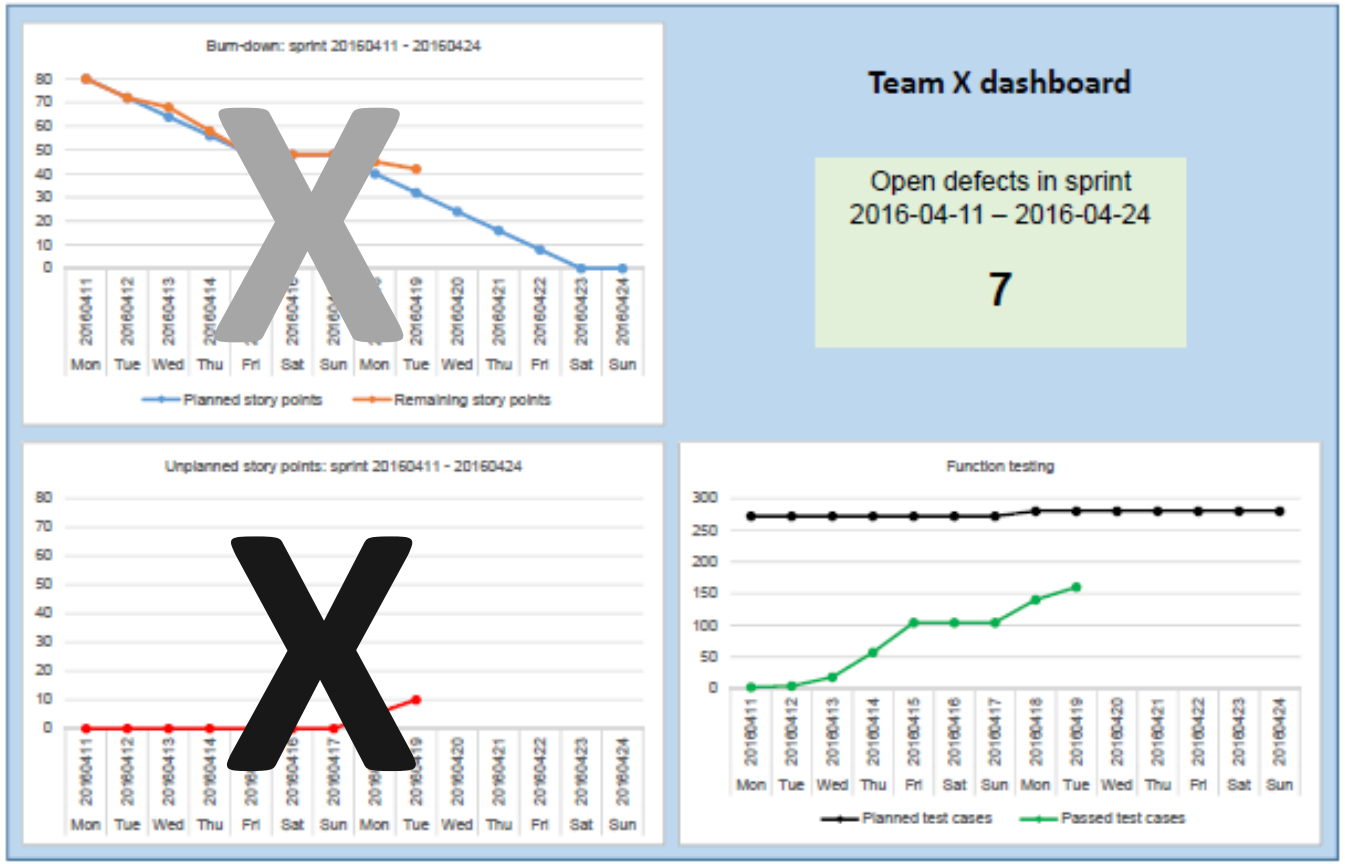
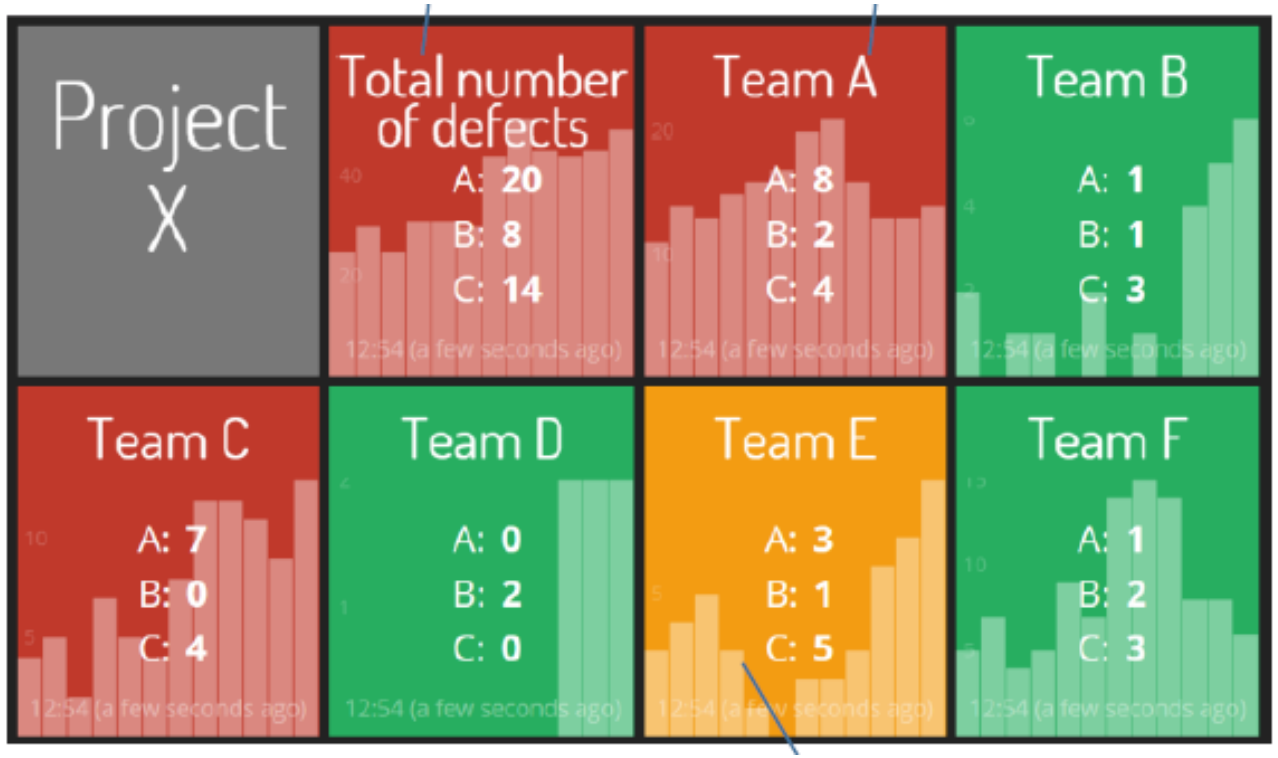


Fig. 6.11 Dashboard for a development team, with related measures.

Picture taken from book:
"Software Development Measurement Programs.
Development, management and evolution"
Publisher: Springer
ISBN: 978-3-319-91835-8

Software defect related measures



- Systems/Sub-systems
- Test results
- Time (speed)
- Availability of branches
- Performance
- …

Customer defect inflow

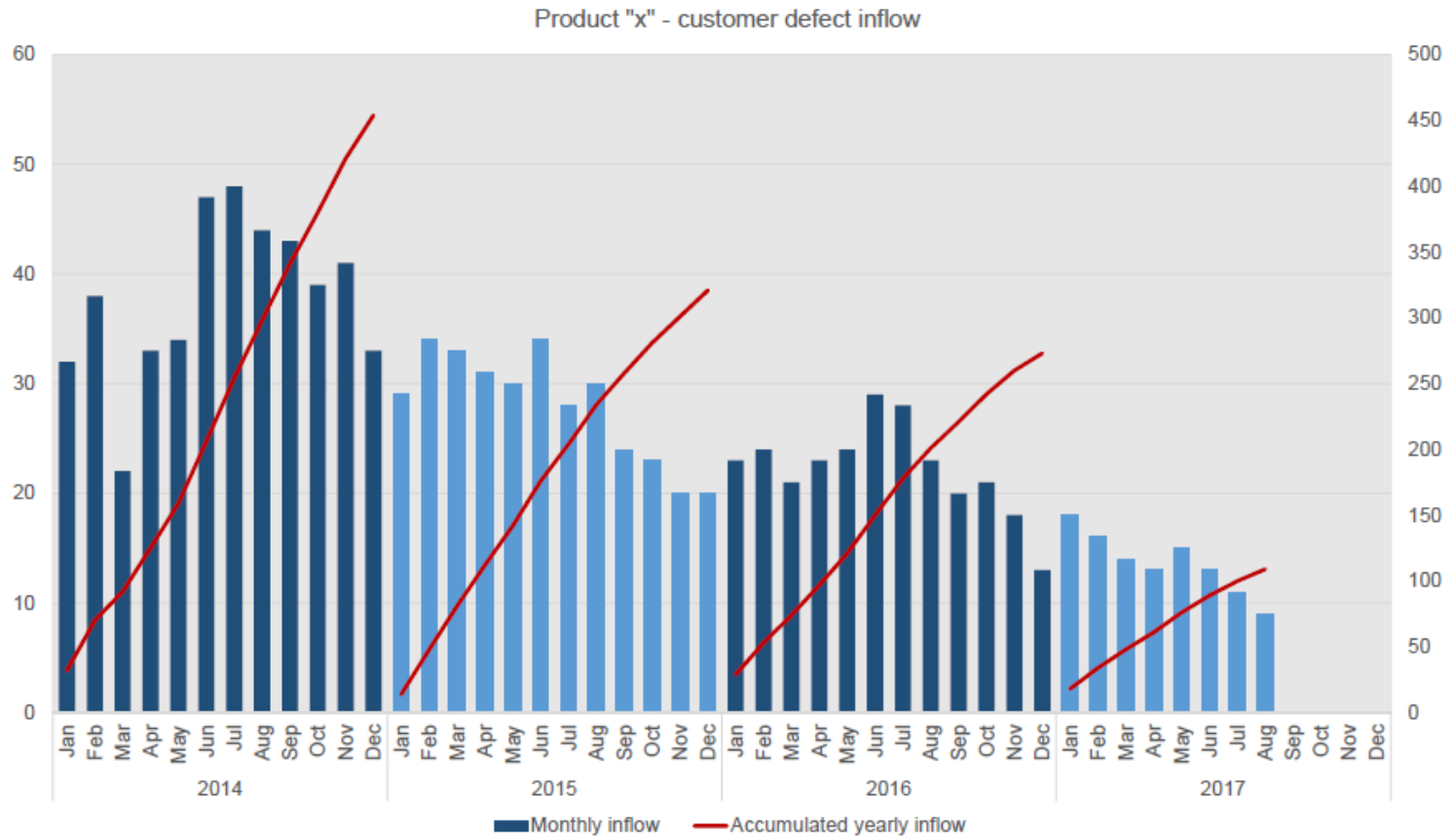


Fig. 6.12 Customer defect inflow, of the fictitious product x.

Picture taken from book:
"Software Development Measurement Programs.
Development, management and evolution"
Publisher: Springer
ISBN: 978-3-319-91835-8

Sum-up

- Measurement system
  - Mechanism that collects data, process data and presents a result
- Information Quality
  - Addresses the question: ”Can I trust what I see?”
- ISO/IEC/IEEE 15939:2017
  - Systems and software engineering – Measurement process