Software Engineering Reviews

TDDC90

autumn 2024

Kristian Sandahl
Department of Computer and Information Science
Linköping University, Sweden
kristian.sandahl@ida.liu.se

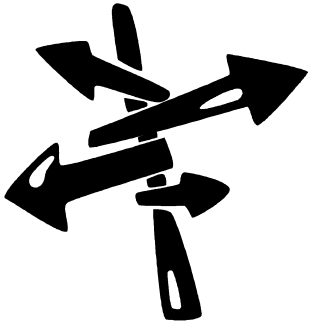
Linköpings universitet

Agenda - Theory

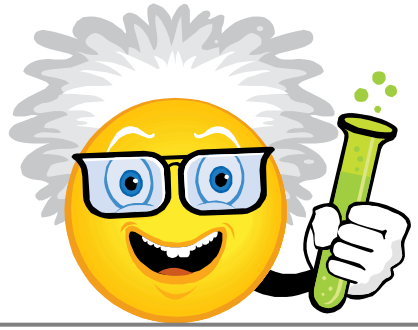
kristian.sandahl
@liu.se

Part I
Inspections

Part II
Other reviews



Part II
Variants and research



kristian.sandahl
@liu.se

Part I
Inspections

Systematic inspections

kristian.sandahl
@liu.se

The best way of finding many defects in code and other documents

- Experimentally grounded in replicated studies

Goals:

- Find defects (anomalies)
- Training
- Communications
- Hostage taking

Development over the years

kristian.sandahl
@liu.se

- Fagan publishes results from code and design inspections 1976 in IBM systems journal
- Basili and Selby show the advantage of inspections compared to testing in a tech-report 1985.
- Graham and Gilb publish the book Software inspections 1993. This describes the standard process of today.
- Presentation of the Porter-Votta experiment in Sorrento 1994 starts a boom for replications.
- Sauer et al compare experimental data with behavioural research in a tech-report 1996
- IEEE std 1028 updated 2008

Roles

kristian.sandahl
@liu.se

- Author
- Moderator (aka Inspection leader)
- Reader (if not handled by the Moderator)
- Inspector
- Scribe (aka Recorder)

kristian.sandahl
@liu.se

Process

- Initial:
  - Check criteria
  - Plan
  - Overview
- Individual:
  - Preparation, or
  - Detection
- Group:
  - Detection, or
  - Collection
  - Inspection record
  - Data collection
- Exit:
  - Change
  - Follow-up
  - Document & data handling

Inspection record

kristian.sandahl
@liu.se

- Identification
- Location
- Description
- Decision for entire document:
  - Pass with changes
  - Reinspect

Data collection

kristian.sandahl
@liu.se

- Number of defects
- Classes of defects
- Severity
- Number of inspectors
- Number of hours individually and in meeting
- Defects per inspector
- Defect detection ratio:
  - Time
  - Total defects

Our inspection record

kristian.sandahl
@liu.se

Id	Loc.	Description	Class.
1	6,7	Should be maximum number	minor
2	6, 14, 15	Input output inconsistent	major
3	10	Should be next series	minor
4	16	Don’t use right to	minor
5	17	Superflous loop	major
6	20, 21	Wrong parentheses	minor
7	20,21	Should be largest max	minor
8	20,21, 26	Inconsistent example	minor

kristian.sandahl
@liu.se

Practical investigation

- 214 code inspections from 4 projects at Ericcson
- Median number of defects = 8
- 90 percentile = 30
- Majority values:
  - up to 3.5 h preparation per document
  - up to 3 h inspection time
  - up to 4000 lines of code
  - 2 to 6 people involved

Inspection rate (IEEE Std 1028-2008)

Requirements or Architecture (2-3 pages per hour)

Source code (100-200 lines per hour)

Regression wrt defect detection ratio

kristian.sandahl
@liu.se

- Preparation time per code line typically 0.005 hours per line (12 minutes per page)
- Size of document have negative effect on DFR, max recommendation 5000 lines
- A certain project is better than two of the others
- 4 inspectors seems best (not significant)
- *Analysis performed by Henrik Berg, LiTH-MAT-Ex-1999-08*

kristian.sandahl
@liu.se

Part II
Other reviews

Other reviews

kristian.sandahl
@liu.se

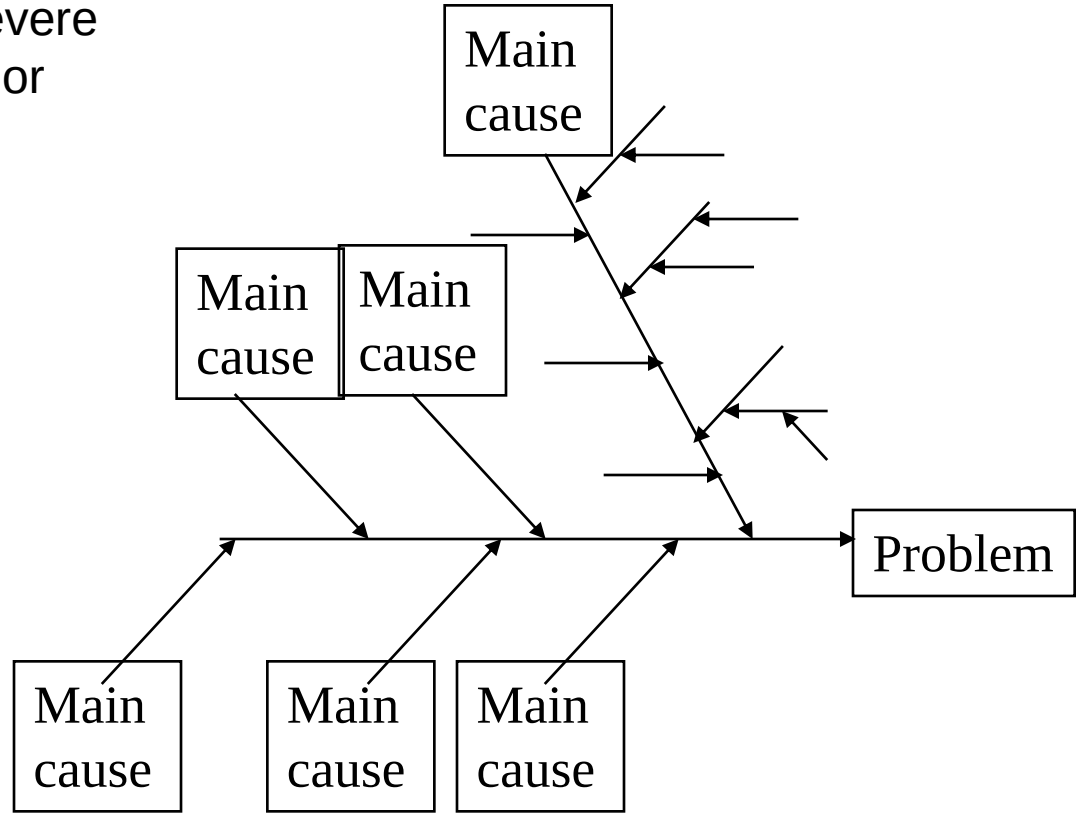
- Management review – check progress
- Technical review – evaluate conformance
- Walk-through – improve product, training
- Audit – 3rd party, independent evaluation

- (Peer) Review
- Buddy-check
- Desk check

kristian.sandahl
@liu.se

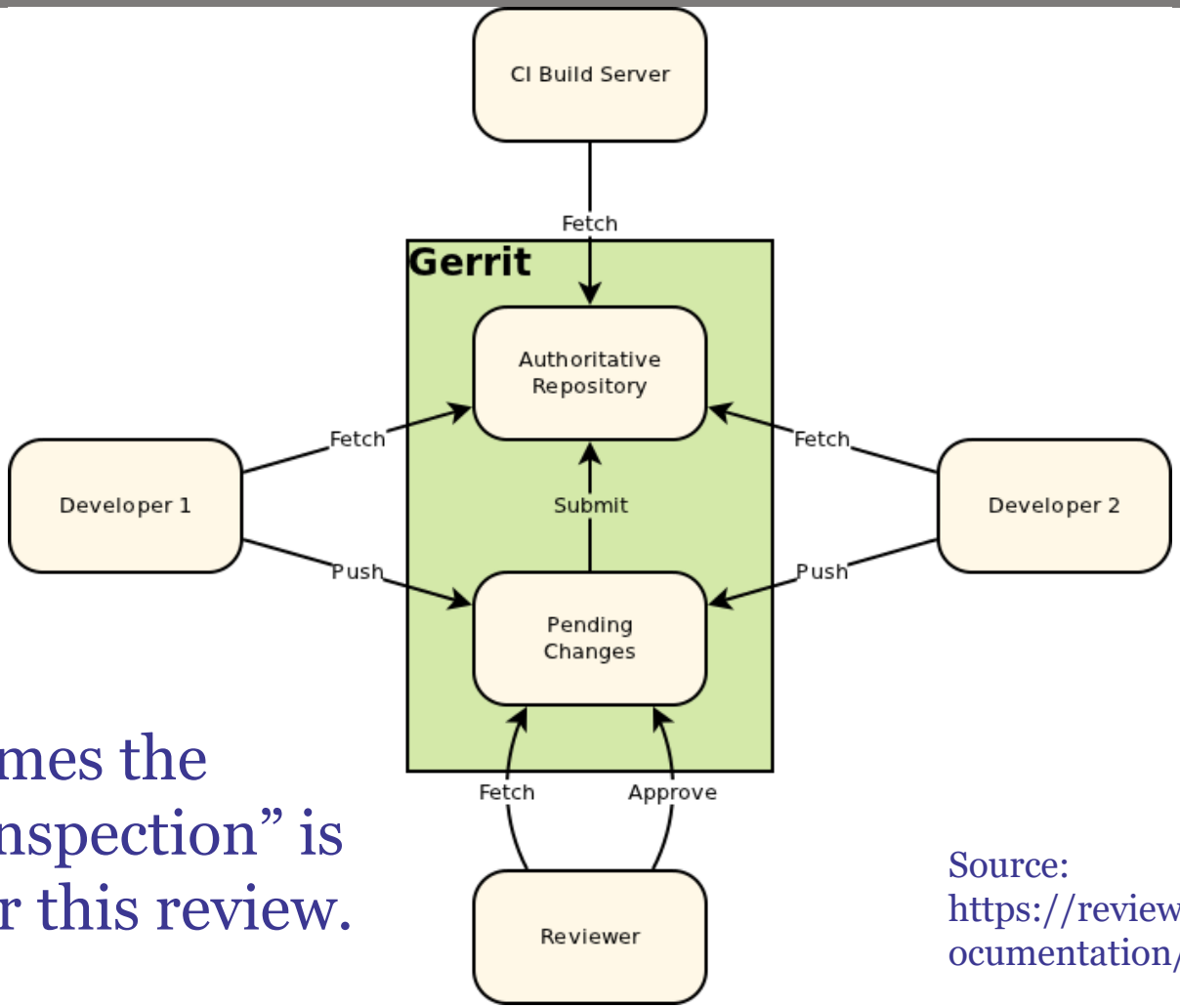
Root-cause analysis

- Performed regularly for severe defects, frequent defects, or random defects
- Popular mind map:
  The Ishikawa diagram
- Parameters:
  - Defect category
  - Visible consequences
  - Did-detect
  - Introduced
  - Should-detect
  - Reason



kristian.sandahl
@liu.se

Tool-based code review in Gerrit



Sometimes the term ”inspection” is used for this review.

Source: https://review.openstack.org/Documentation/intro-quick.html

kristian.sandahl
@liu.se

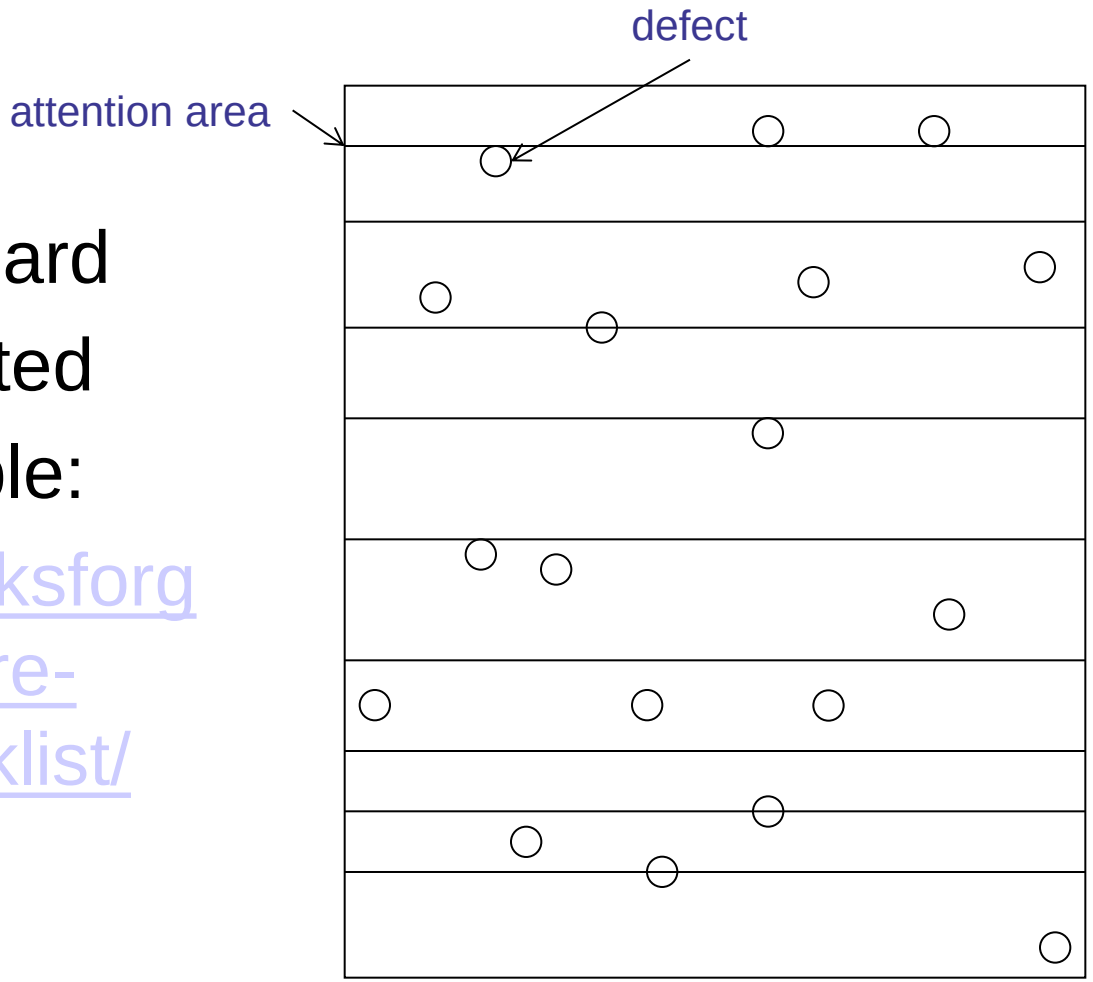
Part II
Variants and research

kristian.sandahl
@liu.se

Reading techniques - checklist

- Checklist
- Industry standard
- Shall be updated
- Simple example:

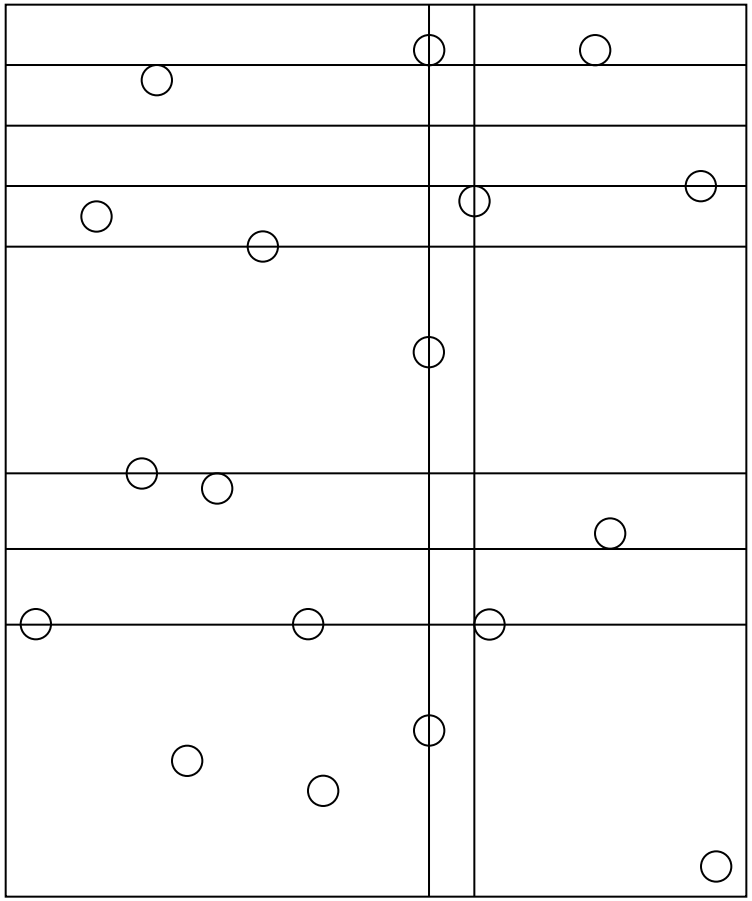
https://www.geeksforgeeks.org/software-inspection-checklist/



kristian.sandahl
@liu.se

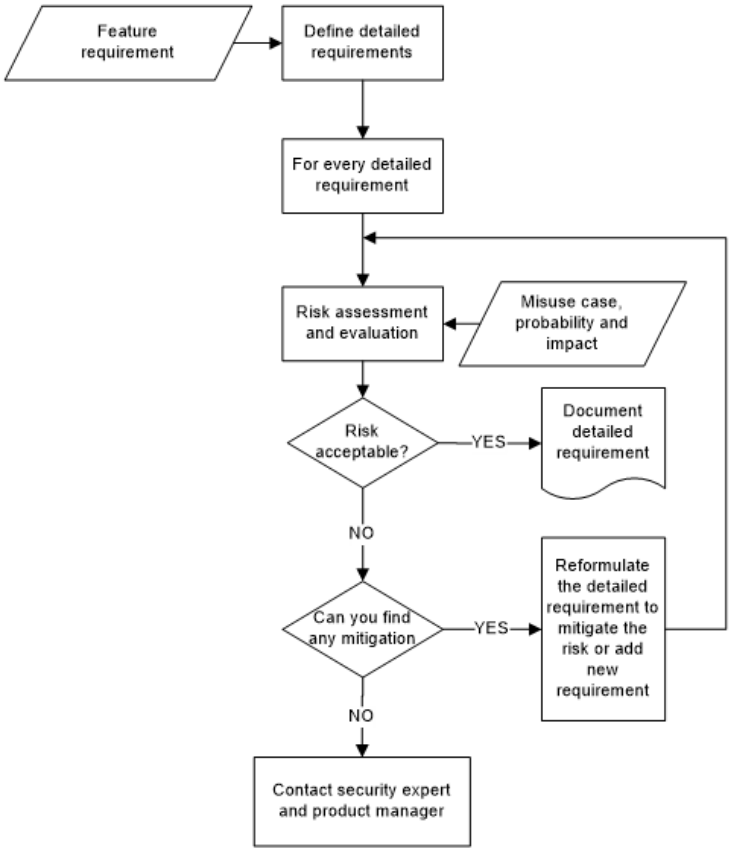
Reading techniques - scenario

- Scenario
- A checklist splitted to different responsibilities
- 30% higher DFR ?



The SRA approach scenario example

- A light-weight security risk assessment method (SRA) to be applied by non-security experts in requirements engineering
  - For every function-level/detailed requirement, perform a risk assessment by answering following questions:
    - What is the asset? What shall be protected?
    - Who has access to asset and how?
    - Can the actor/user, identified above, misuse the asset?
    - What is the probability over certain period and what is the impact of harm?



SRA example

Context: Automated operation and maintenance of handover functions when neighbor nodes provide services jointly.

R2: The node shall collect and log Automatic Neighbor Relationship (ANR) measurement results from the User Equipment (UE) selected for reporting.

SRA example

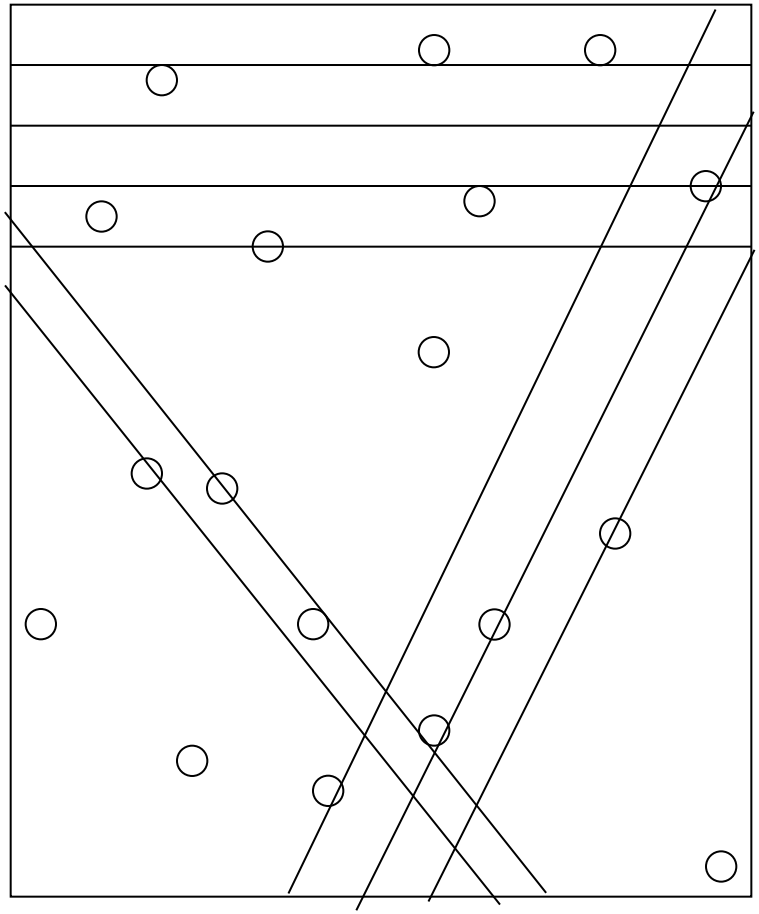
R2: The node shall collect and log Automatic Neighbor Relationship (ANR) measurement results from the User Equipment (UE) selected for reporting.

Asset	Access	Misuse	Probability/ Impact	Risk level
ANR measurement data	End-user of UE	Malicious actor can modify measurement reports	Possible/Serious	Medium

Reading techniques – perspective-based

kristian.sandahl
@liu.se

- Different inspectors repre-sent different roles
- Real or played roles
- 30% higher DFR ?



Cost of quality

kristian.sandahl
@liu.se

- Person-hours
- Calender time
- Good reading techniques
- Good data recording

MS Copilot

1. **Inconsistent Units**:
   - The thickness of the vertical bars is specified as “2,325 mm”. This seems unusually precise and large for a plotting component. It might be a typo or require clarification.
2. **Ambiguous Instructions**:
   - The instruction to “Print a copy of the header at the bottom of the last series” could be clearer. It’s not specified if this should be done for each plot or just once at the end of all plots.
3. **Range Calculation**:
   - The formula for determining the value range, “If 10n-1 < max((abs(smallest min); abs(smallest max)) < 10n”, is complex and might be prone to misinterpretation. It could benefit from a more detailed explanation or an example.
4. **Plotting Symbols**:
   - The use of “>” and “<” to mark the median might be confusing. Typically, a single symbol or a different notation is used to avoid ambiguity.
5. **Header Example**:
   - The example header provided (“Measurement equipment, Analysis module, Plotting module, Feeder, Reports, Monitor and Control”) seems more like a list of components rather than a header for a plot. This might need clarification or correction.
6. **Series Limit**:
   - The document states “max 15 series” but does not specify what happens if there are more than 15 series. Should the excess be ignored, or should a new plot be created?

ChatGPT

1. Thickness Precision:
- •The specification states "vertical bars of thickness 2,325 mm," which
- •likely includes a misplaced comma (European notation for decimal points) or i
- •s overly precise for typical plotting needs. Verify if this should be "2.325 mm"
- •or simply "2 mm."

2.Median Marking:
- •The description for marking the median suggests placing > and < to the left
- •and right of the bar, respectively. This might cause ambiguity in visualization
- •since these symbols could overlap or make interpretation challenging.

3.Header Redundancy:
- •The header is printed at the top and at the bottom after the last series.
- •Repeating the header might clutter the plot or consume unnecessary space.

4.Value Range Calculation (Sub-Process):
- •The rule for determining the value range header involves the condition
- •10n-1 < max((abs(smallest min); abs(smallest max)) < 10n. This is complex
- •and could benefit from clarification or a concrete example to avoid
- •misinterpretation.

5.Series Limit:
- •The specification allows for a maximum of 15 series per plot. If the data input
- •exceeds this, how to handle additional series is unclear (e.g., create a new
- •plot or truncate the data).

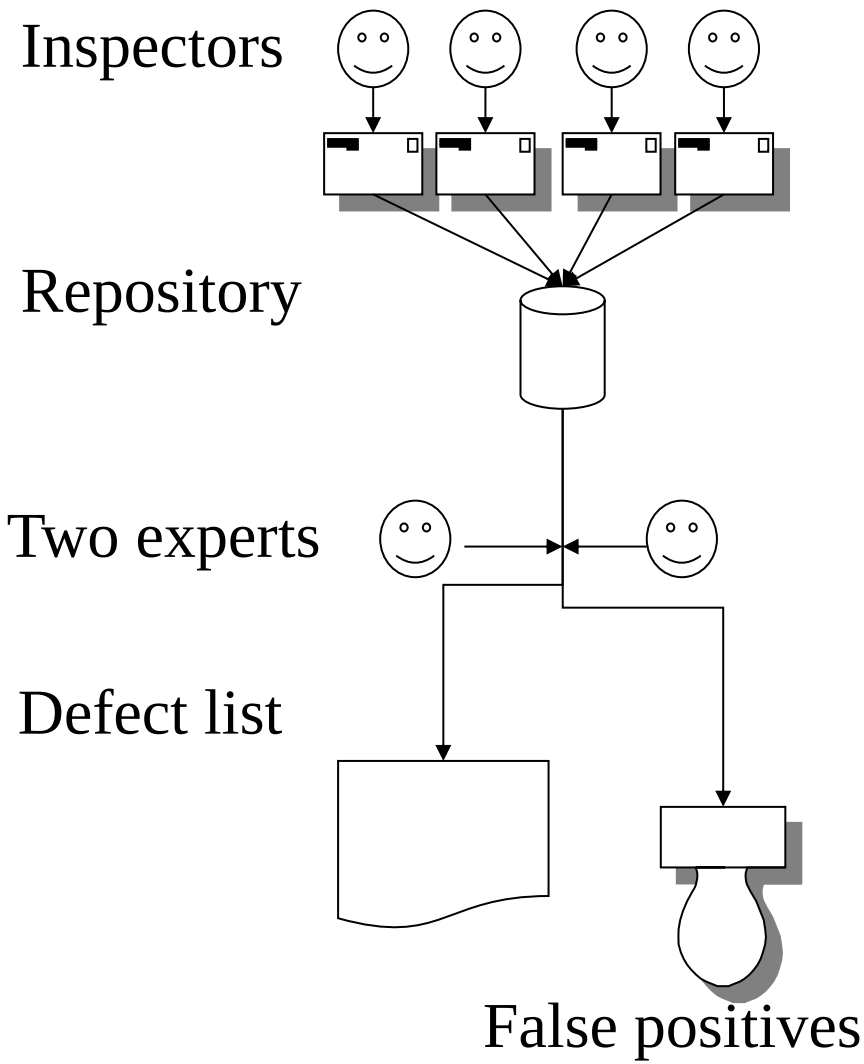
6.Ambiguity in Output Example:
- •The example provided for the heading and first row seems incomplete or
- •unclear. The data below the "Trial 11 1" line is missing or unspecified.

7.Inconsistent Sub-Process Terminology:
- •The term "value range" could be interpreted differently depending on the
- •reader. Aligning terminology with standard plotting conventions might avoid
- •confusion.

8.Axis Labeling:
- •The specification does not mention explicitly labeling the axes, which is essential for interpretability
- •in plots.

”Optimal” method

kristian.sandahl
@liu.se



Summary - What have we learned today?

kristian.sandahl
@liu.se

- Inspections rule!
- Inspections are expensive