SENG321: Requirements Engineering

Requirements inspections/technical reviews

Dr. Daniela Damian

thesegalgroup.org

danielad@uvic.ca

Outline

Requirements inspections: integral part of Software V&V activities or **Software Quality Assurance (SQA)** activities

Types of technical reviews

Guidelines for conducting requirements technical reviews (inspections)

A slight detour: Technical Reviews

Technical reviews conducted during:

- specification
- design
- Coding

Other names:

- (Fagan) inspections
- Walkthroughs
- Round robin reviews

Benefits of technical reviews

Errors found	Number	Cost unit	Total
Reviews conducted			
During design	22	1.5	33
Before test	36	6.5	234
During test	15	15	315
After release	3	67	201
			783
No reviews conducted			
Before test	22	6.5	143
During test	82	15	1238
After release	12	67	804
			2177

[Roger Pressman, Software Engineering: A practitioner's approach, 1997]

Research (Univ. of Maryland) has continued to show since the benefits of technical reviews over automated testing

Characteristics of technical reviews

Well defined roles

Typically no more than 6-7 participants

Duration: no more than 2 hours

Well-defined and agreed outcome

Seek to uncover errors close to the time they might have been produced

Modern code reviews
make these seem heavy weight…

Where did it all start?

Fagan inspections, 1970s

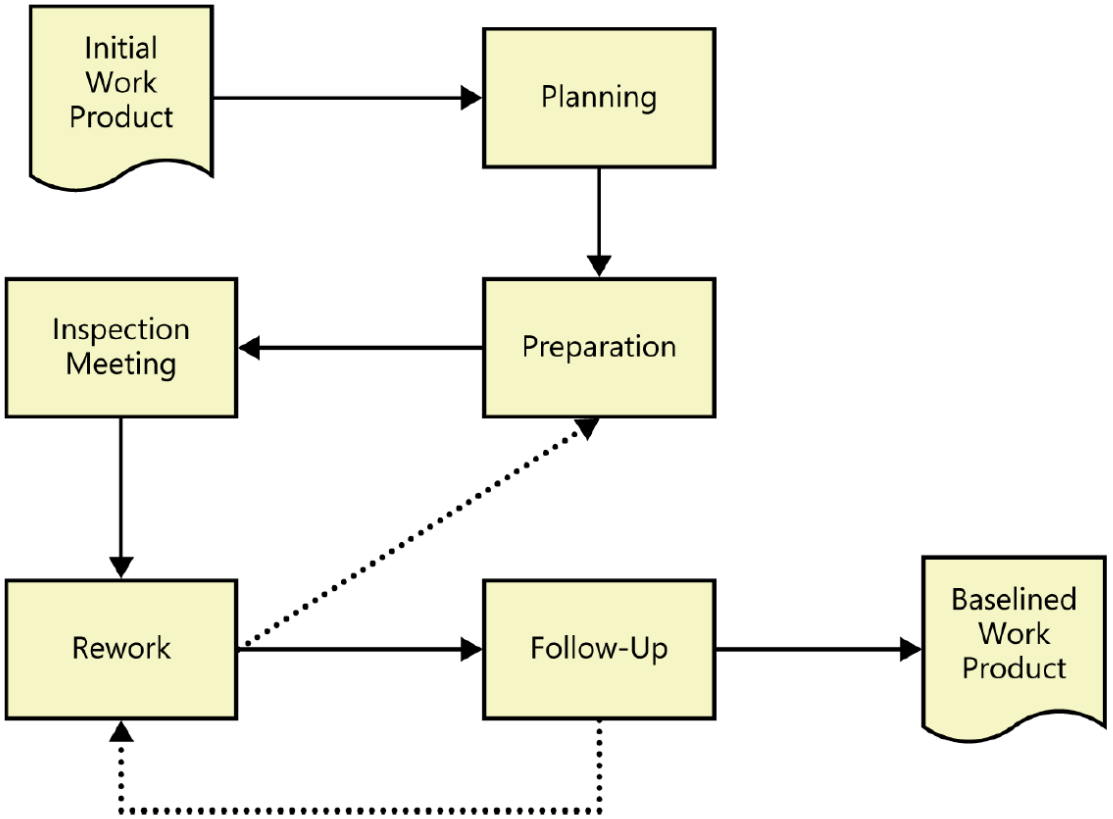

FIGURE 17-2 Inspection is a multistep process. The dotted lines indicate that portions of the inspection process might be repeated if reinspection is necessary because of extensive rework.

Fagan inspection process-- Phases

Planning

Prepare material, educate inspectors, schedule the meeting

Overview and Preparation

Present the overview to participants, read the material to identify defects

Inspection meeting

Find defects! Note but do not solve problems

Process improvement

Learn from current inspection to improve next inspection

Rework

Author reworks all defects

Follow up

Moderator verifies all fixes; product is re-inspect if 5% of document is reworked.

Roles in technical reviews

Walkthroughs

- Leader
- Recorder
- Reader (implementer)
- Other reviewers

Fagan inspections

- Moderator
- Designer
- Coder/implementer
- Tester

Requirements reviews

Round robin (more useful)

- Encourages equal participation
- Participants roughly at the same level of knowledge

Modern code reviews

A lightweight version of the Fagan inspections, do not require synchronous meetings or large number of reviewers

Supported by online tools for code review that allows annotations and summarization of issues

Adopted by geographically distributed teams and Open-source projects

However, effective mostly in *code* reviews; Round robin or the more structured, synchronous reviews are recommended for reviews/ inspections of requirements or design documents

Code Reviews from the Google SWE Book

Be **Polite** and **Professional** (trust and respect)

Write **small changes** (reduces downtime)

Write good **change descriptions**

Keep Reviewers to a **Minimum**

Automate Where Possible

Code Reviews from the Google SWE Book

Checks for code **correctness**

Ensures the code change is **comprehensible** to other engineers

Enforces **consistency** across the codebase

Psychologically promotes team **ownership**

Enables **knowledge sharing**

A Requirements review Checklist (see textbook page 339)

Completeness

- ❑ Do the requirements address all known customer or system needs?
- ❑ Is any needed information missing? If so, is it identified as TBD?
- ❑ Have algorithms intrinsic to the functional requirements been defined?
- ❑ Are all external hardware, software, and communication interfaces defined?
- ❑ Is the expected behavior documented for all anticipated error conditions?
- ❑ Do the requirements provide an adequate basis for design and test?
- ❑ Is the implementation priority of each requirement included?
- ❑ Is each requirement in scope for the project, release, or iteration?

Correctness

- ❑ Do any requirements conflict with or duplicate other requirements?
- ❑ Is each requirement written in clear, concise, unambiguous, grammatically correct language?
- ❑ Is each requirement verifiable by testing, demonstration, review, or analysis?
- ❑ Are any specified error messages clear and meaningful?
- ❑ Are all requirements actually requirements, not solutions or constraints?
- ❑ Are the requirements technically feasible and implementable within known constraints?

Quality Attributes

- ❑ Are all usability, performance, security, and safety objectives properly specified?
- ❑ Are other quality attributes documented and quantified, with the acceptable trade-offs specified?
- ❑ Are the time-critical functions identified and timing criteria specified for them?
- ❑ Have internationalization and localization issues been adequately addressed?
- ❑ Are all of the quality requirements measurable?

Organization and Traceability

- ❑ Are the requirements organized in a logical and accessible way?
- ❑ Are all cross-references to other requirements and documents correct?
- ❑ Are all requirements written at a consistent and appropriate level of detail?
- ❑ Is each requirement uniquely and correctly labeled?
- ❑ Is each functional requirement traced back to its origin (e.g., system requirement, business rule)?

Other Issues

- ❑ Are any use cases or process flows missing?
- ❑ Are any alternative flows, exceptions, or other information missing from use cases?
- ❑ Are all of the business rules identified?
- ❑ Are there any missing visual models that would provide clarity or completeness?
- ❑ Are all necessary report specifications present and complete?

FIGURE 17-4 A defect checklist for reviewing requirements documents.

Requirements Reviews Tips

Plan the examination (no need to start from the top of the page!)

Start early (we begin with Business Requirements!)

Allocate sufficient time (read before the meeting…)

Set review scope (decide focus of attention)

Limit re-reviews

Prioritize review areas (start with high risk requirements)

Guidelines for technical reviews

Develop a checklist for each work product that is likely to be reviewed.

Review the product, not the producer. Be polite and professional

Set an agenda and maintain it.

Limit debate and rebuttal.

Enunciate problem areas, but don't attempt to solve every problem noted.

Guidelines for technical reviews

Take written notes.

Limit the number of participants and insist upon advance preparation.

Review your earlier reviews and the current review process

Allocate resources and time schedule for technical reviews

Conduct meaningful training for all reviewers.