05. Formal Inspections

Formal Inspections - Code Complete, McConnell, Steve pág. 525.

An inspection is a specific kind of review that has been shown to be extremely effective in detecting defects and to be relatively economical compared to testing.

- Inspections were developed by Michael Fagan and used at IBM for several years before Fagan published the paper that made them public.

- Checklists focus the reviewers' attention on areas that have been problems in the past.

- The inspection focuses on defect detection, not correction.

- Reviewers prepare for the inspection meeting beforehand and arrive with a list of the problems they've discovered.

- Distinct roles are assigned to all participants.

- The moderator of the inspection isn't the author of the work product under inspection.

- General management doesn't attend the inspection meeting

- Data is collected at each inspection and is fed into future inspections to improve them.

**Benefits**

Designers and coders learn to improve their work through participating in inspections, and inspections increase productivity by about 20 percent (Fagan 1976, Humphrey 1989, Gilb and Graham 1993, Wiegers 2002).

Inspections will take up about 10–15 percent of project budget and will typically reduce overall project cost.

Inspections can also be used for assessing progress, but it's the technical progress that is assessed. That usually means answering two questions: Is the technical work being done? And is the technical work being done well? The answers to both questions are byproducts of formal inspections.

**Roles During an Inspection**

One key characteristic of an inspection is that each person involved has a distinct role to play.

Here are the roles:

**1. Moderator.**

The moderator is responsible for keeping the inspection moving at a rate that's fast enough to be productive but slow enough to find the most errors possible.

Tasks:

* distributing the design or code to be reviewed,
* distributing the inspection checklist,
* setting up a meeting room,
* reporting inspection results,
* and following up on the action items assigned at the inspection meeting.

**2. Author**.

The person who wrote the design or code plays a relatively minor role in the inspection. Part of the goal of an inspection is to be sure that the design or code speaks for itself. If the design or code under inspection turns out to be unclear, the author will be assigned the job of making it clearer.

The author can explain why things that seem like errors are actually acceptable.

If the project is unfamiliar to the reviewers, the author might also present an overview of the project in preparation for the inspection meeting.

**3. Reviewer**.

A reviewer is anyone who has a direct interest in the design or code but who is not the author. A reviewer of a design might be the programmer who will implement the design. A tester or higher-level architect might also be involved. The role of the reviewers is to find defects. They usually find defects during preparation, and, as the design or code is discussed at the inspection meeting, the group should find considerably more defects.

**4. Scribe**.

The scribe records errors that are detected and the assignments of action items during the inspection meeting. Neither the author nor the moderator should be the scribe.

**5. Management**.

Including management in inspections is not usually a good idea. The point of a software inspection is that it is a purely technical review. Management's presence changes the interactions: people feel that they, instead of the review materials, are under evaluation, which changes the focus from technical to political.

Similarly, under no circumstances should inspection results be used for performance appraisals.

Overall, an inspection should have no fewer than three participants. It's not possible to have a separate moderator, author, and reviewer with fewer than three people, and those roles shouldn't be combined.

Limit an inspection to about six people because, with any more, the group becomes too large to manage. Researchers have generally found that having more than two to three reviewers doesn't appear to increase the number of defects found (Bush and Kelly 1989, Porter and Votta 1997).

**General Procedure for an Inspection**

An inspection consists of several distinct stages:

**1. Planning**.

The author gives the design or code to the moderator. The moderator decides who will review the material and when and where the inspection meeting will occur; the moderator then distributes the design or code and a checklist that focuses the attention of the inspectors.

Materials should be printed with line numbers to speed up error identification during the meeting.

**2. Overview.**

The design or code should speak for itself; the overview shouldn't speak for it.

**3. Preparation.**

Each reviewer works alone to scrutinize the design or code for errors. The reviewers use the checklist to stimulate and direct their examination of the review materials.

Some organizations have found that inspections are more effective when each reviewer is assigned a specific perspective. A reviewer might be asked to prepare for the inspection from the point of view of the maintenance programmer, the customer, or the designer, perspective-based reviews might uncover more errors than general reviews.

variation in inspection preparation is to assign each reviewer one or more scenarios to check. Scenarios can involve specific questions that a reviewer is assigned to answer, such as "Are there any requirements that are not satisfied by this design?" A scenario might also involve a specific task that a reviewer is assigned to perform, such as listing the specific requirements that a particular design element satisfies.

**4. Inspection Meeting.**

The moderator chooses someone other than the author to paraphrase the design or read the code (Wiegers 2003). All logic is explained, including each branch of each logical structure. During this presentation, the scribe records errors as they are detected, but discussion of an error stops as soon as it's recognized as an error.

- The scribe notes the type and the severity of the error, and the inspection moves on.

- The rate at which the design or the code is considered should be neither too slow nor too fast. If it's too slow, attention can lag and the meeting won't be productive. If it's too fast, the group can overlook errors it would otherwise catch.

- Don't discuss solutions during the meeting.

The group should stay focused on identifying defects. Some inspection groups don't even allow discussion about whether a defect is really a defect. They assume that if someone is confused enough to think it's a defect, the design, code, or documentation needs to be clarified.

- The meeting generally should not last more than two hours. Reviewers can't concentrate for much more than about two hours at a time. For the same reason, it's unwise to schedule more than one inspection on the same day.

**5. Inspection Report.**

Within a day of the inspection meeting, the moderator produces an inspection report (e-mail or equivalent) that lists each defect, including its type and severity.

- it's used to develop a checklist that emphasizes problems specific to the organization.

- you collect data on the time spent and the number of errors found over time, you can respond to challenges about inspection's efficacy with hard data.

- Data collection is also important because any new methodology needs to justify its existence.

**6. Rework.**

The moderator assigns defects to someone, usually the author, for repair. The assignee resolves each defect on the list.

**7. Follow-Up.**

The moderator is responsible for seeing that all rework assigned during the inspection is carried out. Depending on the number of errors found and the severity of those errors, you might follow up by having the reviewers reinspect the entire work product, having the reviewers reinspect only the fixes, or allowing the author to complete the fixes without any follow-up.

**8. Third-Hour Meeting.**

You can hold an informal, third-hour meeting to allow interested parties to discuss solutions after the official inspection is over.

**Checklists**

- you'll notice that certain kinds of errors occur more frequently than other kinds. Create a checklist that calls attention to those kinds of errors so that reviewers will focus on them.

- Over time, you'll find kinds of errors that aren't on the checklist; add those to it. You might find that some errors on the initial checklist cease to occur; remove those.

- some clues about trouble areas in which its programmers need more training or support. Limit your checklist to one page or less. Longer ones are hard to use at the level of detail needed in an inspection.

**Egos in Inspections**

The point of the inspection itself is to discover defects in the design or code. It is not to explore alternatives or to debate about who is right and who is wrong. The point is most certainly not to criticize the author of the design or code. The experience should be a positive one for the author in which it's obvious that group participation improves the program and is a learning experience for all involved.

- the moderator should make their inappropriateness unmistakably clear.

- Acknowledging a criticism doesn't imply that the author agrees with the content of the criticism. The author should not try to defend the work under review. After the review, the author can think about each point in private and decide whether it's valid.

- Reviewers must remember that the author has the ultimate responsibility for deciding what to do about a defect.

---------------------------------

Práctica:

* Que todo grupo ejecute primero los casos test.
* Los casos test tb se refactorizan.
* 5 cartas con 5 numeros distintos: 2 revisores.
* El escribiente publica en el drive de la unidad 7 la checklist
* El moderador publica el informe después de la corrección /valoración del autor/a.