

# Every change now needs a meeting it did not need before

A change that one person could once make in an afternoon now takes a calendar invitation, a roomful of people and a wait.

|              |                                                                                                                                                                                                                 |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Author       | Amit Govil, Founder and CEO, XONIK                                                                                                                                                                              |
| Published    | October 22, 2026                                                                                                                                                                                                |
| Reading time | 8 min read                                                                                                                                                                                                      |
| Online       | <a href="https://xonik.com/insights/perspectives/every-change-now-needs-a-meeting-it-did-not-need-before/">https://xonik.com/insights/perspectives/every-change-now-needs-a-meeting-it-did-not-need-before/</a> |

## IN BRIEF

A small change needs a meeting when what it touches is known only to people, because gathering them is the cheapest way to find out. The meeting then gains authority, grows an invitation list and sets the pace of change. Writing dependencies down as contracts, and letting a team decide alone inside them, returns routine changes to the team.

## KEY FINDINGS

1. A coupling that is not written down is negotiated again in a meeting whenever one side wants to change something.

Conway's argument is that a system's structure copies the communication structure of the organization that designed it, and every connection is a standing agreement between people. The meeting is a lookup of facts, done by hand.

2. The meeting turns from informing the change to approving it.

Once an owner has been caught out by an unannounced change, a no becomes protection. The invitation list lengthens a name at a time and never shortens, and the cadence then sets the pace of change.

3. External approval does not make changes safer.

DORA's research found a negative effect on software delivery performance and no evidence that formal external review lowers the rate of failed changes. Heavyweight approval pushes toward larger batches released less often.

4. Sorting what people say in the meeting shows what to write down, assign or send.

A fact about a dependency belongs in a written contract, a judgment about risk belongs to a named owner with a stated rule, and news is a notice. The fact only one person holds is the fragility itself.

## 5. Contracts between teams let a team decide alone again.

A team may change anything inside its own part while the contract it keeps with other parts holds, and a contract test that fails when an output changes form gives the answer a meeting would have given.

## Why does a small change end up needing a meeting?

Because the person making the change cannot see what it touches, and the people who can are in different places. A change to how a customer's status is calculated may affect billing, the portal, a report and a partner feed. If knowledge of those dependencies is written down, the person reads it. If it is held in four heads, the efficient way to consult four people at once is to put them in a room. The meeting is a sensible tool applied to the wrong job. It is a lookup of facts, done by hand, repeatedly.

Melvin Conway's 1968 paper explains why the facts end up spread this way. He argued that a system's structure copies the communication structure of the organization that designed it, and in his account every connection between two parts of a system exists because the two groups responsible for them negotiated an interface specification (How Do Committees Invent?, Datamation, April 1968). Each coupling is therefore a standing agreement between people. When the agreement is not written down, it is negotiated again in conversation whenever one side wants to change something, and a meeting is where that conversation is held.

## How does the meeting grow into the change process?

It starts with a surprise. A change goes out and breaks something its author could not have known about, owned by someone who was not asked. The remedy is obvious and cheap: invite that owner next time. The list lengthens by a name at a time and never shortens, because removing someone is a risk with a visible cost, while keeping them costs a little of everybody's time, which no one books.

Then the meeting changes function. Early on, attendees tell the proposer what they know. Once an owner has been caught out by an unannounced change, they begin to say yes or no, because a no is protection and an absence gets blamed. From that point the meeting is not informing the change but approving it. The cadence then sets the pace. A change waits for the next slot, is held over if a question cannot be answered in the room, and is grouped with others so as not to miss another round. Exhibit 1 draws the sequence and marks the point where informing turns into approving.

DORA's research, summarized in its guidance on streamlining change approval, finds that approval by an external board or senior manager has a negative effect on software delivery performance, and found no evidence that formal external review lowers the rate of failed changes. The reason it gives is that heavyweight approval pushes toward larger batches released less often, with a higher impact on production. The same page observes that a central board is good at broadcasting change but that people far from the work may not grasp its implications, and it lists responding to problems by adding more process as a pitfall that drives up lead times and batch sizes in a vicious cycle.

## EXHIBIT 1

## How an invitation list becomes a change process



No one decides that the meeting becomes the approver. It happens at the step where attending turns from a courtesy into a protection.

## Why does fragility produce meetings instead of repairs?

A repair is a project: it needs a sponsor, a scope and a place in a budget. A meeting needs an invitation. After an incident, the meeting is the only response anyone can start the same afternoon, and its cost, the hours of senior people, appears on no ledger. The organization buys protection against a fragile system in the one currency it does not count. The fragility is unchanged, because a meeting catches errors by asking people whether they can think of one, and what makes the system fragile is that nobody can reliably think of all of them.

There is a legitimate need inside this. One common requirement is that a change be approved by someone other than its author. DORA notes this separation of duties, and says peer review during development can meet it, with reviews and approvals recorded in the team's own tools, supported by automated tests and monitoring that catch bad changes early. The need for evidence of authorization does not justify a standing board. It justifies a recorded second pair of eyes.

## What does it look like in a business?

A telecoms operator wants to change how a tariff rule applies to customers who move between plans partway through a billing cycle. The rule is held in the billing system, copied into provisioning, shown in the customer portal and read by a revenue report. The proposer books the change meeting and brings the billing owner. The provisioning owner asks a question about part-period charges that nobody in the room can answer, and the item is held over while someone finds the portal's developer, who is away, and the report's owner, who learns of the change from the minutes. A change that took an afternoon to build takes repeated cycles to approve, and each cycle adds an attendee.

The meeting is a sensible tool applied to the wrong job: a lookup of facts, done by hand, repeatedly.

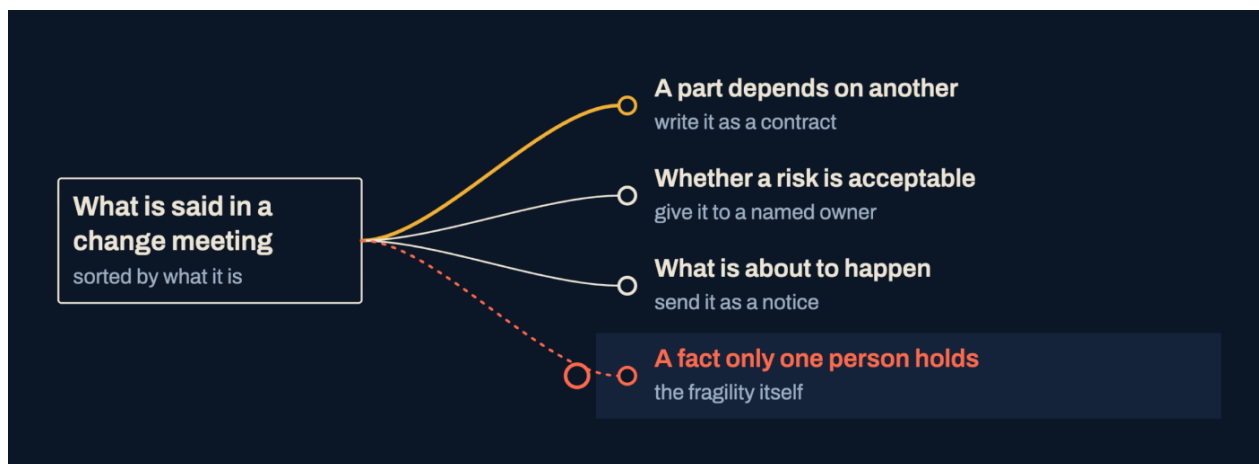
## What is the meeting actually doing?

Listen to a change meeting and sort what is said, as Exhibit 2 does. Some of it is a fact about a dependency: the portal reads that field. That belongs in a written contract between the parts, because it will still be true next month. Some is a judgment about risk: a change to that is not allowed during month-end close. That belongs to a named owner with a stated rule, so it can be applied without convening anyone. Some is news: that job moves on Thursday. That is a notice, which is sent and not discussed.

What is left is the fact that only one person holds. It tells you what is wrong with the system, because it is the fragility itself, and the right response is to write it down at once and then ask why it was known to one person only.

### EXHIBIT 2

#### What people say in a change meeting



Only the last kind needs a person in the room, and not for long. The rest have better homes than a meeting.

## What lets a team decide alone again?

Boundaries with contracts. A team may change anything inside its own part so long as the contract it keeps with other parts still holds. The contract says what crosses the boundary, in what form and with what meaning, and what each side may assume. D. L. Parnas's 1972 criterion for dividing a system is that each module hides one design decision from the rest, so that a change to that decision stays inside one module. A contract is the same idea applied between teams.

Two things make it enforceable. A contract test that fails when an output changes form gives the team the answer a meeting would have given, without anyone being called. And the map of dependencies lives where changes are made and is updated by the same change. DORA recommends the same direction: peer review and automated checks close to the work, and a central board whose role moves to coordination, improving the process, and the trade-offs that need sign-off from higher in the business.

The meeting remains, for changes to a contract and for decisions that need a trade-off made above the team. It is smaller, and who needs to attend is known in advance. Test your own by asking each attendee of the last change meeting what would have happened had they not come. The answer is either a dependency you can write down or a habit you can stop.

## Sources read for this piece

Each source was read for the passage cited, on 20 October 2026. The way a meeting grows is described from delivery work, as a pattern, with nothing counted.

1. [Melvin E. Conway, How Do Committees Invent?, Datamation, April 1968, as reprinted on melconway.com: structure of a system and of its design organization, interfaces as negotiated agreements](#)
2. [DORA, Capabilities: Streamlining change approval, last updated 30 October 2025: effect of external approval, separation of duties, role of a change board](#)
3. [D. L. Parnas, 1972, Communications of the ACM, December issue: modules that hide design decisions](#)

### USE THIS

[Review your architecture with the checklist](#)

[Read why the third change costs more than the first](#)

List the people invited to your last change meeting, and beside each name write the one thing that person knew that the others did not. That list is your map, in outline.

---

This document was made from the article at

<https://xonik.com/insights/perspectives/every-change-now-needs-a-meeting-it-did-not-need-before/>. The online article is the current version.