Herbsleb & Grinter: Conway’s Law Revisited
ICSE 99

Product and Process

• Conway and Parnas relate work breakdown to product; the process also has a structure
  > Product, Organization, Process, all entangled and reflecting each other
• But ...
  • “there will always remain many decisions that cannot be made ahead of time, unanticipated problems, details to be filled in, mistakes to be corrected and recovered from”
• These demand effective informal communication

The “water cooler”
(on the west coast, substitute an espresso machine)

• “What appear to be merely ‘casual conversations around the water cooler’ often serve to informally exchange the kinds of information and experience that are critical to project coordination.”

• Especially: The connection you didn’t even know to ask about.

Integration Plan

• ... “a plan for integrating ... consisted of 40 steps”
• “no surprise that the components were not ready for integration on the schedule described in the plan”
• “We chopped and changed as things became ready.”

– Side question: How would you avoid this situation, considering the “virtual impossibility of predicting the effort and timing of a new product being developed in a new organization”
Interface specs

- “Interfaces were specified primarily by event tracing, or ‘fence diagrams’ that showed sequences of messages among processes”
- “it became clear that the interface specifications lacked many essential details”
- “not until the initial attempts to make the pieces work together that these alternate assumptions were exposed”
  
  How do you avoid this?

Change management (revision control, ++)  

- “separation allowed two change management processes to evolve independently, one for each site.”
  
  Don’t. Ever. And also don’t let people bypass or ignore the change management system. You will suffer if you do.

Documentation

- “Under time pressure to build the system, the developers proceeded with coding, and slowly the code diverged from the design.”
- “The developers were left with one accurate source of documentation – the code itself.”
- “As [some detail] was not included in the documentation, the headers were not correct, and it took weeks to figure out that this was the problem”

Barriers to informal communication

- Lack of unplanned contact
- Knowing who to contact about what
- Cost of initiating contact
- Ability to communicate effectively, and
- Lack of trust, or willingness to communicate openly.
Unplanned Contact

- “just the usual sort of friendly exchange, common in the workplace ... seem to be surprisingly important in keeping projects coordinated”
- “project members are often unaware of the assumptions they are making, or that others might be making conflicting assumptions”

Communication & technology

- Face-to-face is most effective ... but if you can’t have that ...
- “every conference call I walked out of, if I ask somebody what do you understand from it, and they say, ‘I don’t know’.”
- “E-mail was ... preferred ... , especially [for] non-native English speakers”
  - Today: Video calls help some; IM may or may not help. Still very short of a face-to-face chat in front of a whiteboard

Knowing whom to contact

- “no straightforward way to find out who was responsible for the component on the other site”
- “developers [who] had spent ... time at the other site ... became ‘contact people’ or ‘liaisons.’”

Culture differences

- “the Germans have a greater tendency to take hierarchical relationships a bit more seriously, expecting and receiving a greater degree of direction from managers and supervisors”
  - Note: Besides national cultures, differences also exist between the cultures of different companies and professions. Bankers are from Jupiter, programmers are from Mars.
Trust and Confidence

• “they just did not seem to be able to make progress until they had worked together face to face”
• “manifested itself in ‘uncharitable’ interpretations of behavior”

— We observed a similar need to build trust and confidence in globally distributed CIS 423/523 projects last year, without recourse to travel. Resolved, but not easy.

Conclusions

• Software design promotes independence
  • Conway’s law, Parnas’s guidance on software architecture: Limit dependence to simple, precise interfaces that correspond with communication structure
• But you’ll never get it entirely right
  • Informal communication is needed because things change and break, and there are always details to be filled in and negotiated. Geographic, linguistic, and cultural distance makes it harder.