



# Open Process

The new frontier



# Outline

---

- Motivation and background
- Open Process examples
- Ethnographic study
- Conclusions

# Benefits of “Openness”

- Open Systems
  - Interoperability
  - Network effects
  - Plug-and-play
- Open Source
  - Continuous code review
  - Early user feedback
  - Network effects

# Open Process

- Exemplified by Free Software and Open Source
  - Open (archived) discussions on design and implementation decisions
  - Source code available for anyone, including users
  - Inclusive communities

# Open Process Examples

- Corporate Source
  - Grass roots effort at HP Labs
- Collaborative Development Program (CDP)
  - Executive sponsored (mandated) program at HP Printer Group

# Open Process Study

(with Catharina Melian,  
Stockholm School of Economics)

- Ethnographic studies at Hewlett-Packard different sites October 2001—April 2002  
Unique access to corporate data, no earlier studies
- Inductive approach. Constructed template with questions based on participatory observations, and thereafter semi-structured interviews based on the template
- 50 interviews with developers across different geographical sites, managers at different levels, human resource, and third parties.

# Openness—the manager's view

- *Advantage: The fact that people know they can go somewhere and get information, it's efficient if you know where to go, because you don't have to walk over to someone's desk. If they are not there, leave them a note, or leave them a voice mail, and then wait for hours before they get back to you. You can actually go and it forces us as a team to have our documentation more in line, more simplified. Overall it's been a really good move.*

# Openness—the developer's view

- *And that's maybe why I like the closed conference room on the black board. Nobody else is listening. It's just a limited number of people that people feel comfortable with and so you have a good point that's one of the reasons why that wouldn't work. On the other hand you could limit the virtual meeting to people that are just doesn't have to include the whole world it could be limited to a group of people that one is comfortable with and it could work too*

# Reluctance of developers

- Experts are hesitant to disclose all their knowledge, since that may jeopardize their competitive advantage.
- *There's a multiplicity of reasons why we need to share. Knowledge grows as it's shared. If someone solves a problem here and other people see how it's done, it spurs their thinking and they begin to find ways to come up with even more creative solutions and there's kind of a symbiotic thing that starts to happen where one division or another, or different teams in different co-located areas start to actually help each other get better, technically. The downside is there's a lot of fear about 'he's going to steal my work' or 'this isn't protecting HP's intellectual property, if we let a third party have access to it.'*

# The perceived threat of openness

- *But, for example this new person that was on that became a member of the CDP community yesterday committed a file and I got notified of it last night. Now I'm really concerned that this file is legit. CDP and CVS just like our previous revision control systems do not substitute for common sense and when and what kind of changes need to be made. So unlike firmware that where firmware get to be changed and oh yes I made a mistake I can change it back. If an error isn't caught in an ASIC, it could result into a half million to a million dollars worth of error with us if the masks of all of the fabrication processes. There ought to be some control on access. Or there needs to be some sort of a system where somebody reviews changes before they are actually committed.*

# Fear of Openness

- *I think people are very conscious about anything that's permanently recorded, of voicing it. Many people; I suggested this at one time also. A very, in the best technical meetings there is a flow that develops, and an awful lot of technical details come out in a very short amount of time. And I've found that frequently after the meeting we would lose some of those details, that people would forget. People would solve this or didn't really dismiss this idea... Somebody suggested once let's videotape the meetings solely for the purpose of remembering afterwards what was said. And maybe we only need the tapes for a week or something. Absolutely no one wanted that. Nobody wanted to be recorded, people were afraid of being on record as saying something and then being held accountable for it later. And that's not at all what it was for. It was just for the purpose of capturing the ideas so that we could slowly unwind it afterwards and remember everything that was said, because some of it happened so quickly. Nobody wants to be seen recorded asking a stupid question.*

# Third party view on Openness

- *HP is known as the market leader in development of ink jet and laser jet printers. Motorola are market leaders in microprocessor technology as well as IC development. We have technologies that we would like to reuse, that we'd like to use somewhere. HP has their proprietary system code, architecture. How can we dovetail the two in is the point of interest. Both at HP that they're telling us and at Motorola that we'd like to provide them. Why should HP waste engineering effort redeveloping things that Motorola already could provide? What if we could have a library available and accessible and have project managers from many HP sites looking at it and somebody is going to say: This is interesting! We try it. Use it. It works. I want that in my next business quotation. That's the best-case scenario!*

# Faster product lifecycles

- *The ability for HP external clients, our business partners to have access to information that they would not normally have access to, and for HP to access the information that the business partner had in a quick and efficient manner. Not having to worry about transferring data slowly through a firewall. We have found that, an example is we deal with some customers overseas and in Europe. In order to get a package from there to here that has an CD image could take 4 to 7 business days. For us, we can transfer that data in 2 hours. The customer can get it back, we can get it back from the customer or business partner. And the development time has been cut down remarkably in this. It's a great benefit. It causes a lot of stress, because now that the developers, when they leave at night, they used to say oh, I have three or four days until I'm going to get a response back on an image. On an idea now, they leave at 5:00 Boise time. When they come back to work the next day, they have things that they need to fix already, for an image that they just created the night before. So the product cycle is just way speeded up. Way faster.*