Human Issues of Processes

Resistance to change: Issues with early and often delivery

Jutta Eckstein
http://jeckstein.com
jeckstein@acm.org

Sometimes people are not consciously resistant to change, actually they want to change, they want to embrace change but their old traditions keeps them unconsciously from changing. It's just a habit to live a process the old way. Sometimes the change feels to them so dramatic that it is beyond their imagination. For example, I discussed with a team leader a couple of times what is meant by early and often delivery, small release, continuous integration, time-boxing, iterative and incremental development. She seems to understand; at least this reflected the way she contributed to the discussion. And although we even talked often about time frames – when it came down to define the deadline for the first small release, she gave me a date five months away. Only then I realized, we still have a long way to go and probably people will fall back in their old habits, just because they can't imagine it in any different way although they don't really want to.

We tried to handle this by even pushing this further, which must have been completely absurd to the whole team. We decided to have a first release after three weeks, with three one-week iterations. Well we lost already one week for getting started, but we didn't move the deadline of the release – it was a time-box. After the next week we asked what the teams delivered to the build, well there was almost nothing, but this feedback round started some change. The teams got ambitious to be able to deliver something for the upcoming week, they started also to be honest and tell their fears and the reasons why they think this just can't work. We discussed these different things, by always pointing them to our main objective: Making as many errors as we can early on, rather than late in the project. And believe me, we were good at that (making many errors). This gave the teams the feeling that even if they weren't able to contribute to the build in this round, they still contributed a lot to the whole process. This helped us a lot to keep moving and develop a team spirit.

I guess the bottom line from this approach is: to change something so radically (shock-effect) that the real change you want to make, doesn't seem so strange anymore.

Well this is only what happened inside the company, but the project has additionally another big issue outside the company: outsourcing. Some teams who are not part of the company, would like to be better integrated, but they develop far away. So a lot is going on via phone and email, which is not really satisfying. Other – actually most of those not on-site – companies refer to contribute to small releases; they want to stick to develop against milestones and don't want to see anything from the project along their way. For now we try to be sensitive and remind them on the advantages they will have if they also live short cycles. But we are still trying to find a way to (a) have the contracts specific enough so they have to deliver early and often and (b) to make them believe that this is a successful way to develop software.

It's so much easier to convince people if you see them everyday, than with the ones who don't see you and the project. They have extreme problems trusting a new approach, because they hardly see and feel the advantages of the change. Well and of course, because they are not onsite it's hard to influence them. However I'm pretty sure that I will be able to provide some news regarding to this topic when time comes down to the workshop.