Session 1: People, Processes, and Practice

Dewayne E. Perry

Software and Systems Research Center
AT&T Bell Laboratories
Murray Hill, NJ 07974 USA

Abstract

Session 1, entitled ‘People, Processes and Practice’, was divided into two discussion sections. The first included discussions of how people are treated and supported in process models in terms of control and guidance, issues of context switching, and aspects of informality. The second focused primarily on deviating from the defined processes, detecting and resolving those deviations, and tailoring processes.

Alex Wolf began the session discussion by reviewing his presentation from ISPW8 on Humans in the Process (see the session summary in the ISPW8 proceedings). While the intent of the review was to set the stage for this session’s discussion, it was clear that there were still a number of contentious issues.

Not surprisingly, the issue of control was raised immediately. On the one hand, the only reliable way to achieve a desired outcome is to control the conduct of those executing the process. In opposition is the position that any control only “turns people off”. A related issue is that whether the processes are goal-directed or step-directed, prescriptive, proscriptive, or descriptive. Alfonso Fuggetta rightly pointed out the interaction style is separable from the modeling style. There are tendencies for coupling, but they are not necessary ones.

Watts Humphrey claimed that we should focus on informing people and providing knowledge and skills. Moreover, in terms of their process support, we should provide them with instrumentation. With the data, we get improvement. Bob Balzer stated that we need both forward and backward chaining so that we can provide both guidance and understanding. If we do that then we have support that is much more open to what Watts is suggesting. It was pointed out by Manny Lehman that we rarely set a goal and head straight for it. We start out and apply corrections. Having the right information at the right time, as Watts suggested, is a major factor in the process of getting there.

According to Lee Osterweil, context switching is an important problem: humans do not do it well and we must keep this in mind when we consider humans as process agents. There was quite a bit of disagreement with Lee over this. Simon Kaplan claimed that we switch very well: we work in so many overlapping contexts concurrently. Two kinds of context switching were proposed by Alex Wolf: those where we switch because of interrupts and those where we switch by design. In either case, we have relatively little data about the costs of switching contexts. If it takes 5 to 10 minutes to re-engage, then according to Larry Votta, we are not very good at it.

Gail Kaiser asked whether or not we want to model explicitly context switches and interrupts and likened the situation to context switching in operating systems. Dewayne Perry’s response was that we do not want to model them. There is a difference between the mechanistic aspects of context switching in an OS and the kind of context switches we do while doing our processes. The important thing is to provide sufficient support for the people doing the process to be able to move easily from one thing to another. Mark Dawson took a different position: we want to be able to talk about it, but don’t want to talk about every context switch. Larry Votta noted that there are different kinds of people: those who are monochronic and who work better on only one thing at a time, and those who are polychronic and who are comfortable at working on multiple things concurrently. The latter probably are better at context switching.

Swamy argued for [constraining] processes: it works in other fields—for example, in manufacturing. [But of course, you get much happier workers where they are allowed to switch what they do.] Bob Balzer pointed out [rightly] that Dewayne Perry’s real point was that you can not predict when the context switches will take place, but that you cannot support it without...
knowing that it happens. Bob went on to suggest that contexts are associated with threads and that we need to support the movement from one scoped space to another.

The thread of constraint versus creativity continued throughout this discussion where the former are seen as inhibiting the latter. Dewayne Perry brought up the example of the functional potter: there are constraints placed on the potter by the kinds of wares that he makes, but within those constraints there is a wide latitude for creativity and individuality. There is a basic fact, Alex Wolf pointed out, about software development: it is a design activity. Building the 747 involved people in the design with a fair amount of freedom. But when it came to wiring the plane, creativity was not allowed.

Mark Dowson suggested that we need notations for advice, where divergence from the modeled approach is not only allowed but supported. Note that this freedom is not inconsistent with Dewayne Perry's analogy of functional (potting) constraints: within the desired functional goals, the only other constraints are engineering ones. Manny Lehman noted that one of the things we are very good at is making associations of things previously not associated. Of course, if we have not foreseen them then we are not able to model them. This is another aspect of support that is needed for processes.

Watts Humphrey responded to Mark Dowson's suggestion on advice with the observation that people do not take well to advice, they usually take the opposite. They will however, often arrive at the appropriate conclusion on the basis of data and be excited about it. Kathleen Culver Lozo noted that her group had started with a task-oriented approach, but found that the people doing the process always talked in terms of product goals. They now use both templates and quality assurance goals with an emphasis on how they work together.

At this point just before the break, Lee Osterweil admitted that he agreed with Watts Humphrey and Bob Balzer and that bothered him a lot [general mirth]. There needs to be a coordination of what people do and what machines do, and, further, not all people should be in charge. In order to do planning and analysis, you have to understand the costs and effects of context switching. It is useful to remember that we have the ability to turn off certain kinds of interrupts — for example, no phone calls for the next hour.

After the mid-session break, Kathleen Culver Lozo gave a presentation on her experience with deviations from the defined processes. Alex Wolf had asked her to consider the following questions:

- Can people deviate from the process?
- Is it useful?
- How can they do it?
- How can the deviated process be reconnected to the defined process?

Kathleen's group defines and supports processes for a software production center in which there are a reasonable number of small to medium size projects. The kinds of observed deviations included the following:

- skipping a review,
- cutting scenario descriptions,
- using a different tool for a scenario,
- separating out OAM,
- doing information modeling before the data flow diagrams are done,
- not providing a project management model,
- writing functional requirements, and
- using a non-prescribed database.

The ones presented here were only a few of the many that occurred. There was approximately 25% deviation at a high level; at a lower level there might well have been more deviations [she was not sure about this]. Some of these were appropriate, some were stupid.

People deviate from the defined process for a variety of reasons: the process definitions omit or do not allow for relevant project contingencies; sometimes risks are taken; some process definitions are more amenable to deviations; people have good ideas, some of which are better than the defined processes; the process does not make sense either because of individual differences or because of lack of training; and finally, of course, some are due to a lack of commitment or interest.

Dewayne Perry pointed out that there is always a tension between best in class and the need for shorter time intervals, that sometimes risks are taken because other factors are more important. Of course, what is high risk in one context might well be low risk in another [Watts Humphrey]. Bob Balzer suggested the tension between the ideal and the practical as a source of deviation. And, of course, we cannot foresee all possible contingencies. Even if we could, according to Kathleen Culver Lozo, we would not want to make them all part of the standard process.

Kathleen continued by stating that the purpose of considering these deviations is not to act as the process police, but to be both proactive and reactive in partnership with the various projects. It is something
that needs organizational support and personal buy-in. The process engineers also need to be seen as part of the group, not as outsiders.

The problem with deviations is not detecting them—they are very easy to find—but what to do with them, choosing which of them will have the most significant and beneficial impact. The process of reconciling the differences between the defined processes and the deviations consists of three steps: detecting the differences via project plans, postmortems, modification requests, and process-generated project directories; determining the root causes of the deviations; and determining the resolutions. Resolutions may take one of three forms: no change—often the case where risk taking, lack of commitment or training is the cause of the deviation; a generic change—where, for example, there are good ideas or some typical contingency; and a local change—this allows for individual differences and the testing of potentially good ideas.

Gail Kaiser asked how they got people to report the deviations. Kathleen’s response was that they put most of their faith in what was actually produced, in what shows up in the directories. Variations in activities show up in the modification requests. Determining what the right deviations are is the responsibility of a review board. They prioritize the problems: if they are about templates or dramatic improvements they are considered major contributions and put into the standard processes; if they are minor, they are more likely to be deferred.

Mark Dowson brought up the example of using ISTAR to build ISTAR. We could plan at a fine granularity and replan easily instead of diverging more and more. There were good automated capture tools and good managers using them. Keeping the plans synchronous with reality was perfectly doable with classical techniques. We do have well-managed projects. One response to this was that we often feel that we are trying to compensate for bad management.

One solution to the problem of global versus local processes was suggested by Dewayne Perry, namely, to use stratification as a means of separating the two. A generic process is instantiated by binding in the local environment, tools and methods. This requires a more architectural approach than we have been taking.

There seemed to be a loose set of project management requirements. Simon Kaplan asked Kathleen where she saw automation helping out. Her response: where things are well understood and where interfaces are predictable. Simon thought that it was interesting that so much had been done without automation. Dewayne Perry pointed out that at ICSP1 the reported process improvements were done without automation; they relied on introducing well-understood software engineering techniques, exploiting the technical aspects of the product and experimenting with the organizational structure. There is a process engineering cost to be taken into consideration: reduction in interval requires a large process engineering cost, small improvements don’t [Alex Wolf].

Naser Barghouti indicated he was confused with this discussion—there are two aspects to processes: social and technical. We should not be in the social business, but in the technical. We don’t know about the social but do know about the computer based stuff. By monitoring the actual execution, we could find out a lot of the deviations. Larry Votta and Dewayne Perry disagreed with Naser and think that you can do something about the social side of process. Kathleen responded that the problem is not detection, but reconciliation. We had monitoring and got too much data which we could not effectively use. The focus needs to be on the important problems. Sergio Bandinelli agreed with Naser that much can be recorded automatically; however, Sergio thought should be given to what are valuable and relevant events.

Process formalisms can be used as means of helping people or as means of automation. Lee Osterweil thought that Kathleen’s group had found a process of tailoring that works. How is that tailoring done? Kathleen: we began with fine grain steps but found we could not keep up with the changes needed; we then changed the grain from a daily level to a weekly level.

Nazim Madhaji asked how they arrived at the definitions that they gave to the developers? Did you do analyses of projects or the best projects? Kathleen indicated that they had a set of best in class practices, a standard environment, and a knowledge of the type of projects [generally in the same domain and producing similar types of products]. They also took experts in various aspects of software development and asked how they did their jobs.

The discussion wrapped up with a general discussion about the fact that a lot of the process and project issues boil down to good management and the effect that bad management has on both the processes and the project. These problems are compounded by the fact that it is difficult to make organizational changes in the interest of improving processes. The last comment of the session was made by Dewayne Perry: process is not something that is needed just for developers, it is something that is needed for managers as well.