SESSION SUMMARY

Human Aspects of Process Design

Dewayne E. Perry

1. Session Introduction

Tetsuo Tamai introduced the session by defining process design as an activity that starts from a decision to have a process and that ends when the process is specified for use. One of the primary decisions to be made in designing a process is whether the process is to be a specific or a generic one. The output of the process design process is 1) a set of process definitions, 2) design principles and strategies, 3) an organizational structure, and 4) accumulated knowledge for decision making, etc. Secondary considerations in the design process are those of process reuse and education.

The main focus of this session's discussion is to consider the following three general issues:

- What is it that we design and how do we describe it?
- How do we make the process usable?
- What is the human role in the process?

In considering the human role in processes, we encounter some of the informal and non-technological aspects of software processes. Some of the typical roles are those of the process manager, the process engineer and the process user. The basic questions include:

- How do we tackle the problem arising from the human activities within the process?
- What are good strategies and techniques for designing software processes?
- How do we analyze human dominated processes and keep institutional knowledge or improve and innovate processes?

Other basic questions to consider are as follows:

- How general or specific (or, customized) should a process design be and to what extent can or should it be usable?
- How do you evaluate a process design?
- Is there a difference between process design strategies and software design strategies?

In addressing these issues and questions, there are several different approaches that might be taken: that of formal modeling with formal methods and formal languages; that of knowledge engineering; that of empirical studies; and that of sociological analysis.

The three panelists for this session were Watts Humphrey (providing an overview, emphasising process learning), Kathleen Culver-Lozo (providing the process engineer's viewpoint, emphasising process design for an organization), and Sam Redwine (providing the process user's viewpoint, emphasising making processes usable).

2. A View from Process Learning

Watts Humphrey proposed the following thesis: ultimately the productivity and quality of software organizations will depend on 1) the quality of the process used by the individual professionals, 2) the fidelity with which they follow this process, and 3) the effectiveness with which these individual processes are coordinated into projects.

Note that process improvement is itself a process: define, use, measure, analyze, and re-define. However, measurement and analysis will not change the process. Moreover, just ‘‘trying harder’’ is counterproductive. Hence, the issues in improvement are that 1) methods must be changed, and 2) the tools which we use must be improved.

There are three lessons concerning process learning: process learning is incremental (that is, we must use it, test it, and learn from it), learning is influenced by representation (there is no ‘‘best’’ answer; however, even manual representations are useful and simple representations are both possible and effective), and what we think is a useful process rarely is (there are always initial bugs and the process must change as we change).

The key issue is that of separating the people from the process. We want to do this because even the best people are error prone. Moreover, we must distinguish between talent (which varies significantly) and method (which should be uniformly consistent). The important thing is to learn from the variation between the defined process and the realized process.

Discussion

It was noted that there were three different uses of the term ‘‘learning’’ in Watts’ presentation: learning what the process was, learning properties about the process, and learning the underlying rationale for the process (in order, for example, to assign credit or blame, understand the capabilities of people, or to isolate the innate methods).

A number of interesting ways were mentioned for improving a process. One was to capture the intuitive process that talented people use. Another was to ‘‘learn from infidelity’’ — that is, to examine the work-arounds in the current process, or the deviations from the prescribed process. This latter approach was considered particularly important: it reflects the ‘‘creative flexibility’’ with which people follow the process.

There is a generic problem in getting people to use an orderly process. What are the motivational factors in getting people to follow the process? Deviations should be accompanied with feedback to the process designers. Additionally, there is a problem in getting the process users to see changes in the process as improvements. One useful approach to gaining acceptance is ownership where the people involved are party to the definition.

Another way of looking at the problem of process insertion is to consider the way in which it is introduced. If it comes from management (that is, top down), there is generally some skepticism about the underlying motivation. Is the introduction for the purposes of tracking and controlling people, or for purposes of improving the overall performance of the project? A bottom-up approach (that is, growing the process from those who will use it) provides warm, fuzzy feelings for the developers, but may be objected to by management.

In any case, processes are enacted by a variety of people: some are orderly and some are not; some are more talented that others. A ‘‘perfect’’ process will exploit the appropriate talents of the people in the process. This inherently limits the details of the process since people are not machines and should not be treated as if they were.

3. A View from Process Engineers

Kathleen Culver-Lozo defined a process engineer as one who is responsible for defining processes for an organization and helping the organization adopt the processes. The group that she is in at AT&T Bell Laboratories has been in existence for 1.5 years and has 7 process engineers. A list of sample activities was presented together with the process engineer’s worries for each item on the list.

process documentation	time
environment selection	consistency, feasibility
process evaluation	confidence
knowledge extraction	wrong job
technology assessments	missing something
resource allocation guidance	attention span
job explanation	resistance, misunderstanding
‘‘data’’ collection	job observation: big brother surveys: number abuse modification requests: responsiveness
process modifications	prioritizing

Process models have supported (to varying degrees) each of these activities. The important question is ‘‘how can they provide more support?’’ One solution is to use prototyping as a means of evaluation with an emphasis on the characteristics of the people for each task rather that on the details of each task (especially for tasks that they did not understand). This approach was taken because of the need to motivate the people in the organization. Currently, there are 20 people in the organization and there have been 150 process modification requests — obviously more changes than can be accomplished given the number of process engineers available.

Discussion

One can support activities with environments: tools are used for supporting developers, not replacing them. What the process-centered environment can do is to automate the routine stuff. This can be viewed as easing the burden and allowing for more creativity. The problem is that the routine activities are often embedded within more difficult and less routine activities. This makes the routine things hard to separate out.

While it is never wrong to ask what process is being used, there may be unforeseen consequences. For example, it was a mistake to replicate what an expert use-interface designer was doing. The process was ok, but the expectations were wrong; the process did not take into account the characteristics of the process users. In some sense, this intuition is counter to Humphrey’s previously mentioned suggestion: one should look to methods rather than talent.

It is also the case that things that are hard to extract and externalize are often the most helpful. The extraction process is made more complicated by the fact that people often say something different from what they actually do. For example, when asked, they often cite chapter and verse in the methodology; what you really want is what they actually do. Two ways of dealing with this problem are to have post mortem reviews and to monitor what is actually being done. The difficulty with the latter is that it is hard to distinguish what looks identical to the observer but is seen as different to the person in the process.

An interesting, but unresolved, question that came up in the discussion was ‘‘what is the difference between process engineering and software engineering?’’

4. A View from Process Users

Sam Redwine’s presentation was entitled ‘‘Developing Valid Processes for Line Project Engineers’’. The general problem is that we must aim for a valuable capability. In order to do that we must integrate capabilities that are already in use and codified with new capabilities that must first be externalized and completed. This integration then yields guidebooks, courses, videos, automation, marketing materials, etc. Sam’s primary thesis is that widely used products provide benefits.

A serious problem within corporate structures is that of validating various capabilities. Some possible validation mechanisms are:

- Technical advisory board
- Technical advisory group per capability
- Visits, surveys, existing documents
- Workshop, demonstrations, course feedback
- Reviews, consultants, ex-practitioner staff
- Pilot projects, consulting experiences
- Comparisons to the competition
- Validation laboratory

Sam then proposed a process development process as outlined below. It should be noted that processes are analogous to products in that they have a series of releases.

- Have an explicit capability and product requirements statement
- Start with existing material
- Try increasing the realistic situations
- Ask questions whenever something is not in the existing material
- Record questions and issues, and track the resolutions
- State and track the quality ‘‘metrics’’
- Produce guidebooks and examples
- Produce related products

In summary, process engineers must understand their customers and involve them throughout the process development process. It is desirable to deliver value early if possible. As with software products, one must plan for iteration in process development and meet quality and market windows. This entire enterprise is much easier if one builds on what exists that if one creates from nothing.

Discussion

It was noted that Sam did not talk about enaction support or automating that support. His response was that they were doing what they are good at: they are not good at software environments and have taken the strategic position to find vendors to use in conjunction with their work. In that, they have been reasonably successful: they provide a lot of support for the human enaction, but less on the machine side. There has been better luck in getting minor vendors to change their products that in getting major vendors to change theirs.

5. General Discussion

A major issue getting processes adopted is that of motivation: what are the motivational factors to get people to follow a process, note the infidelities, and to evaluate it? One solution is to build the motivation into the reward structure, but in such a way as to separate out performance assessment. Another suggestion was that it should be proactive rather than reactive. In some sense we can have both: interviews as well as a modification request process. Using a process must be a satisfactory experience in which credit is given for feedback on both conformance and infidelity. All this requires participation: in Japan, the Total Quality approach is well-known and works from the bottom up with suggestions for improvement coming from small groups.

Once a process has been institutionalized, there is the major problem of evolving and improving it. Typically, if the changes are small, then acceptance is easier to come by; if the changes are large, then they tend to be ignored. It is important that people need to feel that they are in control of the process and not vice versa. One approach is to sample users: use focus groups to monitor responses and results from process changes. Another approach is to start with a very high level description and slowly evolve it by adding project-dependent constraints. Particularly successful was a case where management imposed a process on themselves. Eventually the process migrated to the professional level.

There are a number of problems with regard to rewards within a defined process. Since one of the primary purposes of a defined process is to improve the general project quality and productivity, how do we balance the rewards to individuals with the rewards for team participation? How do we recognize the interdependencies and promote results that are useful to someone else? Are different kinds of individual motivation important factors in process selection? Clearly, some people are team motivated and some are not.

In current practice, projects are run by project managers who determine the process plan, control, and resources. Their skill is implicit and in their heads. How do we exploit it and support it with process? There general duties are plan, replan, monitor, and control. With process, we should be able to provide help for following the project plans — that is, do more or less what they do now, but supported by the process environment. Of critical importance at the project level are the organizational issues. While people are in general considered as identical and interchangeable cogs at the project level, they are radically different when considered at the individual level. Good managers can differentiate these two levels.

How do we organize and support the process so that it has a beneficial effect on individual programmers? There is some feeling that what we are doing does not improve their lot at all. It is helping someone, but not the individual programmer. One answer was proposed: the architects and requirements engineers now have a much clearer view of their process while the individual programmer may not. Another view was that we have assumed that the group is all that matters. However, there are two classes of motivation: group- and self-interest. Various hooks were suggested by which to motivate the individual programmer with regard to process: professionalism, making their job interesting (by removing or automating the burdensome details), providing a feeling of progress, and giving them a sense of protection from management.

At this point in the discussion it was noted that while we were supposed to be discussing human aspects, most of the time has been spent on motivation. Now that we have motivated them, how do we help them do what should be done? One important way is to provide knowledge of the options available at any particular point in the process. This may be of minor help to the knowledgeable. A second important way is to make known what the current state of affairs is. This is very hard. A third way is to provide reduction of effort. The supporting system can be aware and do more automation. For example, the system can provide documentation as a side effect. The same is true of measurement: it can be collected automatically because the process state is used internally. Another aspect of automation is of course tool invocation.

The design phase was suggested as a part of development where process might have a significant impact. It is a serious problem: either they don't know how to do it or they do not have a practice for doing it. Certainly for large systems, there is no orderly way of doing design and they end up design while coding. Hence, we will probably have the most leverage at this level. There are hard technical issues that process will help to organize and there are various levels of skills that it will help to support or manage. The problems in design often stem from incomplete thinking or from not remembering what had already been considered. What is needed is a non-overlapping way through the maze.

Reduction of effort is part of a general reduction in cognitive load. The more options, the more cognitive load. Given that we generally have not done it three times before, we usually struggle through. Hence, we need to help people understand the effects or results of options. In a way, it is like understanding the future state of the process. There is an enormous amount of information in software development and there is nothing to support capturing, ordering, analyzing, etc., the potential information.

The need to legislate part of the process is a tradeoff between locally optimal and globally optimal decisions. The two kinds of decision are almost never the same. In part this has to do with understanding the current state of the process. If one does not understand the vector of what is going on, it is all too easy to make the wrong local decision. That understanding is now handled informally in our development processes. One could define a process that makes that informal propagation of global understanding into a visible and tangible understanding.

What is different about talent in process programming and product programming? The output of the former is a product with a quality requirement not against the application domain but against the development domain. In specifying processes, the non-functional requirements may play a more important role in modeling the process than functional requirements. However, it is important to note that the metaphor is really not process programming, but process system development.

Process improvement is an incremental and evolutionary activity. It is based on the application of the familiar or what is known elsewhere. We need to be able to support the reuse of process components, to support descriptive modeling as a base, and to capture and record what actually happened (not only the actual measurements and artifacts, but the rationale as well).