

Product Line Development Experience I (Session 2 Summary)

Dewayne E. Perry

Software Production Research
Bell Laboratories
Murray Hill, NJ 07974

Wilhelm S. Schaefer

Computer Science Dept
University of Paderborn
D-33095 Paderborn

Lawrence G. Votta

Software Production Research
Bell Laboratories
Naperville IL 60566

Votta introduced the session by delineating three components of the software development enterprise (process, technology and organization) and posing the following questions. How do organizations affect process and product reuse? How do we characterize or talk about organizations? What are the limits?

Nancy Staudenmayer provided the keynote talk comparing the product line development efforts of Lucent Technologies and Microsoft from an organizational theory standpoint. The case comparison emphasized two aspects of product line development in these two organizations: the implications of heterogeneity and the implications of change. Staudenmayer takes a midway position between the macroscopic view of organizations (looking for patterns of behavior) and the microscopic view (looking at individual and psychological factors).

She outlined a conceptual model as consisting of a context (resources, environment and history), a strategy or vision, throughputs (human resources, tasks, culture and formal organization), and outputs (individual performances, project performances, organizational performances such as profits).

Staudenmayer's strategy was to do inductive, comparative studies. For the Lucent side, she looked at two systems (implementing three products) with about 25 developers and 100 support people. For the Microsoft side, she looked three projects (Windows, NT and Office) with about 100 to 200 people. Data was collected over a period of three to six months of observation using over 40 interviews, access to project and process documentation and an ongoing log tracking larger technological issues.

Given that the software organizations are so different, the research question to address is "what are the implications of heterogeneity for product and process reuse?" To answer this question, Staudenmayer claimed that the organizational structure is fundamen-

tal and thus looked at three projects from each organization and compared these projects both within each organization and across them. Given that there is a lot of variability and that she started each study in different places in their process, it was important to control for size and legacy. It proved to be difficult to find PC products comparable to the Lucent ones. In both cases, she tried to select those where interdependencies were getting harder.

The initial state for each company was characterized as follows.

- Key Products
 - Lucent: switches
 - Microsoft: tools, languages, OS in PC
- Customer Industry
 - Lucent: captive supplier and not much rivalry
 - Microsoft: individuals and companies, intense rivalry
- Dimensions of Merit
 - Lucent: robustness, reliability
 - Microsoft: neat features as fast as possible
- Corporate Strategy
 - Lucent: vertically integrated, global expansion, cost efficiency
 - Microsoft: independent groups, speed for global markets
- Human Resource profile
 - Lucent: experienced, educated, mid-west
 - Microsoft: young, hackers
- Culture
 - Lucent: engineering, loyalty
 - Microsoft: aggressive, hacker
- Organizational Structure

- Lucent: hierarchical, large departments which mirror product architecture
- Microsoft: organic, small teams using synch and stabilize approach

Both companies had competition and both were “winners”. Microsoft’s “synch and stabilize” strategy began in Excel but did not become widespread in Microsoft until the 90s.

Alex Wolf raised the critical issue of building a piece of a system versus building an individual product. In Lucent, the organizational structure is very hierarchical with many layers of management; in Microsoft, it is a flatter structure. Tully asked whether the hierarchical structure might not be the result of a long organizational legacy. Business process improvement (BPI) approaches have been trying to escape from this form of functional structure.

Perry raised the issue echoing Alex Wolf’s point above as to whether you could apply Microsoft’s structure to Lucent where extremely large-scale software systems are produced. Staudenmayer’s response was no, each organization put together the right structures at the time.

The matter of formal versus informal structure was raised by Groenewegen where Microsoft’s organizational structure was much more informal. Wolf and Tully both noted that there is another important point within even formal structures: that of informal lines of communication. Both of these are important.

Balzer asked whether, given the degree to which a product structure is reflected in the organizational structure, is Microsoft being forced into a hierarchical structure? Staudenmayer characterized the synch and stabilize approach as development, testing and integration as done in parallel, sometimes with pre-defined interfaces, sometimes not.

The implications of organizational structure of Lucent for product and process reuse are as follows: reuse offered significant benefits in terms of reduced costs and improved quality. These benefits were supported by stability in the organizational structure, an open and adhered to technical standard, and infrequent releases of the product. For Microsoft, the benefits were not so clear: there has been massive change in the products and multiple products were shipped on different schedules. Whereas speed was critical, reuse takes time to organize and a clear vision of what to reuse is needed.

With Lucent, initially the product was a homogeneous one and the reuse of both hardware and software was standard. With the advent of international markets, the problem of customization has become in-

creasingly important resulting in a base system that is used to support the customization. Osterweil pointed out that quality was the primary motivation and that that is a fundamental distinguishing characteristic between Lucent and Microsoft with reliability as the basic quality driver. Lehman added that the early systems were pretty much telephony and not much else. More recently applications are getting more mixed in. This causes organizational problems.

Osterweil disagreed that reuse necessarily takes time — look for example at chip reuse. Staudenmayer replied that Microsoft did not understand the basic pieces to use and reuse. Boehm pointed out that in reuse experience at HP, two projects took it on the chin before the benefits of reuse became reality. No one at Microsoft wanted to do that. Balzer suggested that the driver is unpredictability: you get the benefits of reuse from narrowing, and they did not know what to narrow to.

For both companies, the critical driving factors are changing over time. For Lucent, time to market is becoming critical. For Microsoft, customers are demanding more reliability and quality, and are less concerned with time factors. In both cases, the technological and market conditions are dynamic. In Lucent the emphasis historically has been on quality; they are now trying to increase their speed to market. Perry noted that we don’t know how to make the tradeoffs between quality and cost. We don’t know how the knob works. Quality tends upward but we don’t know how to diminish quality two units and get two units decrease in interval.

Microsoft’s initial strategy was independent products with different delivery dates. Office then collected the components together, integrated and shipped them. This provided a simple structure. The more recent strategy is to ship Office first and then the individual applications. For this, they now need a coordinated shipping. Given this coordinated effort there is also the need for sharing across applications — for example, identical tool bars. Where there were individual teams, there is a now also a core component team. Now, much like the Lucent, the integrated product has massive interdependencies and tremendous compilation problems.

Thus, as Balzer noted, they have increased interdependencies but not the structure that matches it. However, the strategy is features and if you don’t make the features you can still ship. Note that the common features are critical and have to be agreed on. It also has to have the features it has had in the past. In addition you have to make sure that the individual parts

don't get too ambitious.

The key changes in Microsoft technically are that its products are increasingly large and complex and much more system-like. From a marketing standpoint, there is feature creep, compounded by the requirements for commonality, consistency and robustness. Microsoft is beginning to suffer from product saturation.

Lucent's initial strategy consisted of shipping the product once a year with improvements and new features, carefully controlling the dependencies. The more recent strategy calls for componentization and customization. The market is customer-demand driven. This approach is complicated because of the subsystem architecture (that is, it is not flexible enough).

Essentially, the two companies are being driven towards each other by the market and technological forces. Lehman noted that this was nothing new: Microsoft mirrors OS360 and IBM of the 60s.

Staudenmayer's conclusion was that the two software organizations differed markedly, with the motivation for reuse different in each organization. Osterweil claimed that in both cases, there was an insufficient grasp of who they were and hence they could not predict where to go. A better understanding would lead to a better process and to a better adaptation. Staudenmayer countered that both companies were aware and knew what they did. Wolf noted that processes are tied to the products and both need to change the products, but can't change their processes. Balzer again brought up the issue of predictability: they have something new and they cannot predict what to change, they do not have a mechanism to reason about it.