Software Processes and Business Processes

Ian Thomas
Mountain View CA USA

This panel paper addresses two issues: the position of an organization's software process with respect to its other business processes, and a comparison of characteristics for software and business process support mechanisms.

Although many modern organizations are software–dependent, this does not mean that software development is a necessarily a critical business process for them. For many leading companies, software products, or services built on software capabilities, provide the organization's competitive advantage. Many of these companies develop some of their own software systems in order to retain their software–dependent competitive advantage. Some have isolated functions of their software systems that do not provide competitive advantage and have bought off–the–shelf systems to support those functions.

The discussions on whether to invest in the software process is a legitimate business question for companies. A recent "Business Week" article [1] pointed out that investments in Quality programs need to be justified, as do other company investments. These discussions are resemble the 1989 "Computerless Computer Company" debate in Harvard Business Review for semiconductor companies and fabs. The initial position in the debate was that most of most of the added–value provided by semiconductor companies comes from their designs, rather than the execution of the manufacturing process in a fab owned by the company. The original article was followed by heated rebuttals from semiconductor manufacturers.

Our conclusion is that, even though it may be a process that a company carries out, software development may not be a critical process for the company.

As for common characteristics of mechanisms for software and business process support, our limited experience is that there is a high degree of commonality. We need to be careful when we make the comparison that we compare like with like. We should not compare complex software processes (involving multi–person coordination over different data sets) to simple sequential business processing of a single form. We need to exercise similar care in comparing the software process with manufacturing processes.

We have recently been involved in the definition and modeling of a sales process. A sales cycle within this process typically extends over two or more years, with several intermediate milestones before the final deal is signed. An analog in the software development world would be requirements negotiation and prototype construction for an important contract.

The sales process we modeled is one of a class of "informal" processes that are critical to successful business and software development. We have no precise definition of an "informal" process, but believe that one of its characteristics is a large number of exceptions to "normal" processing, both in the process model itself and its enactments. A second characteristic (that may be a corollary of the first) is that process performance is "opportunistic", where many actions are initiated by the state of the sales process and the sales cycle information, rather than the completion of previous actions.

The sales process also involves lots of interaction with the client, just as prototyping and iterative development would for software development.

In addition to this example, we see that recent work on mass customization in the business world [2] is using many of the same concepts that are used in software process discussions. Mass customization is achieved by assembly of a process from constituent sub–processes.

These indications, and other experience, lead us to believe that there are strong similarities between the mechanisms necessary for software and business process enactment.

[1] *Return on Quality*, Business Week, August 8th 1994.

[2] B. J. Pine II, B. Victor, A. C. Boynton, *Making Mass Customization Work*, Harvard Business Review, Sept – Oct 1993.

0-8186-6695-1/94 $4.00 © 1994 IEEE