# The Tale of Two Projects — Abstract

Dewayne E. Perry[1] and Lawrence G. Votta[2]

[1] Bell Laboratories, Murray Hill, NJ 07974, USA
[2] Bell Laboratories, Naperville IL 60566, USA

We have two hypotheses that we want to demonstrate in this study: organization is not independent of process, and process is not independent of technology. Clearly, one can imagine cases where organizational issues can be separated from process issues. Similarly, there are levels of abstraction where aspects of process are independent of particular technologies. In general, the three should be considered as interrelated and interdependent. We show the validity of our claims in the discussions of the two case studies below.

We have selected these two projects for the following reasons: they have executed a complete cycle of development, they have well-documented post mortems of their experience, and they have the requisite quantitative data. We measure product development projects using three interrelated, macroscopic variables: cost, quality, and time interval. In our analysis we focus primarily on time interval, with some consideration given to quality, and emphasize the contrast between the prevailing process, organizational and technological development structure and the one implemented in each of the case studies.

The Y0 Packet Features Development is composed of four features with a total code size is 54.6 thousand noncommentary source lines (KNSCL), a total staff size of 39.8 and a total number of faults at delivery to the first customer of 12. The goal was to reduce the development interval from 16 to 12 months, while maintaining or slightly improving the quality of the product. The fault density, as delivered to the first customer, is the measure of the product quality.

The standard development [3] results from an assembly line like approach to developing software. At each stage a major milestone is defined. Throughput was increased by making one organization responsible for each stage, but at the cost of many handoffs which are costly in time and quality, and are difficult to coordinate. The unit of planning is one month and is the result of balancing the need for tight coordination required to control interval with many organizational handoffs and the cost of tracking the process.

The Y0 development process altered these factors. Instead of a functional organization approach, a team approach was used to minimize handoffs. This organizational approach minimizes the number of formal reviews between functional teams and thus reduces the time interval. The milestones were more naturally matched to the structure of the Y0 features and the team's talents. This allowed the team to exploit characteristics of the problem making the entire development less prone to fault insertion.

[3] Joseph S. Colson, Jr., and Edward M. Prell.: Total Quality Management for a Large Software Project. AT&T Technical Journal 71:3 (May/June 1992) 48-56

The FNMS-R3 software development [4] was an enhanced release of 45 KNCSL of C++ on a base of 140KNCSL undertaken by about 25 people and consisted of three major features and a number of minor features. The previous release (FNMS-R2) took about 16 months to complete. The process was too unresponsive to customer needs and the products were too unstable in the field.

The development schedule was mapped out and used as the management plan directing the development process. The general intent was to support incremental development. Except for a one day high level design review and an external architecture review of the FNMS-R1 architecture, there were no design reviews or code inspections. Moreover, there was no formal unit testing and only minimal integration testing (with no clear exit criteria). Documentation was done after the fact — while the product was being soaked at a field site.

The organization was structured functionally: systems engineering, development, and system test. Problems that arose from the separation of these functions included: interface problems; lack of support in reviewing requirements in a timely manner; an inactive MR review board; and status meetings which were reduced to fighting fires and managing crises.

The FNMS-R3 development process altered four things in order to achieve their goals. 1) They added some standard quality gate techniques. 2) They decreased interval time by decoupling features that could be developed in parallel. 3) They changed from a functional to an interdisciplinary team organization. The teams were empowered to be responsible for their features from feature specification through integration. 4) Within the individual feature developments, team members were encouraged to do as much in parallel as possible.

The results of these changes were as follows. 1) The cycle time was reduced by about 25% to 12 months. 2) Decoupling features enabled short features to be implemented and delivered very quickly. One of the major features was delivered three months ahead of the other two features. 3) Defects were removed earlier with very few problems encountered after integration testing. 4) The team organization increased the effectiveness of the development process with team members assuming various roles that were previously in different functions. Moreover, the team approach significantly increased the effectiveness.

Thus, both the development interval and the product quality were increased by effectively exploiting the structure of the organization and the technology of the product, and introducing sound software engineering techniques.

Both projects display the same strong trends even though they were done in very different parts of the business, in different kinds of software developments, and in different geographic locations. Both projects had an organizational structure of strongly empowered teams, understood the technology at a fundamental level, and used some innovative software engineering technology. Without these organizational structures and the technology exploitations, neither of these projects would have achieved the resulting level of success.

[4] H. T. Yeh.: Re-Engineering a Software Development Process for Fast Delivery - Approach & Experiences. Proceedings of the First International Conference on the Software Process (Redondo Beach, CA, October 1991) 106-112

This article was processed using the LaTeX macro package with LLNCS style