From Technology Migration to Organizational Culture Change

Uolevi Nikula[1], Christian Jurvanen[2], Olly Gotel[3], and Donald C Gause[4]

[1]Lappeenranta University of Technology, Lappeenranta, Finland
uolevi.nikula@lut.fi
[2]Agricultural Data Processing Centre Ltd., Finland
christian.jurvanen@mloy.fi
[3]Pace University, New York City, New York, USA
ogotel@pace.edu
[4]Binghamton University, Binghamton, New York, USA
dgause@stny.rr.com

Abstract. Organizational changes are often difficult to implement. However, neglecting organizational development does not stop technological development from taking place, and the consequent impact on the organization may give entrant organizations an attacker's advantage over those organizations that fail to develop themselves in any way. Since business organizations are complex systems in which most elements affect many other elements within the organization, it behooves us to develop a better understanding of the dynamics of organizational change. In this paper, we present a case study of technology migration followed by process improvement actions and larger organizational changes, activities that span a ten-year period in the life of a single organization. The migration project was followed by a sustained process improvement phase in which organizational changes were implemented in a structured way, thus enhancing the visibility of the change process itself within the organizational culture. The paper includes a discussion of the most important organizational changes that occurred during the ten-year period, and important observations and critical issues are summarized during the organization's evolution into a *learning* software organization.

Keywords: Organizational change, cultural change, systems thinking, process improvement, technology migration, learning organization, organizational innovation.

1 Introduction

Introducing organizational change is seldom easy [1, 2]. In many cases, established organizations become fixed to their prevailing situation to the extent that justifying a change may be more challenging than actually implementing the proposed change. The standard response to suggested changes often follows the line "the way we do things around here" referring to the rites and rituals of the organization [3]. In extreme

cases, the change resistance may even risk the entire existence of an organization by giving competitive new organizations an "attacker's advantage" [4].

The five key elements of *the learning organization* have been claimed to be systems thinking, personal mastery, mental models, building shared vision, and team learning, which together form essential management tools and perspectives for organizing, monitoring and managing the learning system [5]. As a consequence of modeling and managing today's business systems from the perspective of the complex system, the introduction of change in one element or subsystem is most likely to affect all parts of the system and nearly all system elements. Thus getting any important change accepted in an organization can lead to an easier acceptance of other consequential changes. In the present paper, experiences of a technology migration project from a mainframe computer to a network server system are reported as this change operated also as a stepping stone for the introduction of process improvement, ultimately leading to a major cultural change in the organization concerned. The system migration project took over five years to complete, but for the purpose of this paper, we also take the following four and a half years of organizational development into account so as to get a better understanding of the cultural change that occurred.

The study was conducted based on an analysis of the organization's annual reports for 1998 through 2007 and of its internal development documentation of major events during 1999 to 2008. The annual reports were studied to find key information on the technology migration project and other organizational development events. More detailed information on the organizational development was received from ADPC's Corporate Vice President, who is also a coauthor of this paper. The collected material focuses on factual information (e.g., actual events, dates, and number of employees) and, since the annual reports have been accepted by the organization's board members and distributed widely to its owners and customers, the accuracy and credibility of the study material is considered high.

The remainder of the paper is structured as follows. Section 2 reviews research related to this topic and Section 3 presents the case study description. The relevance of the findings for learning software organizations is discussed in Section 4 and the conclusion of the paper is provided in Section 5.

2 Related Research

This section summarizes the related research on organizational structure, technological development, and organizational change. For the purpose of this paper the term *technology* means the processes by which an organization transforms labor, capital, materials, and information into products and services of greater value, including also managerial processes [6].

The role of the organization has generally received limited interest in the software process improvement literature. One exception is the IDEAL process improvement model which describes the organizational elements that are generally needed to support process improvement (i.e., software engineering process group, management steering group, and technical working group) [7]. Zahran [8] also provides a broad

discussion on software process infrastructure that consists of *corporate process culture* and *software process infrastructure*, from which the software process infrastructure is further divided into *organizational and management infrastructures* and *technical infrastructure*.

The technology S-curve has been developed to depict the magnitude of performance improvement that can be expected of the effort put into improving a technology (Fig. 1) [4, 9]. Generally, the performance increase is slow with a new technology before the essential knowledge of the technology is learnt. After that, the improvement rate increases rapidly until the technology matures, and finally the curve flattens out as the technology approaches asymptotically a natural or physical limit. At this point, only marginal improvements can be gained irrespective of the effort put into technology improvement. Fig. 1 indicates the turning point on the S-curve since passing this point moves the technology to the "management comfort zone" suggesting that the technology has now proven its capabilities and is performing as desired [10]. However, since technology matures before sales and profits, not investing in a new technology after the turning point would be a real risk [10]. The "management discomfort zone", on the other hand, refers to the time before the turning point of the S-curve and means that the technology is too immature for research and development to provide the business with reliable information about its potential and probability of success [10]. Even though the technology S-curve is a very simple model, it has been shown to depict the performance increase gained from various technologies in retrospect (cf. [4, 10]). Thus it can help technology users to both understand and act appropriately at different phases of a technology's lifecycle.

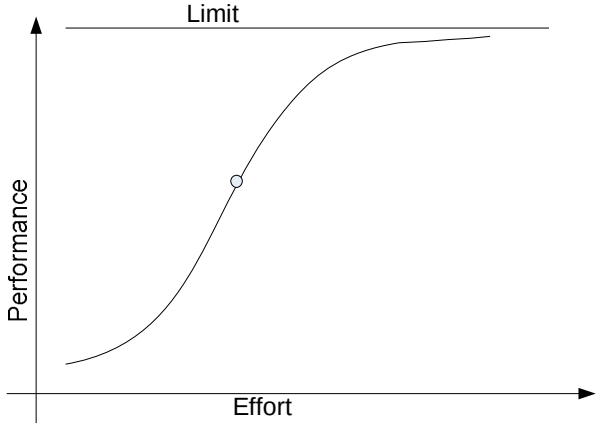

Fig. 1. The technology S-curve with limit and turning point [4].

The flattening of a technology S-curve means that to maintain the achieved improvement rate, a switch to a new technology and a new technology S-curve are needed [4]. Examples of such technology switches are frequent today as illustrated by, for example, chemical photography being replaced by digital photography, cathode ray tube televisions and computer displays being replaced by liquid crystal displays, analog televisions yielding to digital televisions, and light emitting diodes replacing the filament bulb. The transition from one technology to another often entails a discontinuity, as shown in Fig. 2. Managing such technological

discontinuities has been found problematic and it has been estimated that leadership changes hands in about seven out of ten cases when discontinuities strike [4]. Overall, only one third of such major technological changes in organizations typically succeed, mainly due to change resistance [11].

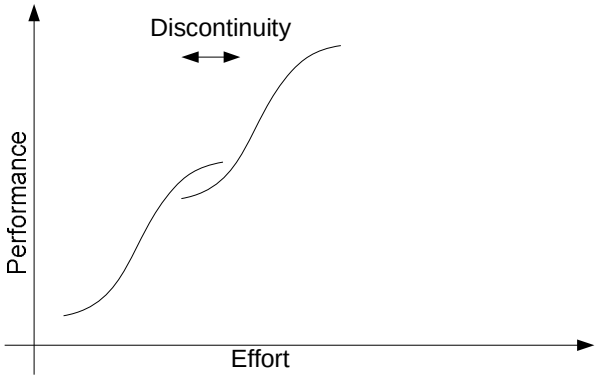

Fig. 2. Two technology S-curves with a discontinuity [4].

Organizational change has been studied since the early 1950s when change was first characterized as a three-step process involving unfreezing, moving, and refreezing phases [12]. Numerous change models have since been proposed in a variety of disciplines, ranging from personal grief [13] to workplace morale [14]. Irrespective of the labels and number of phases of positive and negative growth, all these change models tend to follow the general pattern of some transition period falling between two states of initial and final equilibrium [15], as identified by Lewin [12]. The "Classic Change Curve" is commonly used to portray the decrease in organizational performance that occurs during the transition period of change (Fig. 3). In a typical change project of this nature, the performance at the final equilibrium state may differ little from that at the initial equilibrium state, but improved performance is expected in an effective change program. What characterizes this Classic Change Curve in particular is a single drop in performance during Lewin's intermediary transition period, a performance dip.

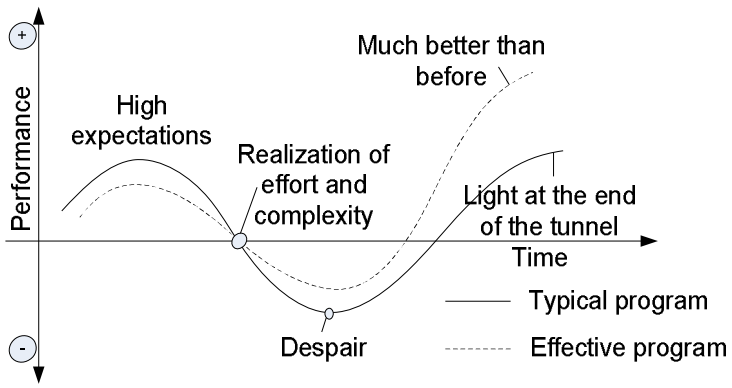

Fig. 3. The Classic Change Curves for typical and effective programs [16].

Even though the Classic Change Curve has intuitive appeal and is supported by anecdotal evidence, limited empirical data has been reported to substantiate and explain it. Elrod and Tippett present one of the few empirical studies that demonstrates that the performance dip associated with organizational change is real and measurable [17]. Elrod and Tippett [15] conclude that people generally do not anticipate change and tend to resist it, thus causing the dip in the change curve. Also, when a change project is initiated, high expectations prevail, but there is an inevitable degradation of performance as employees learn new ways of working and implement the actions needed to undertake the change. The actual scope of the required effort and complexity of the work is often only realized after the project has started, ultimately deepening the performance dip to the point that it is often called the "death valley" of change [15]. After the necessary learning and change implementation activities have been completed, the organizational performance should start to recover.

3 Case Study – Agricultural Data Processing Center (ADPC)

3.1 Background

The origins of Agricultural Data Processing Centre Ltd. date back to 1958 when the first computers were sold in Finland and the National Insemination Association acquired one of them. The Statistical Department of the Association started growing and it acquired its first stored memory computer in 1965, an IBM 1401, followed by a 360/30 in 1968. The growth of the Statistical Department continued until it became an autonomous, self-contained organization in 1986, thereafter known as ADPC. By 1998, ADPC was a medium-sized software house with 44 people providing IT services and software solutions in the agricultural domain. ADPC was organized around the four main functions of the organization's business – Software Development, System Services, Customer Service, and Administration. The Software Development function included 20 people who were primarily responsible for the development and maintenance of software, while the System Services function ran the organization's services with eight people. The Customer Service function was composed of nine advisors who entered the non-electronic customer announcements into a database and answered customer enquiries based on the data, while the Administration function comprised seven people. Seven associations and organizations working in the agricultural domain owned ADPC. Since all the owners were also customers, ADPC was operating in a co-operative manner, meaning that it was not expected to maximize the profit to the owners, but rather to service all the customers profitably.

The ADPC customers also included individual farmers and public administration. The primary production registers covered basic information for all bovine, pigs, and lambs in Finland, and over 12,000 farmers used them. The institutional customers included breeding and dairy specialists and secondary production like slaughterhouses that had to check the complete history of every animal before slaughter. Since the

beginning of 1998, the European Union has required that each member nation maintain a national cattle register to be able to identify, register, and track all the major events of all bovine animals. In Finland, such a registry was developed by ADPC and put into use in 1995. Due to numerous customers with varying needs the registry, just like all the other ADPC systems, was developed to be available on a 24/7 basis. All the data inserted in the system since the mid-1960s was stored in the same database and the principal way to develop applications was to devise a program that accessed and manipulated the relevant parts of the database.

In the 1990s, ADPC was faced with increasing requests for new functionality such as Internet connectivity and usability improvements to provide customized reports and standardized user interfaces across applications. However, implementing such changes in the existing system had proven difficult and laborious. Most of the software development was done by individual people who managed the tasks on their own from the initial customer contact to customer delivery with minimal management interaction. There were no development standards and the system documentation was minimal to the point that the software development was very person dependent and running a project with more than one developer was difficult. Attempts to recruit new people had failed as applicants did not know the mainframe environment and the High Level Assembler HLASM macro assembler language used with it, and were not willing to learn them either. The attempts to outsource development tasks suffered from similar problems. Thus the perceived limitations of the system and development process, combined with the overall high operating costs and imminent need to renew the mainframe licenses, created the impetus to reconsider the future development platform at ADPC. A migration from the mainframe to a network server based system was estimated to take about three years and require an effort of between 20 to 25 person years, so ADPC made a decision to start a migration project from the then current IBM 9021 compatible mainframe to a Microsoft NT based network server system at the end of 1998. The migration project had two explicit goals: (1) to establish a new technical data processing environment that was simpler and more cost efficient than the current system; and (2) to establish new development practices that were simpler, more cost efficient, and would more readily enable application for an ISO-quality certification.

3.2 Migration Project

The migration project started at the end of 1998 and the end of 2001 was the estimated completion date. However, the project was not completed on time and another 15 months was needed. Even after the system switchover, a further year was needed to assure that all the applications worked as expected. Thus the migration project continued until May 2004, after which ADPC continued implementing process improvement actions but in a better planned and more managed way.

3.2.1 Original Project Timeframe

The migration project began by organizing the software development based upon the technologies used within ADPC, the Windows NT, statistical development with the SAS software package, and the IBM mainframe groups. The management was also

strengthened at this point in time by hiring a Senior Management Consultant to support the project steering group and four agricultural specialists as Project Managers for the migration project. The evident changes in required competencies were addressed by organizing a training program on management practices along with the newly required supporting tools and techniques. The Windows platform made outsourcing also possible, and the first nearshoring contract was established with a Russian company to develop the server-client communication system in 1998, followed by a contract with Finnish database specialists in 1999. Even with ten external consultants working on the migration project at its peak, most of the development was conducted in-house. Twelve new developers were hired in 2000 and seven in 2001, which increased the employee count by 25% and 12% respectively.

The migration project was started with Microsoft NT as the target platform but, with the introduction of Windows 2000 in mid-1999, the migration target was changed to Windows 2000. At this same time, the target technology changed from the Component Object Model (COM) technology to COM+ technology. Due to these changes occurring in an early phase of the project this did not result in the need for much extra effort, but towards the middle of 2000 the quality of the new system became a concern. At this point in time, the NT Development Manager resigned and a newly hired developer started as the Software Manager in the beginning of August. The quality issue was tackled by creating a test competence center that was responsible for acquiring testing tools, methods, and skills in the organization. The increasing testing effort was addressed by investing in test automation and the first test automation tools were acquired in the spring of 2001. The test practice improvement began with consultant support and in the fall of 2001 testing became an integral part of the software development process. However, along with active integration and system testing it soon became evident that the developers could not provide the expected test quality, so a separate Testing Unit was established with an externally recruited manager in October 2001.

The migration project was not completed to the original planned schedule by the end of 2001 even though important milestones were achieved. The system was developed on top of a single database and TCP/IP-based communication system and the overall system architecture was described to the owners and customers of the organization in the annual report for year 2000. The software development followed a basic process including design, development, and testing phases according to internally developed coding standards. In the annual report for 2001, the Managing Director described the project by noting that the employees were competent with the technologies being applied, the working practices and standards were at a good level, all the technological surprises were addressed, and the project just needed more work to be completed. However, about 40 person years had been expended on the project by the end of 2001, exceeding the original estimate by 100%. Due to ADPC resource constraints some customers had started doing the software specifications on their own and some other customers sought other suppliers to satisfy their needs.

3.2.2 Extended Project Timeframe

The migration project was not completed by June 2002 and, since there were no clear indications of a date for project closure, the Managing Director who had initiated the project resigned. The Vice President was promoted to Managing Director and the

Software Manager was promoted to Vice President. In addition to these internal promotions, a new full-time Senior Project Manager was hired to coordinate the migration project activities and to replace the previous Senior Management Consultant. The hiring of employees continued in early 2002 and the employee count increased by 13% in 2002; in the meantime, the use of external consultants was being reduced.

The new system was put into operation on March 14th, 2003, one week after switching off the old system. However, the work with the new system continued and the migration was only fully completed in May 2004 for two reasons. First, the migrated system included numerous applications that were used only periodically, and a year was needed to test them all in practice and to get the users to accept them. Second, the migration project was initially completed with the Windows 2000 tools, but Microsoft had released two versions of the development environment during the migration and so a half-year system revision was done after the migration project to upgrade to Visual Studio 2003, SQL Server 2000, the .NET 1.1 framework, and the C# programming language. The existing applications were tested to work in this environment while all new development was done with the new tools supported by a company-specific .NET application framework developed in parallel with the system revision. By the end of 2003, the system had 11,000 users and the system reached a total of 1 million logins. The system had also proven stable as unplanned interruptions had taken up only 0.5% of the 24/7 operation during the first nine months.

Meanwhile, the migration project had driven ADPC into financial straits requiring organizational actions. In the fall of 2003, all the major outsourcing contracts were terminated and the employee count was cut by 18%. The project organization structure was replaced by a line organization comprised of four departments: Software Development, Customer Services and Relations, Development Projects, and System Services. The Software Development department included the Development and Testing Units while the Customer Services department was formed from the application domain specialists hired as Project Managers for the migration project and now titled as *Key Account Managers*. The Development Projects department was responsible for both large customer projects and internal process development work while the System Services department operated the production system. At the same time as these structural changes, management changes took place as the Software Manager was promoted to the Department Manager for Software Development and to the Corporate Vice President. Since the Test Manager left ADPC at this point, both the Development and Test Units got new managers through internal promotions. The migration Project Manager remained in ADPC as the head of personnel.

The development of practices focused on finalizing and documenting the initial efforts while the training emphasis moved to internal training. The testing was strengthened by acquiring a load testing tool and standard test practices were established before the system switchover. The switchover was followed by the release of the first software development process in the organization, called the “Task Model” (TM). Since 75% of the software development at ADPC comprised small software development projects, enhancements, or maintenance work typically requiring one to eight person weeks of work, the TM process was developed to fit such activities and to reduce the associated project management overhead. The idea was to write all the information related to a task in a task description, a ticket, and to pass it through the

development process along with the actual code. The focus of training moved from externally acquired courses to monthly internal training sessions and utilizing materials available from tool vendors. The training needs were most apparent among the developers due to the fast development of the tools and technologies. The Test Unit adopted similar practices. The employee responsibilities were also formalized after completing the migration project by written job descriptions.

From the project management point of view the extended project time was a challenge. The recorded work efforts indicated that in 2002 35 person years (PYs) were invested in the migration project, in 2003 another 30 PYs, and in 2004 a further 20 PYs. The migration project required, in total, approximately 65 months and 120 PYs to complete. Thus the original estimate of 20 PYs and 36 months was exceeded by a fair margin. However, at this point ADPC had tangible improvements to show for the investment: the technical infrastructure had been completely changed from a mainframe to a network server based solution; the database, including data from the past 40 years, had been converted into a standard relational database format; and systematic practices had been introduced in the organization, supported by tools, processes, techniques, and training. The organizational structure and management practices had also been brought up to date with the new technology and development practices.

3.3 Continuous Software Process Improvement

After completing the migration project, ADPC focused on modular process improvement, but they could not avoid architectural changes entirely. Namely, the design process improvement efforts were problematic to implement in part and their resolution was not possible without organizational and management changes. In the line organization introduced in 2003 the process improvement work was given to the Project Department with few concrete changes. In January 2007, a Research and Development department was established with six designers from different departments, and it was given the responsibility of the design process. The first design process was released in March 2007 and it covered both requirements and design processes. However, the process was not adopted in use and, after a thorough problem analysis, the Design Unit was moved to the Software Development department in March 2008 with its own unit manager. The new Design Manager had managed the ADPC Test Unit for the past four years and so started by creating a development plan for the unit. By the end of 2008, the Design Unit improvement was progressing along the published development plan. An initial state assessment was conducted in May [18], training sessions were arranged in June and December, and a requirements management tool evaluation was conducted in the fall 2008.

Other organizational actions focused on individual competence and motivation issues as well as reinforcing collaboration across ADPC. During the extended migration project a coordination group had been established to manage development efforts in different departments but this group was dissolved after the project. However, the group was reestablished only a year later due to the need to coordinate internal developments. As the migration project upheavals started to level out, work assignments became less challenging, in turn causing motivation problems. This issue

was first tackled by instigating training programs that would lead to certification. By the end of 2008, all the testers and half of the developers had acquired certifications, and ADPC had achieved a Microsoft Gold Certified Partner status [19]. To further emphasize the role of competences, the development of a formal competence evaluation and improvement scheme was initiated at the end of 2008.

The changes in the software development practices were aimed at improved efficiency, the removal of observed constraints utilizing process performance data, and keeping up with the technological developments. The adopted process model, the Task Model, had problems with keeping the code and task information together so a test management tool was adapted to track the tasks through the process in October 2005. New tasks were now inserted in the tracking system with estimated times for each phase. Developers could then pick a suitable task from the developer task queue to work on and as soon as the task was completed its status was changed to "waiting for testing" so that a tester could continue working on it. The automated system started working well and, after the release of the design process, the whole software development process was covered by the tool supported Task Model by April 2008. The data from the system demonstrates how the handling of tasks changed after a measurement program was introduced: the number of tasks completed on schedule increased from 31% in 2005 to 78% in 2008 and, in the same period, the average development time decreased from 76 days to 23 days, while the number of tasks increased from 88 to 246. Even if the effort required by each task was not recorded, the task size had clearly reduced after the tool adoption.

The technological development also kept the ADPC software developers busy. The ADPC development environment, Microsoft Visual Studio, was upgraded to version 2005 and .NET 2.0 in fall 2005 and to version 2008 and .NET 3.5 in spring 2008. By the end of year 2008 about 70% of the development work was done with Visual Studio 2008 tools and only 30% was done with the tools the migration project was initially completed with (e.g., Visual Studio 6, Visual Basic, and ASP.1). The first upgrade took nine months while the second was completed in five months, alongside regular development projects and without major issues. The increasing role of agile software development was also observed at ADPC and a small project model with sprints [20] was tested in 2008. However, the Task Model was used with about 75% of the work, and most effort was put into increasing the efficiency and predictability of this model. For estimation purposes, function point analysis [21] was piloted in 2008, and a project estimation tool was developed to calculate the likelihood of completing a task in the estimated time [22]. Both efforts had promising results and plans on extending the experiments were made.

The completion of the migration project moved the development focus to new applications and 75% of all the development effort went into new applications in 2004. The developed application framework was utilized in about 90% of the applications developed in 2008, but the viability of having only one dominant design for all the applications was also raised. Along with the stabilizing system and practices new customers were also acquired and the biggest customer project ever for ADPC was started in the fall of 2008. The 3.5-year project was, budget-wise, about four times the size of regular customer projects for ADPC. To be able to handle such a project, along with increasing numbers of ongoing and regular customer projects, ADPC started looking at outsourcing and nearshoring options again.

The time after the migration project provided a very welcome change to ADPC financially. The biggest factor contributing to the improved financial result in 2004 was the cost reductions implemented after the switchover of technology including the termination of the outsourcing contracts, reduced hardware and software costs, and reduction in personnel. The improvement trend continued and ADPC's self-sufficiency reached a level greater than prior to the migration project in 2008. Only in the peak of the migration project did ADPC have more employees than at the end of 2008: 72 people, from which 6 were designers, 15 developers, and 8 testers.

4 Discussion

4.1 Organizational Changes

This case study demonstrates how an organization initiated far-reaching changes with a technology change. Even though the previously used technology was still working, its fragility hampered process improvement actions. One way to overcome this kind of change resistance is the deliberate introduction of new technology to ease the adoption of new practices on a larger scale [23]. At ADPC, the changes started with technology, but organizational and management changes were also implemented. The organizational structure was revised, both at the beginning of the migration project and after it, with smaller restructurings during the project and the period after it. However, the organizational changes appear to have been actions of last resort, and the latest change in 2008 was implemented only after a thorough problem analysis. The establishment of a separate Design Unit with its own first line manager appears to have made it possible to start the development of the design function in ADPC. Thus the importance of suitable organizational and management alignment appears evident in making technical process improvement possible in an organization [8].

Software process improvement has been claimed to require three key roles – a sponsor, a champion, and a change agent [24]. In particular, the design process improvement events support the need for all these three roles. The design process development responsibility was given to the Research and Development department in the organizational change of 2007, but to little avail. Only when an individual designer took up the task of developing a design document template and process of her own volition, did changes take place. However, even this had marginal impact at the organizational level in the absence of a champion whose role is to lead the improvement efforts through teething problems and hardships. When the Design Unit acquired a manager who was personally committed to take up the process improvement champion role, and in many cases also the change agent role, actual improvements started to show. Based on the observations in the Software Development department in general, the Department Manager did not hesitate to work as a sponsor for the process improvement efforts when the need was clearly expressed. This case study therefore shows how the champion and change agent roles can be, at least initially, outsourced to some extent to cover for the missing competences in-house, as evidenced in both the test and design process improvement

efforts at ADPC. The case study also demonstrates the volatility of the sponsor role, as the Managing Director who initiated the migration project left the organization in the middle of the project, reflecting what has been claimed as a common phenomenon in technology change projects in general [4].

The case study also highlights the role of tools, technical infrastructure, and competences in process improvement efforts. In particular, all the development efforts have included an investment in new tools (e.g., software development environments, a testing tool, ongoing adoption of a requirements management tool, and getting the workflow system working with the help of a tool). This approach has worked well at ADPC, but it has been complemented by active training in the process areas concerned, and by company-specific development standards and templates (e.g., the technical infrastructure). All these aspects are also included in the formal competence evaluation and improvement scheme for personnel introduced recently.

The standard advice on starting process improvement in an organization is to use an incremental model (e.g., [7]). As per the present study, the problems of large-scale changes are easy to see. However, the study does suggest one way to plan the phasing of process improvement actions – ADPC started by improving the development or programming phase first, then moved to testing, and finally initiated the improvement of the design and requirements phases. In practice, the phasing was a natural outcome of estimating and addressing the biggest problems that were constraining the development process at each point in time (cf. [25]).

4.2 Technology S-Curve

The duration of a technology lifecycle has been reported to vary considerably between organizations. However, an overly long technology lifecycle can inhibit change and make an organization vulnerable to advances from other organizations that have gained an attacker's advantage. When the established status quo at ADPC was broken with the technology change, other changes started rippling through the organization, in turn causing further changes that deepened and widened the performance dip. During the sustained process improvement phase, a more alert technology switching strategy was adopted by ADPC which resulted in a larger number of changes but which, on the other hand, made it possible to isolate the changes better and thereby avoid the ripple effect. The prolonged technology lifecycle made ADPC vulnerable to competition but, as the potential of the Internet was acknowledged and the change project was started without delays, ADPC avoided any major attack on its business and established a stronger market position than ever before by having both up to date tools and processes in use.

ADPC's performance degraded seriously during the migration project. This was, to some extent, caused by essential and unavoidable events like the need to learn new tools, technologies, and practices as well as the development of coding standards and application frameworks. The accidental reasons that should have been avoidable include the general turbulence and uncertainty caused by organizational changes, as well as change resistance at a more personal level resulting from insufficient communication about the changes with employees [16]. The personnel did concur with the management about the need for technology change, but directing this change

to a Microsoft platform raised feelings with people accustomed to working with mainframes. The resistance was evident from the beginning of the project, lasted until the beginning of the continuous process improvement phase, and manifested itself through a general reluctance to take up new tasks, the slow completion of tasks, and motivational problems as the migration tasks continued for a further year after the system switchover.

In previous studies, technology discontinuities have not been associated with the Classic Change Curve. However, in the present study the Classic Change Curve was observed in a technology change project, providing a reason to suspect that major technology changes are natural candidates for the performance dip and, after a while, for attaining significant performance improvement.

4.3 Learning Software Organization

The decade of active development observed at ADPC has highlighted those characteristics that suggest it has evolved into a learning organization [5]. For example, internal training programs, certification efforts, and the initiation of a competence development scheme has demonstrated an awareness of the importance of personal maturity. The documentation of development standards, processes, and job descriptions has demonstrated the desire to build a shared vision. Team learning was exercised in internal training sessions, in the transition from developer specialization on applications to process areas, and with the Design Unit improvement effort. The overall mental model at ADPC was one of quality and efficiency, as shown by the establishment of the Test Unit and process metrics. Finally, systems thinking was evident in the broad approach taken to include personnel development, the introduction of tools, processes and standards, recruiting, employing external consultants and nearshoring.

From the technological point of view, ADPC has started active experimentation of new tools and practices as shown by the project estimation tool development, the conduct of function point analysis, and by piloting a small project process model. These actions indicate the desire to know more about the currently used technologies, their limits, and the potential of competing technologies for their business. Finally, the frequent development technology changes after the migration project suggest that ADPC has moved from adopting a passive technology strategy to employing an alert one. Even if the present active technology switching strategy appears expensive, the alternative of getting locked into an outdated technology and then needing to change it in a major project overhaul at some point in time, with past experience, now appears the less attractive alternative.

5 Conclusion

The case study described in this paper reviews an organizational culture change that started as a technology project switching from an outdated mainframe system to an up to date network server system. The original project schedule and work estimates proved seriously underestimated, and the system switchover took place over five

years after the project start, and another year was needed to assure that all the systems used throughout the year were satisfactory to the users. The technical development was supplemented with process improvement actions to introduce standards, processes, and common practices throughout the organization. However, in many situations the technical development was not possible before the organizational and management aspects had first been addressed. The process infrastructure development has continued in the organization after the migration project completed for almost five years now, and the focus has currently moved to sustained process improvement to support efficient and quality software development.

The key findings from the study support the common perception that major organizational changes are often difficult to get started and laborious to go through. Consequently, it seems justified to aim at smaller changes even if the frequency of the changes becomes higher, and more effort needs to be invested to learn the limits of current technologies as well as the key features of competing technologies. In the present case study, the outcome of the migration project was a success, even though the original project estimation and planning for the change project left a lot of room for improvement. Finally, both the change project and the sustained process improvement phase thereafter demonstrate the systemic nature of technology and software development changes, whereby changing one element in an organization is likely to affect many others.

The work at ADPC has returned to normal business after the major internal changes. However, even though little can be said about the main issues to be tackled next, it is clear that the environment is changing all the time and that there is little room to be complacent and simply enjoy the past successes. One aspect that will be crucial to enable ADPC to compete for business in the future will be nurturing those employee competences that can help it adapt to changing situations. To respond to this challenge, ADPC is currently working towards a formalized competence evaluation and development scheme to help and motivate employees to develop their skills in the areas considered important for the organization's operations going forward. It is hence reasonable to claim that ADPC has transformed itself into a true learning software organization through the past decade of its experiences and changes.

References

1. Jacobs, S.: Introducing Measurable Quality Requirements: An Experience Report. 4th IEEE International Symposium on Requirements Engineering. IEEE Computer Society, Limerick, Ireland (1999) 172-179
2. Kauppinen, M., Kujala, S., Aaltio, T., Lehtola, L.: Introducing Requirements Engineering: How to Make a Cultural Change Happen in Practice. IEEE Joint International Conference on Requirements Engineering. IEEE Computer Society, Essen, Germany (2002) 43-51
3. Deal, T.E., Kennedy, A.A.: Corporate Cultures: The Rites and Rituals of Corporate Life. Addison-Wesley Publishing Company, Reading, Massachusetts (1982)

4. Foster, R.N.: Innovation: The Attacker's Advantage. Summit Books, New York (1986)
5. Senge, P.: The Fifth Discipline: The Art and Practice of the Learning Organization. Currency, New York (1990)
6. Christensen, C.M.: The Innovator's Dilemma. HarperBusiness Essentials, New York (2003)
7. McFeeley, B.: IDEAL: A User's Guide for Software Process Improvement. Software Engineering Institute, Pittsburgh, PA, USA (1996)
8. Zahran, S.: Software Process Improvement: Practical Guidelines for Business Success. Addison-Wesley, Harlow, England (1998)
9. Foster, R.N.: Boosting the Payoff from R&D. Research Management **25** (1982) 22-27
10. Becker, R.H., Speltz, L.M.: Putting the S-Curve Concept to Work. Research Management **26** (1983) 31-33
11. Maurer, R.: Beyond the Walls of Resistance. Bard Press (1996)
12. Lewin, K. (ed.): Group Decision and Social Change. Henry Hold, New York (1952)
13. Kubler-Ross, E.: On Death and Dying. Touchtone, New York (1969)
14. Menninger, W.W.: The Meaning of Morale: A Peace Corps Model. In: Moyniham, D.P. (ed.): Business and Society in Change. American Telephone and Telegrapch Co., New York (1975)
15. Elrod, P.D.I., Tippett, D.D.: The "Death Valley" of Change. Journal of Organizational Change Management **15** (2002) 273-291
16. Schneider, D.M., Goldwasser, C.: Be a Model Leader of Change. Management Review **87** (1998) 41-45
17. Elrod, P.D.I., Tippett, D.D.: An Empirical Study of the Relationship Between Team Performance and Team Maturity. Engineering Management Journal **11** (1999) 7-14
18. Nikula, U., Oinonen, P., Hannola, L.: Extending Process Improvement into a New Organizational Unit. In: Fidge, C. (ed.): 20th Australian Software Engineering Conference. IEEE Computer Society, Gold Coast, Australia (2009) 267-276
19. Microsoft Corporation, `https://partner.microsoft.com/`
20. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall (2001)
21. Jones, C.: Applied Software Measurement. McGraw-Hill (2008)
22. Davis, A.M.: Just Enough Requirements Management: Where Software Development Meets Marketing. Dorset House (2005)
23. Schein, E.H.: Organizational Culture and Leadership. Jossey-Bass Publishers, San Francisco (1992)
24. Humphrey, W.S.: Managing the Software Process. Addison-Wesley, Reading, Massachusetts (1989)
25. Goldratt, E.M.: Theory of Constraints. North River Press (1999)