Improving globally distributed software development and support processes – A workflow view

June 26, 2012
November 14, 2012 (1st revision)
February 11, 2013 (2nd revision)

ABSTRACT

We document a three year long process improvement project in a globally distributed engineering company developing, delivering and supporting a complex software system with tailored hardware components and unique end-customer installations. By applying the domain knowledge from manufacturing and production operations management on lead time reduction and its multiple benefits to process performance, the workflows of globally distributed software development and multi-tier support processes were measured and monitored in a company-wide manner. The results show that the global end-to-end process visibility and centrally managed reporting at all levels of the organization catalysed a change process towards significantly better performance. Due to the new performance indicators based on lead times and their variation with fixed control procedures the case company has been able to report faster bug fixing cycle times, improved response times and generally better customer satisfaction in its global operations. In all, lead times to implement new features and to respond to customer issues and requests were reduced by 50%.

Keywords performance metrics; software process improvement; software quality; support process; complex system product; global software engineering

1. Introduction

Software process improvement projects are recurrent management initiatives in most companies developing, maintaining and supporting software products. These projects seldom have long lasting impact on the organizational performance as the changes are not sustained (Napier, 2008). Heavy duty approaches like the implementation of the capability maturity model (CMM) are laborious and also suffer from long term support from the management (Jalote, 2000). In a similar fashion, the implementation of various recommended software standards is seen as the first step, yet far from the conclusive step towards consistent software quality and processes (Yang, 2001). As process development is concerned, the Total Quality Management (TQM) legacy is vast and has been broadly implemented in software processes, yet even there critical improvements have not been lasting, especially when projects and processes are changed (Ravichandran & Rai, 2000). The increased outsourcing of software development tasks has not improved the situation as these tasks generate changes in the processes and cost reduction targets are not met (Caldwell, 2002). Maintaining and improving software quality remains to be a challenge both for in-house and geographically distributed networked operations.

To address this challenge we document a longitudinal and in-depth case study on how a globally operating engineering company delivering complex and tailored hard- and software systems improved its development and customer support processes. The problems prevailing in distributed software development are known to many companies and several approaches have been applied to improve these processes. They have also been extensively studied by academic scholars (for a summary see Da Silva et al., 2012). Issues related to quality are a major concern in these studies. After studying 189 globally distributed software projects Cataldo and Nambiar (2012) show that distribution of developers across locations along with architectural and technical linkages between these development teams affect significantly software quality. Taxén (2006) documents how complex telecommunication system delivery processes were improved by establishing a central control on different subproject interfaces and through the operationalization of the engineering processes and their coordination; the latter meaning the division of the process into elements in which they can be measured and observed as an independent entity. Despite the challenges related to the management of distributed and knowledge intensive operations, they make it possible to harness diverse expertise and ease access and improve the service provided to globally scattered clients.

According to Boutellier et al. (1998) the challenges could be managed with modern IT tools and centralized project management. However, it is, in reality, more complex than that, especially when operations are performed in a geographically distributed manner by interconnected units in different continents.

Relationships between organizational context and management information system (MIS) structures are significantly correlated with organizational structures which, in turn, are closely associated with organizational size (Ein-Dor and Segev, 1982). Most global companies are the result of numerous mergers and acquisitions and therefore their structure by default is decentralized, heterogeneous with a mosaic of different cultures, information systems and leadership traditions. In their thorough case study Sambamurthy and Zmud (1999) show that companies with a past filled with mergers and acquisitions seem to result in a federal and decentralized governance mode, meaning that IT responsibilities vary from one business unit to another and that centralized control is hard to accomplish and information system related decision-making is located at divisional or even lower organizational levels. This type of corporate environment, which has emerged through numerous large scale mergers with its problems to manage complex software development and support processes, sets the scene for our case study. Therefore, we simply reflect on how a company in this kind of heterogeneous and global environment can initiate and maintain company-wide software development and support process improvements. We also assume that applying the operations and production management domain knowledge to lead time reduction can be used to the workflows related to the globally distributed software development and support processes.

This paper addresses operational efficiency issues related to global companies with cross-continental workflows in complex system development and support processes. Improving these processes continues to be challenging as the business environment is changing continuously through increased outsourcing and off-shoring of operations. Edgell et al. (2008) indicate that economic slowdown and fluctuations in general direct companies towards cost-driven outsourcing, despite the fact that, over the long term, service-driven or value-driven deals tend to deliver more stable, successful relationships. This entails that the value network in which the development and support processes are managed is continuously changing, thus making process improvement even more difficult. But managing remote sites, be they own or outsourced, may generate all kinds of friction to the development work through distrust and fear (Piri et al., 2012).

The challenges related to global software process improvement are numerous; cross-border and intercontinental workflows, different computer systems and protocols inherited over time, collaboration with off-shored units and, naturally, increasingly demanding customers with their unique requests (Šmite and Wohlin, 2012). To address these challenges we document a case of a global company and its efforts to speed up the workflows related to software development process and customer request response times, in order to improve the overall software quality and customer experience.

To achieve this, the rest of the paper is structured following the action design research approach (Sein et al., 2011) in the following way. First, a literature review of software quality, process improvement and distributed operations is presented to detail the main drivers for the case study. Then the case company is described by reviewing its history and evolution to a global, yet heterogeneous and cross-cultural organizational structure and the particular challenges faced by their software processes. This is followed by a chapter detailing the key research questions from the literature review and the case characteristics, including a discussion of the justification for the single case study and for the action research methodology together with the scope of the study. Details of the software development and support process are then discussed and the improvement project is documented, mentioning how it was initiated, what goals were set and which metrics were implemented. Finally, the results and lessons learned are documented with implications for managers and for future research.

2. Literature review on software process improvement

Several high-profile software failures in the 1980s and 1990s, like the Airbus 320 and Ariane 5 together with the seminal Standish Group study (1995) on software projects rarely meeting their objectives triggered a software quality improvement effort in the industry (Niazi et al., 2005). Numerous studies indicate that the root cause for software quality problems lies in the management and/or in the lack of the software process and its continuous improvement. Vitharana and Mone (2008) study software process improvement (SPI) models like CMM and ISO 9000, and identify six critical factors of software quality management (SQM) to develop an instrument that can be used to measure critical factors of SQM. These six critical factors are top IS management commitment, education and training, customer focus, process management, quality metrics, and employee responsibility. These factors are very broad and describe the

whole software process environment with all the key dimensions, which serve well for the structure of the literature review.

Human factors, be they commitment or competence, seem to play a crucial role in software quality. McDermid and Bennet (1999) have argued that human factors in SPI have been ignored and this has damaged the effectiveness of SPI programs. Hall and Wilson (1998) also suggest that the experiences, opinions and perceptions of software engineers have an indirect impact on the quality of the software produced. Further in this line of research, Baddoo and Hall (2002) show that there are different motivational clusters among the software developers which should be understood to better manage the software development process. Following this trend Prikladnicki (2012) applies psychological and physical distance to analyse the collaborative workings of the remote teams, proving that things unknown to project managers are happening irrelevantly to physical distance. Naturally, training and concerted actions are important when managing distributed software processes, thus actions like continuous training and education of those involved in the software process along with disciplined reviews and respect on design standards are important (e.g. Rainer and Hall, 2002; Subramanian et al., 2007). Different skills and working traditions in globalized software processes call for unified procedures in order to keep the software process under control, therefore most of the SPI training aims to establish common working procedures (Pettersson et al., 2008).

Software process maturity is directly linked with the expertise and discipline of the people involved in the SPI, which, in turn, affects the productivity of the process (Chapin et al., 2001). Harter at al. (2000) studied the relationship between process maturity, quality, cycle time, and effort for the development of 30 software products by a major IT firm. They found that higher levels of process maturity as assessed by the CMMI are associated with higher product quality, but also with increases in development effort. Findings indicate that the reductions in cycle time and effort due to improved quality outweigh the increases from achieving higher levels of process maturity. This means that the net effect of process maturity shows in reduced cycle time and development effort. Thus, process maturity correlates not only with product quality but also with operational efficiency.

Ultimately software quality is determined by user satisfaction. After a thorough literature review and a follow up survey, Issac et al. (2006) conclude that employee competence is

a crucial factor affecting software quality. They also found that product attributes i.e. reliability, integrity, portability, extensibility, flexibility, reusability, functionality, and maintainability, are vital for customer satisfaction. Customers see that the characteristics of software are highly influenced by process quality management and competence of the employees. Quality is judged indirectly by focusing on the effectiveness and usefulness of the delivered system in performing the task for which it was designed. In other words, quality refers to whether the system is constructed and performs as it should have been built and as it should perform. Moses (2009) argues that direct measurement of quality attributes should be encouraged and that such measurement can be quantified to establish consistency and continuous improvement. Sousa and Voss (2009) show that services failures, once happened, can be successfully worked out when correcting things swiftly.

Service and support activities are directly linked with customer satisfaction and company success. Watson et al. (1998) show evidence suggesting that management's attention to service quality has not been consistent and that ignoring it is harmful for the company. Information system management needs to recognize that service quality is not a fad but an on-going commitment and the CIO must continually pay attention to IS service quality. Continuous process performance measurement and information visibility at all levels of the process are vital for software companies as Bharadwaj (2000) documents results indicating that firms with high IT capability and process maturity tend to outperform a control sample of firms on a variety of profit and cost-based performance measures. Better control reduces process variability, thereby improving operational efficiency and software product quality. After studying 37 CMM level 5 projects in four organizations, Agrawal and Chari (2007) find that high levels of process maturity, as indicated by CMM level 5 rating, reduce the effects of most factors that were previously believed to impact software development effort, quality, and cycle time. They also show that the biggest rewards from high levels of process maturity come from the reduction in variance of software development outcomes that were caused by factors other than software size.

Lead time based metrics can be applied to any business process and workflow (Bartezzaghi et al., 1994), yet the metrics for software development projects and processes are numerous and seldom based on time-based measures. Project related metrics follow traditional earned value and effort versus outcome metrics (Raffo, 2005). Process related metrics are traditionally based on various lead time, work-in-progress,

throughput and punctuality versions depending on whether testing, in-process activities, architectural design or software engineering are concerned (Kan, 2002). Traditionally, seven tools of quality have also been applied to software development and support processes and total quality management is a well-established part of the software process improvement (Vitharana & Mone, 2008). The total quality management approach is widely used in any process development (Kueng, 2000), not only in software, but also in manufacturing, services and governmental process. SPI does not differ from other process improvement projects. They all fundamentally aim to reduce variation and lead times, while increasing output and improving punctuality. Challenges to improve software related processes remain and become more significant when the work is done in an emergent network of different cultures and technologies, as the case company will show.

Global software development and outsourcing has been widely studied. Like with any outsourcing the main motivation for it is motivated by cost efficiency. Yet, it does not come without challenges and vast amount of global software development research has detailed numerous problems related to managing processes, quality and to maintain the costs at right level. Global software development is increasingly a multisite, multicultural, globally distributed undertaking, where challenges stem from temporal, geographical and socio-cultural distances between the virtual development teams (Siakas & Balstruop, 2006). Managing these challenges requires clear management procedures, with technical tools enabling workflow monitoring and measurement.

Insert Tables 1 somewhere here

The above literature review shows a glimpse of the vast research tradition in software quality improvement and measurement. Following Jalali & Wohlin (2012), Table 1 summarizes the main issues related to global software development and the management of software quality. We aim to follow this research suit by extending the software quality management and customer service to a global environment of a major company in industrial information technology. This aim is backed also by the profound literature study by Šmite et al. (2010), who show that global software engineering research lacks in-depth case studies. To meet this challenge we document the key findings of a three year long change management project aiming to improve software

quality and request response times in a global setting, where work is done on different continents. Issues concerning how the improvement project should be managed, what metrics to be used and how to institutionalize the changes are reviewed. These are all critical issues for most of companies developing, delivering and maintaining complex software systems, where the systems are often the result of an amalgamation of different sub-systems through various acquisitions and partnerships. Initiating and maintaining process improvement in this type of heterogeneous software development and support environment has seldom been discussed in the academic literature.

3. The case company and its challenges

3.1 Building a global player through acquisitions

The case company was created through a major merger between two large engineering companies in the end of the 1980s. The merged companies were specialized in the engineering and production of electrical equipment, turbines, motors, generators and transformers. Both companies had an industrial history stemming well over a century of pioneering work in building electrical infrastructures. By joining forces, the idea was to build a global industrial giant that would be capable of competing against the two major global rivals in the industry. The rationale behind the new competitive advantage was based on an unsurpassed in-house technology portfolio behind one single corporate brand to be sold directly to utilities and industry conglomerates in large infrastructure businesses.

Radical organizational change was implemented to globally manage local businesses, also known through the slogan "local everywhere". This was to be the key differentiation against the targeted rivals to challenge the fact that the merged company did not have large home markets unlike its key competitors. To pursue its new strategy the company started to acquire domestically operating European companies in power generation, transmission and distribution. The operations of the acquired companies were mostly kept untouched as local governments wanted to secure the local capability to serve their electrical infrastructure in case of crisis. This was an even more understandable request as most of the countries had used different standards when setting up their electrical infrastructure. Most of the acquired companies were made to serve as the main supplier for their home country's infrastructure. The new global products were to follow these markets as soon as the new strategic product and technology development would be completed.

The acquired companies were old-fashioned mainly locally operating engineering and technology companies and therefore their balance sheets included undervalued fixed assets (especially land and buildings). By liquidating these assets the company had the opportunity to fund further growth. It was time to shift the focus outside of saturated Europe, and with its experienced engineering workforce the aim was set on the electric infrastructure in Asia and Middle East. The strategy used in Europe was also applied to the Asian markets, and the approach was well received by Asian and Middle East governments since it provided them with an approach where they could be self-sufficient for their electrical infrastructure in case of economic and social instability. At the same time the company acquired a foothold in the North-American market. This wave of acquisitions took place in the 1980s, and more were to follow toward the end of the century and finally the company achieved its world leading position in automation and instrumentation. In some industries the company practically ruled the market.

Through the acquisitions and growth of local, yet global operations the company had to face the issue of underutilization of its engineering resources which were spread over tens of countries operating in an autonomous manner. Restructuring was costly and barriers imposed by national standards, different languages and cultures hindered the efforts. The different information systems also constituted a significant problem for process integrators. In addition to the "local everywhere" concept, that provided all units with the production capacity to address the global markets, it created strong internal competition. The strong position of local management made it very difficult for the global management to address the internal competition through market allocations and operation closures.

3.2 Building competences

The company started a massive investment in R&D to address two difficult problems: overcapacity in engineering and globalization of product and technology portfolios. The existing engineering capability was leveraged to build leading edge technological products since downsizing of these resources would have been costly and politically too risky. The Internet boom fuelled the company to focus on industrial information technology as a vital part of their product offering. Simultaneously the company continued to develop new breakthrough technologies to replace local products with new and

improved global products. One fundamental issue remained; the company had a lot of software capable engineers, but no history in running a successful software business.

Due to the numerous acquisitions the company was like a patchwork quilt and was faced with serious challenges in engaging in R&D with such a distributed organization. The responsibility for development initiatives was given to different countries based on political needs rather than the true competence of the respective operations. At the same time major efforts were made to integrate IT systems so that some visibility was achieved to global support and development processes. Common to all of these activities was that the development programs resulted in cost and time overruns and had severe quality problems. Products were delivered late with inadequate quality, the company needed more IT skills instead of control systems skills and the system was lacking functionality required by the market, and which the competing systems had.

To tackle these problems the company made a radical organizational change. The company was to run three different business models simultaneously. First, there would be a product division responsible for creating technologically leading products and strong indirect sales. The customer division with an ultimate customer intimacy approach was to sell the total range of the products through a customer segmented sales force. Finally, the information technology and support division was planned to interconnect the products seamlessly and simultaneously with the customer's value adding process. The main idea was that the customer is locked-in with the company for a long term relationship. The aim was to get scalable business with global proprietary products and software, and a common global sales and engineering force for all products.

Focus was also put on life cycle services as the customer base included users of practically all system versions ranging over 20 years of development work. This meant that customer specific application engineering and commissioning as well as life cycle services represented a significant source of the total revenue generated. Being an advanced and innovative company the software products were based on object oriented software architecture long before object oriented programming had become common in software industry. This approach enabled the company to liberate itself form dealing with multiple applications, operating systems and platforms. The new platform allowed any application to see and manipulate data in real time giving the company a significant advantage over the rivals.

3.3 Challenges of distributed operations

The way the company had evolved generated several challenges for the management. Focusing on the software process improvement we detail here the following software process improvements to shape the research questions for the rest of the article. The software engineering platform that is used to develop customer solutions from different hardware and software components is developed in different locations or on-site at external suppliers. Customer solutions are designed at an engineering centre and then finally commissioned to the end-customer site. Coordination is vital. For example, a typical delivery in which platforms and modules are developed in four different countries on three different continents, can be required to be operational in another country on another continent. Special challenges are related to the metrics that must be implemented to improve product quality, operational performance and delivery efficiency in the development and support processes. From the value chain point of view, it is necessary to determine how the company can better integrate its distributed R&D and support operations in a more efficient manner.

The current product offering is monolithic and hybrid, a combination of MS Windows and other platforms, with several in-house built processing units handling I/O streams from tens of thousands of control points in one typical installation. In all, the system consists of tens of millions lines of code. 80% of the customer base runs older versions of the systems than the company is currently offering for new installations. Industrial companies operating their factories are reluctant to upgrade as long as the existing system runs well. The customers expect that the system lifetime is well over ten years without major upgrades. 20% of the customers use the latest version of the system, and sales offers one solution. New product development and related support teams of the latest software version are overloaded with the older system versions, and their work gets hampered with multiple requests. The underlying heterogeneous software platform does not make things any easier, especially when part of the development and testing activities have been off-shored.

The challenges the company faces on its current technology base are linked with challenges listed in the literature review:

- How to organize workflows at different levels of the multi-tier[1] support processes to manage multiple software versions in a distributed environment?

- How to measure and improve the multi-tier support processes to reduce response times and variation in the process?

- How to collect vital user experiences to improve overall product quality of the system versions to come?

Today the case company is the result of numerous mergers and acquisitions resulting in numerous independent units and several different technologies. It is not surprising that the company has sometimes been labelled a "loose federation of independent nations". The challenges they face in their customer support are to some extent known to any company delivering, developing and supporting complex software in a distributed manner.

4. Research methodology, questions, scope and sample

4.1 Research methodology

This research is based on a single company case, and it includes change actions with controlled observations and deductions based on individuals, groups, organizational setting, hardware and software configurations (Lee, 1989; see also Cavaye, 1997). To achieve this we apply action design research (ADR) methodology (Sein et al., 2011), which assumes that the information technology artefacts are ensembles shaped by the organizational context during development and use. Following Sein et al. (2011) ADR shares two key challenges, the first being the definition of the problem situation encountered by the specific organizational setting through interviews and other objective evaluations. The second issue concerns the construction of research problems encountered in the situation in which the artefact or company is found. These two challenges have been defined and documented in the previous chapter on the case company and its evolution in the current operational environment.

We also follow the traditional action research tradition dating back several decades to the development of social theories and group dynamics (e.g., Argyris et al., 1985; Coghlan and Brannick, 2001). In its simple form, action research is an approach to take an action,

---

[1] Here multi-tier refers to the four level support process (L1-L4), where the first two levels are at the customer facility, one on the system and operational level and the other on the application level. Levels 3 and 4 concern back office operations, of which the first concerns regional support services and the second one actual R&D dealing with the source code.

or to document and observe a phenomenon caused by an action, and to create knowledge or develop or refine theory from that action and its consequences. Action research tests the capacity of a theory to resolve problems in real life and in practical situations. With its various paradigms, action research has a strong position in social sciences, and for the present research the focus is on learning through interaction, which results in an increase in the ability for practitioners to solve problems. In our case study the action involves the implementation of a new performance monitoring system for the software development and support process and the related coaching to manage the processes accordingly. This action is based on the operations management principles stemming from efficient manufacturing processes. This means that in order to follow the action research tradition the researchers have been professionally involved in the case for several years and the work has been documented over the past two years through active participation in the change process.

This means that our research action is in line with the design science approach (Holmström et al., 2009, see also Kuechler & Vaishnavi, 2008) as we use the well-established domain knowledge from operations and production research on the efficient management of distributed software development and support processes. According to the design science approach the work is broadly divided into four progressive steps, the first being the development of an initial solution design, and secondly applying the solution to a real problem and documenting the outcomes. The third step is dedicated to explaining and developing a substantive theory and establishing theoretical relevance. To achieve the fourth level of contribution and development to a formal theory, i.e. strengthen theoretical and statistical generalizability, further explanation is needed. This approach follows the Stuart et al. (2002) conclusion that general research as well as case research can be broken down into five critical stages: definition of the research question, instrument development (including site selection), data gathering, data analysis and dissemination.

We cannot avoid touching the case research tradition as Yin (1994) defines it. For him a case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context when the boundaries between phenomenon and context are not clearly evident and where multiple sources of evidence are used. We also apply the case study guidelines for software engineering (Runeson & Höst, 2009). According to Eisenhardt (1989) the main strengths of active case research method are its novelty, testability and

empirical validity. Gummesson (1991) considers that this is also one significant reason why the action research approach is seen to be the most far-reaching method of case studies.

4.2 Research questions

Based on the ADR approach, literature review and the case description the work sets out to study the following research propositions:

− What are the key workflow performance indicators for a process improvement project in a distributed, heterogeneous and multi-tier software development and support organization?
− How to initiate change in such an environment towards more efficient processes and to sustain the transformation with continuous improvement?

The approach, or action, applied to improve the processes followed the key theoretical dimensions stemming from the very core of operations management (Schmenner, 2001). These basically concern four main themes. The first is value-adding and value-creation at task, process and whole operational level, i.e. the work must somehow produce quantified value for the company. Operations not producing value should be eradicated and focus should be directed on increasing value in such a manner that customers are best satisfied and company stakeholders receive a decent return on their investments. The second theme highlights operational speed and its improvement by removing hindrances to improve performance. This simply means that operations should take place swiftly and predictably and that operational lead times should be under continuous control with the aim to relentlessly reduce them. The third dimension states that variability is inherent in every process and generates delays and disturbances. Usually the more players in a process the more prone it is to the negative impacts of variability. Efficient operations management aims to reduce man-made variability and the vulnerability in external and natural variability. The last and fourth dimension promotes service level and customer satisfaction, which should form one of the key measures for operational performance. These include punctuality, fast response and right quality level delivered to fully meet customer requirements. These key operations and production management principles are the basis for the applied process improvement approach.

4.3 Sample and scope

A total of 32 people on three different continents and at different support levels were interviewed numerous times over a period of three years (Tables 2 & 3). The people were either general managers (8) or directly responsible for development and support activities (24). The support process was studied through quantitative data on support requests with their workflow lead times and variations during the whole project period. These requests include all possible requests and their escalation to different levels of the support process. They also concern different versions of the supported system and its different sub systems. Using longitudinal data analysis covering a period of three years, the various trends of the workflow performance in the development and support process could be depicted and indicate how the various changes affected the outcomes. Throughout the data analysis, records were kept on each session and interview, all of which were approved by the concerned people and the interviewees. This means that both quantitative and qualitative data were used to compile the main findings and these were checked to ensure they did not contradict each other. We follow Myers (1997) who writes that as the focus of information systems research shifts from technological to managerial and organizational issues, qualitative research methods become increasingly useful.

Insert Tables 2 & 3 somewhere here

Fundamentally, we document a business process reengineering project (BPR). Following Kettinger and Teng (1997), the BPR techniques and tools form a knowledge base to improve business process change practice and provide a basis for future BPR research. We also contribute to this research tradition, and therefore issues that are beyond the scope of this paper include actual software product configuration and technological solutions and decisions made. With the workflow view we focus on the software development and support processes of the complex system in a distributed organization by using the action design research approach.

5. The change project

5.1 Motivation for the project

The justification for the change project originated from the customer feedback clearly indicating that less software updates were needed, even though most of the updates

included new useful functionality. The company performed an internal and external study to find out how they could improve their customers' performance when they use the system. To establish this outside-in view, i.e. how customers appreciate using the system, was vital to set the benchmark to improve the quality perception of the system both internally and externally. The internal questionnaire was conducted by people covering the whole value chain from sales through delivery to support, while the external questionnaire covered customers working with the system every day. The results showed two main concerns: 1) certain limitations of the system remain even though they were escalated several times, and 2) there was a steady flow of various new quality issues after every update.

To move on along the operations management path, the first objective was to reveal the end-to-end lead times and service levels of the problem reporting and correction process (see Figure 1 for an overall view of the process, and Figure 2 for a typical process for one request). All this was based on real process data on how customer complaints are handled. The data used was retrieved from three different request/incident management systems that basically are enhanced workflow management systems. The data set covers a time period of four years (2005-8, actual change project started in 2006) and the total number of customer requests and their workflows analysed was around 16'000 originating from over 1'000 customers using various versions of the software. Figure 3 sums the overall data set with the number of software versions in use by a customer and how requests originate from these customer groups. The figure shows that the more versions are used by the customer the more requests it generates per customer. It also shows that about 90% of the customers use less than 3 versions of the system.

Insert Figures 1 & 2 somewhere here

From the analysis point of view focus was on back office support levels L3 (system and module level, handled regionally) and L4 (product and source code under the R&D department), which make most of the long lead time experience by customers and are mostly concerned with possible real problems with the software. Requests that are solved at customer unit form levels L1 (plant) and L2 (application at the plant level) are handled by customer service units. The three year time window helps to reveal the dynamics in

the whole support process, including how customer complaints originate (in batches, after upgrades etc.), how complaints are batched and work is routed to different support units. The long term view also supports the research aim to enhance and complement the existing key performance indicators (KPIs) and related performance dashboards for the support processes. Thus, in short, the aim of the change project was to improve end-customer satisfaction and software quality.

Insert Figure 3 somewhere here

5.2 Workflow process analysis

Process data was retrieved from different systems. The first analyses concerned support requests and their average lead times with their development trends and distributions. This general information was then reviewed at the sub-system and product version level. Other attributes for the analysis of the support requests concerned their severity and criticality. Further analyses also included the origin of the supplier request, for example which customer and from which geographical location or business unit the request came. To illustrate the initial analyses, Figure 4 shows how the overall service requests get distributed longitudinally along the sample period. The versions also include the releases of different service packages, which shows that gradually the versions mature and the number of requests decrease. The work load varies not only in terms of number of requests being treated but also by their criticality. Figure 5 shows the quarterly average of request handling times in days and their count. Clearly, L3 support feeds L4 in slight delay and the number of requests shows an increase in handling times. This follows the central operations management principle stating that, while work-in-progress increases and the throughput rate of the system remains more or less the same, the lead times will increase. But measuring the lead times speeds things even at L4, like one software engineer stated that "lead time measurement is very important for a customer, but it was not emphasized in our often quite internal focused KPIs".

Insert Figures 4 & 5 somewhere here

The immediate outcomes of the analysis indicated that the lead times in the handling were very long and their variation was significant. This means that the work-in-process is

large due to the queues and parallel processing of the requests (the same problem can create a request from several customers on different continents leading to repeated support work). This also means that the predictability of the finishing of the task was difficult, thus contributing to low customer satisfaction. Also allocating capacity to these situations is difficult as work load fluctuates making the planning of human resources also more difficult. What could have been the reason for this? The data points indicate that there is a problem with matching the plans with the reality.

Requests are released as they are independent of the capacity situation, no planning is made based on the load situation and all requests are queued as they arrive. Some of them are classified based on their urgency. Open requests prolong their processing lead times eventually affecting the software quality and delaying the testing and releasing of the new versions. Several open requests may also generate a collateral effect due to their interrelated impact when they get handled, thus generating more unforeseen problems in the software that may not be caught in routine tests. Once requests are received their proper and thorough processing is vital to have a proper diagnosis and related decisions made at the first step of the support process. Too lax or even incompetent and over cautious releasing of requests to other support levels often generates unnecessary work and overall loads that further make situations at lower support levels more difficult. It was clearly seen from the data that proper gatekeepers, who are not numerous in the total population taking part in the process, are essential.

To facilitate change and better manage the project centralized reporting database was established to collect actual ticket flow data from multiple systems. This data was used to create a set of standard reports to relevant stakeholders. The global business management took a visible lead in making sure everybody knew about these reports and each KPI was to be understood and monitored. The improvement of customer request resolution time was made to be an essential part of every business review. This generated initiatives at all levels to improve the situation as a regional manager said: "When people became aware that the top management had become very interested in the improvements achieved in the 'Outside-In' KPIs, they themselves analysed what these KPIs meant to them and started their own improvement. This made the improvements come through much faster than the central project team could ever have managed to control."

5.3 Organizational aspects

A cross functional virtual project team with members from all geographic regions was established. The team was supported by external experts on process analysis and data mining. The team was mandated to visit customer support and development centres to perform interviews and to access databases to collect actual process data to capture the true state of the customer service quality. Additional targets for the project were to improve trust and co-operation between various support levels and to provide even L4 teams with a fact based view on how end-customers perceived the service provided by the whole support organization. The project was assigned to report to a steering committee of various stakeholders (business management, sales, service and finance representatives) on regular intervals. The steering committee had a central budget which it could assign to targeted improvement initiatives presented by the project. The project would oversee the implementation of the improvement actions, but the actual responsibility was at the target unit. Impacts and results of these actions were monitored through a centralized reporting system.

Gatekeepers, who share installation specific information, are vital for the overall quality of the process. One of them sincerely suggested, "Why not spend more time on the request evaluation in order to avoid unnecessary non-value adding work in the upstream of the process". Another one had a KPI implemented that prevented his team from "overloading the next support levels with critical requests". These gatekeepers are mostly very knowledgeable people who have worked for a long time in the corporation and its various locations. They have accumulated a massive amount of tacit knowledge that should be documented properly. It became apparent that their decisions were documented in the workflow management system, but neither the reasons nor the inference logic behind the decisions made were documented. These people hold profound product know-how extending even to the knowledge of different customer installations and their past history of problems. Several things came up in the interviews. The database holding information on the installed base should hold detailed information on the history of updates and problems they have experienced earlier. This accumulated tacit product know-how was largely unexploited and not documented in the support database.

One support engineer at L3 summed the situation "The regional support centres have a complex role of prioritizing the customer requests and escalating them accordingly. Using the physical world logistic analogy has been helping us in our realm of information

logistics. Understanding the ABC-classification in regard to issue tickets has been helpful in my co-operation with my colleagues who are facing the customer. Not all issues are equally important, but for every ticket the lead time matters most to the end customer. And of course that the resolution has to be first time right." This highlights that the measuring the request workflow across support levels helps to bring the end-customer closer to lower support levels. The end-to-end visibility and its analogy to physical material flows was nailed down by another support engineer at L2: "I have always wondered why our ticket tracking system could not have the same feature as DHL has with physical packages. Why is it not possible for us or even the end customer to go to web and see where is an open ticket now, who is working on it or is it only waiting and when should the resolution arrive".

The source of the requests varied also, making their prioritization difficult in a system with single window processing. They could originate from daily use of the system, customers experiencing problems after an update, or projects being managed by the software vendor. Their source could also be traced back to brand new installations, new version development and testing. Yet, the main finding of the analysis was that independently from where the request originates it is treated the same way as all other requests. This means that their urgency was defined not by their source but their assessment at L1 and L2 levels. This also means that the same major support team was used to treat any problems that may occur from any possible source. The importance of the gatekeepers and their correct assessment of the request and consequent decisions in analysing the requests play a vital role. The only dedicated resources assigned to handle project specific requests were related to major upgrades in major industrial facilities, although even they were organized into specific regions at the level L3, thus bypassing the main support channel serving all the other requests.

The data shows that new installation and version update projects along the normal request flow should be separated and managed differently. These different sources of requests have their own time lines and urgency. Why should a critical request related to an upgrade jeopardizing the whole operation of the factory wait for a request related to maintenance with no immediate impact on the operation of the facility? Each request is related to a normal operation, special upgrade project or new installation. In addition, the request is related to a particular customer installation and version used in the facility. Prioritization of the processing of the request should be evaluated against the particular

case, i.e. customer, software version, module and the corresponding schedule. The analysis shows that the work is not prioritized and the support system treats requests as they arrive. This is also shown in the analysis related to severity level of requests and their processing lead times.

More detailed analyses were made on the specific performance of support process at the different levels, i.e. cases that do not cross from one level to another. Furthermore, the case company started to match implementation projects and version release timetables with the lead time performance. There was a special focus on how project milestones affect lead times and performance and how disciplined the processes are, e.g., features are accepted/added after a design freeze. Dedicated performance dashboards were developed to visualize the workflow and its performance. In addition, target lead times were set for different support levels, request types and modules. It was also noted that the data used for the analyses were incoherent due to different ways of coding and documenting the requests. Serious efforts were made to build up awareness of the importance of disciplined use of the support system. Special training events were developed to show bad and good cases. It became clear that the way the requests are coded in the database directly affects the software and service quality. Special attention was paid to gate keeping i.e. on the decisions to pass a request on to another support level. More discipline was introduced when documenting the request, including customer specific data on their installation and past support history. This information provides the gatekeepers with the much needed tacit knowledge on the special customer installations.

Gatekeepers' role is crucial in linking the end-to-end value chain from global customers to various global developers. In between is a network of global product management and global software support organization making the whole system a large network of loosely coupled teams. It is vital for the gatekeepers to have transparent and measurable access to request flow, they need to monitor this flow and when necessary actively join the resolution process. They need to interface with various organizational departments and to keep the request process under control, they need also disseminate information on their decisions. It is crucial for the management to define how the gatekeepers' performance is measured and how they are motivated and compensated.

6. The results and managerial implications

From the management viewpoint, the case study highlights the "outside in" process view, meaning that process metrics should be based on customer experience. The way the customer perceives company performance is more important than how the company perceives its performance. Ideally the process of improving software quality should be seen as the process of an express carrier with full traceability and procedures to react in case the shipment is delayed. In our case the throughput measurement was based on input and output flows of the requests and resolution with detailed analysis on process slack times and actual value adding, i.e. hands-on times. These were measured for different software versions, sites and levels of the process. These measures, which point out delays and general improvement trends, initiate a continuous improvement process mimicking the logic of Deming's Plan-Do-Check-Act cycle with a focus on internal communication and achieve the buy-in from the key stakeholders in the line organization.

The case company implemented an independent team to constantly analyse the throughput data and to convert it to actionable information by customer, subsystem and geography. To catalyse prioritization, a special weight was put on each request based on its origin. The priorities were signed accordingly according to whether it came from the customer, service manager of the company or consultant. Progress in speeding the process was reviewed periodically and reports were distributed to all those involved, including top management, which also monitored the process on a monthly basis. To promote management involvement an escalation routine parallel to a technical routine was established based primarily on the time the case was opened and second on the maturity of the product and the customer affected.

When new product introductions take place the management has clear benchmarks (time to resolve, case criticality, number of cases per month) which take into account number of systems in installation, ramp up and steady operation correlated to the number of cases from the field to different support levels. These indicators are part of the R&D team's compensation schema. In addition to these measures and incentives the continuous improvement – *kaizen* – mind set was promoted to drive towards zero R&D cases and fixed throughput and response times for each support and criticality level, which are measured, reported and used as part of the team's compensation plan.

As for the quantitative results and implications, the collection, classification and analysis of the request flow took 3 months. The newly defined customer experience metrics were

added to the standard global management team agenda in 2 months. Every unit in support and R&D were required to report monthly progress according to a separate dashboard. The main result achieved was the reduction in the overall throughput time from the opening to the closing of the request. In one year the reduction was about 50% in all severity classes (Figure 6). The overall reduction in request handling lead times was 19 days, i.e. from the initial 38 to 19 days. The number of open requests was also reduced proportionally to lead time. What does this mean in reality? This means that support people may focus on fewer requests at the same time, and in all they have released capacity to take on more requests and development tasks, and perhaps even to improve their competences through additional training. It is difficult to assess the cost reduction implications of this improvement due to skill and location mix in the global business, yet it is obvious that all employees involved now have more time for more value adding duties. The idea of classifying and prioritizing requests by value for the customer and the effort needed to provide the support proved to be difficult. More analysis is needed to define simple enough rules for the call desk to correctly classify the incoming requests.

Insert Figure 6 somewhere here

Due to the complexity of the system with its different versions running on different hardware and operating systems, customers have started more and more to push the system management responsibility to suppliers due to the complexity and expertise required to operate the system efficiently. One of the latest trends in customer relationships has led the case company to take over the operational responsibilities of the customer's plant, i.e. the company not only develops, supports, delivers and maintains the process control system, but actually runs the plants on a daily basis for the customer according to predefined performance guarantees. This makes ease of operation and maintenance of the system directly business critical for the company since the inefficiencies directly impact the company's bottom line.

When adding new measurements the organization always reacts nervously due to the increased oversight and control. Therefore it was important to motivate people working in different support and R&D organizations - both in developed and emerging countries – to

understand that the new measurements are a good tool to increase customer satisfaction not a tool for increased finger pointing. Yet, the relatively significant effort and top management involvement in the project showed that when adequate resources on a specific topic are placed the organization shows its capability to change. This means that numerous improvement efforts die in vain over time as their momentum runs out due to inadequate top management support and investment. Table 4 summarizes the case learning outcomes by linking with the critical factors discovered in the literature review (Table 1).

Insert Table 4 somewhere here

Following the theoretical framework set by Meijboom and Houtepen (2002) for international service operations along with their case in IT-sector, our results support their findings and underlying model. The more diverse and global is the service offering, the more important is the need to distinguish between services and related operations that have to take place either locally in close proximity of the end-customer, or in centralized manner in strategic global locations. Our case amends to this by showing that the service offering evolves through acquisitions, outsourcing and product development, and that this process of allocating operations and improving their productivity has to be under constant scrutiny. Theory of production and operations management with its lead time based principles complements well the existing performance metrics used in distributed development and support operations, thus helping management to maintain and improve continuously competitiveness in their operations.

7. Conclusions

Essentially the above described change project is very similar to any productivity improvement effort normally seen in manufacturing industries, where focus often is on reducing inventories through better flow and faster and less variable lead times. Similar time based performance metrics that are used in manufacturing can also be used for any workflow management system. Time is a critical component of any value adding process and therefore it should form one of the key metrics used to control and improve its performance. The present case and its follow-up customer questionnaire showed that customers not only appreciated faster response times, but also that they were

predictable, i.e. variation was reduced. The request throughput of the whole support organization increased while resources remained the same, thus the capacity of the existing organization increased without investments.

In the complex distributed support and development environment described in this case, the full end-to-end visibility on the source of the requests and full resolution lead times as experienced by the client are needed to provide a comprehensive view of the true performance of the organization. The initial analysis set the benchmark for the management responsible for improving quality and customer experience. As such, uniform, easily understandable and well communicated lead time measurement on a global scale seems to lead positive change when this is followed regularly by upper management. An additional benefit provided by uniform measurement is that it not only makes comparison and improvement tracking easier, but it also requires a uniform application of common tools and processes to be implemented. This helps to institutionalize the change and makes adaptation to other parts of the organization easier.

Key performance indicators are important to institutionalize the change in the organization. Our case shows that the KPIs are actually similar be they financial, logistics, or software process related. In financial management orders, revenue, outstanding sales, profit margins and various balance sheet ratios turn into logistical KPIs as forecast accuracy, actual shipments, punctuality, operational cost and inventory levels. We may translate these KPIs to complement software process measurement by tracking new tickets and change requests, what is the lead time to solve requests and the amount of open requests, i.e. work-in-progress, delayed requests and cost per request. Like in material flow management focus should be put on problematic requests that show long throughput times and higher cost due to more work. If bottlenecks are found one should locate the low performing unit and if necessary trace the problem even to the level of individual performer. The aim is establish a continuous improvement mentality both for the software development process and customer request process. These metrics and the fact based process monitoring ensure over the time the improvement of software quality and customer experience.

The case study shows that reducing request handling times improves productivity and customer satisfaction, but the link to actual software quality remains to be studied in more detail. The assumption that, if an organization is able to treat software requests faster, it

would also improve the software quality, cannot be established directly from this case study. However, operations management studies in manufacturing industries have been able to establish the positive link between operational speed and quality, which should also be valid in any value adding process including software development and support processes.

There are also other issues that emerged during the project that could make interesting research topics for the future. The general issue related to the support of multiple versions at the same time and how the outsourcing part of the support tasks should be organized remains to be studied in more detail. Another future research topic is related to the management and archiving of the tacit customer support knowledge that accumulates gradually on various systems. This special know-how on individual customers, system versions and their interrelations seems to be critical when decisions are made at each support level.

8. References

Agrawal, M., Chari, K. (2007), "Software Effort, Quality, and Cycle Time: A Study of CMM Level 5 Projects", IEEE Transaction in Software Engineering, Vol. 33, No. 3, pp. 145-156.

Argyris C., Putnam, R. and Mc Laine-Smith D. (1985), Action science, San Francisco, Jossey-Bass.

Baddoo, N., Hall, T. (2002), "Software process improvement motivators: An analysis using multidimensional scaling", Empirical Software Engineering, Vol. 7, No. 2, pp. 93-114.

Bartezzaghi, E., Spina, G., Verganti, R. (1994), "Lead-time models of business processes", International Journal of Operations & Production Management, Vol. 14, No. 5, pp. 5-20.

Bharadwaj , A.S., (2000), "A Resource-Based Perspective on Information Technology Capability and Firm Performance: An Empirical Investigation", MIS Quarterly, Vol. 24, No. 1, pp. 169-196.

Boutellier, R., Gassmann, O., Macho, H., Roux, M. (1998), "Management of dispersed product development teams: the role of information technologies", R&D Management, Vol. 28, No. 1, pp. 13-25.

Caldwell, B. (2002), Outsourcing Cost Reduction Creates Paradox: How to Still Make a Profit. Gartner Dataquest Report, Stamford, CT, April 12.

Cataldo, M, Nambiar, S. (2012), "The impact of geographic distribution and the nature of technical coupling on the quality of global software development projects", Journal of Software: Evolution and Process, Vol. 24, No. 2, pp. 153-168.

Cavaye, A.L.M. (1997), "Case study research: a multi-faceted research approach for IS", Information Systems Journal, Vol. 6, No. 3, pp. 227-242.

Chapin, N., Hale, J.E., Khan, K., Ramil, J.F., Tan, W.-G. (2001), "Types of software evolution and software maintenance", Journal of Software Maintenance and Evolution: Research and Practice, Vol. 13, No. 1, pp. 3-30.

Coghlan, D., Brannick, T. (2001), Doing Action Research in Your Own Organization, London, Sage Publications.

da Silva, F. Q. B., Prikladnicki, R., França, A. C. C., Monteiro, C. V. F., Costa, C. and Rocha, R. (2012), An evidence-based model of distributed software development project management: results from a systematic mapping study, Journal of Software: Evolution and Process, Vol. 24, No. 6, pp. 625–642.

Edgell, J., Meister, G.E., Stamp, N. (2008), "Global outsourcing trends in 2008", Strategic Outsourcing: An International Journal, Vol. 1, No. 2, pp. 173-180.

Ein-Dor, P., Segev, E. (1982), "Organizational Context and MIS Structure: Some Empirical Evidence", MIS Quarterly, Vol. 6, No. 3, pp. 55-68.

Eisenhardt, K.M. (1989), "Building Theories from Case Study Research", Academy of Management Review, Vol. 14, No. 4, pp. 532-550.

Gummesson, E. (1991), Qualitative Methods in Management Research. London, Sage Publications.

Hall, T., Wilson, D. (1998), "Perceptions of software quality: a pilot study", Software Quality Journal, Vol. 7, pp. 67–75.

Harter, D.E., Krishnan, M.S., Slaughter, S.A. (2000), "Effects of Process Maturity on Quality, Cycle Time and Effort in Software Product Development", Management Science, Vol. 46, No. 4, pp. 451-466.

Holmström, J., Ketokivi, M., Hameri, A.-P. (2009), "Operations Management as a problem-solving discipline: A design science approach", Decision Sciences, Vol. 40, No. 1, pp. 65-87.

Issac, G., Rajendran, C., Anantharaman, R.N. (2006), "An instrument for the measurement of customer perceptions of quality management in the software industry: An empirical study in India", Software Quality Journal, Vol. 14, pp. 291–308.

Jalote, P. (2000), CMM in Practice: Processes for Executing Software Projects at Infosys. Addison Wesley Longman.

Jalali, S., Wohlin, C. (2012), Global software engineering and agile practices: a systematic review, Journal of Software Evolution and Process, Vol. 24, No.8. pp. 643–659.

Kan, S.H. (2002), Metrics and Models in Software Quality Engineering. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Kettinger, W. J., Teng, J. T. (1997), "Business process change: a study of methodologies, techniques, and tools", MIS Quarterly, Vol. 21, No. 1, pp. 55-80.

Kuechler, W. and Vaishnavi, V. (2008), "The emergence of design research in information systems in north America", Journal of Design Research, Vol. 7, No. 1, pp. 1–16.

Kueng, P. (2000), "Process performance measurement system: a tool to support process-based organizations", Total Quality Management, Vol. 11 No. 1, pp. 67-85.

Lee, A.S. (1989), "A scientific methodology for MIS case studies", MIS Quarterly, Vol. 13, No. 1, pp. 33-50.

McDermid, J., Bennet, K. (1999), "Software engineering research: a critical appraisal", IEE Proceedings on Software Engineering, Vol. 146, No. 4, pp. 179–186.

Meijboom, B., Houtepen, M. (2002), "Structuring international service operations – A theoretical framework and a case study in the IT-sector, International Journal of Operations and Production Management, Vol. 22, No. 8, pp. 824-842.

Moses, J. (2009), "Should we try to measure software quality attributes directly?", Software Quality Journal, Vol. 17, pp. 203-213.

Myers, M. (1997), "Qualitative Research in Information Systems", MIS Quarterly, Vol. 21, No. 2, pp. 241-242.

Napier, N.P., Kim, J., Mathiassen, L. (2008), "Software Process Re-engineering: A Model and its Application to an Industrial Case Study", Software Process: Improvement and Practice, Vol. 13, No. 5, pp. 451-471.

Niazi, M., Wilson, D., Zowghi, D. (2005), "A maturity model for the implementation of software process improvement: an empirical study", Journal of Systems and Software, Vol. 74, No. 2, pp. 155-172.

Pettersson, F., Ivarsson, M., Gorschek, T. Öhman, P. (2008), "A practitioner's guide to light weight software process assessment and improvement planning", Journal of Systems and Software, Vol. 81, No. 6, pp. 972-995.

Piri, A., Niinimäki, T. and Lassenius, C. (2012), Fear and distrust in global software engineering projects. Journal of Software Evolution and Process, Vol. 24, No. 2, pp. 185–205.

Prikladnicki, R. (2012), Propinquity in global software engineering: examining perceived distance in globally distributed project teams. Journal of Software Evolution and Process, Vol. 24, No. 2, pp. 119–137.

Raffo, D.M. (2005), "Software project management using PROMPT: A hybrid metrics , modelling, and utility framework", Information and Software Technology, Vol. 47, No. 15, pp. 1009-1017.

Rainer, A., Hall, T. (2002), "Key success factors for implementing software process improvement: a maturity-based analysis", Journal of Systems and Software, Vol. 62, No. 2, pp. 71–84.

Ravichandran, T., Rai, A. (2000), "Total quality management in information systems development: Key constructs and relationships", Journal of MIS, Vol. 16, No. 3, pp. 119 - 155.

Runeson, P., Höst, M. (2009), "Guidelines for conducting and reporting case study research in software engineering", Empirical Software Engineering, Vol. 14, No. 2, pp. 131-164.

Sambamurthy, V., Zmud, R.W. (1999), "Arrangements for information technology governance: A theory of multiple contingencies", MIS Quarterly, Vol. 23, No. 2, pp. 261-290.

Schmenner, R.W. (2001), "Looking ahead by looking back: Swift, even flow in the history of manufacturing", Production and Operations Management, Vol. 10, No. 1, pp. 87-96.

Sein, M., Henfridsson, O., Purao, S., Rossi, M., and Lindgren, R. (2011), "Action Design Research", MIS Quarterly, Vol. 35, No. 1, pp. 37-56.

Siakas, K.V., Balstrup, B. (2006), "Software outsourcing quality achieved by global virtual collaboration", Software Process: Improvement and Practice, Vol. 11, No. 3, pp. 319–328.

Šmite, D., Wohlin, C., Gorschek, T., Feldt, R. (2010), "Empirical evidence in global software engineering: a systematic review", Empirical Software Engineering, Vol. 15, No. 1, pp. 91-118.

Šmite, D., Wohlin, C. (2012), "Lessons learned from transferring software products to India", Journal of Software: Evolution and Process, Vol. 24, No. 6, pp. 605–623.

Sousa, R., Voss, C.A. (2009), "The effects of service failures and recovery on customer loyalty in e-services", International Journal of Operations and Production Management, Vol. 29, No. 8, pp. 834-864.

Standish Group, (1995), Chaos—the state of the software industry, Standish group international, Technical Report, pp. 1–11.

Stuart, I., McCutcheon, D., Handfield, R., McLachlin, R., Samson, D. (2002), "Effective case research in operations management: a process perspective", Journal of Operations Management, Vol. 20, No. 5, pp. 419-433.

Subramanian G.H., Jiang J.J., Klein G. (2007), "Software quality and IS project performance improvements from software development process maturity and IS implementation strategies", Journal of Systems and Software, Vol. 80, No. 4, pp. 616-627.

Taxén, L. (2006), "An integration centric approach for the coordination of distributed software development projects", Information and Software Technology, Vol. 48, No. 9, pp. 767-780.

Vitharana, P., Mone, M.A. (2008), "Measuring Critical Factors of Software Quality Management: Development and Validation of an Instrument", Information Resources Management Journal, Vol. 21, No. 2, pp. 18-37.

Watson, R. T., Pitt, L. F., and Kavan, C. B. (1998), "Measuring information systems service quality: lessons from two longitudinal case studies", MIS Quarterly, Vol. 22, No. 1, pp. 61-79.

Yang, Y.H., (2001), "Software quality management and ISO 9000 implementation", Industrial Management & Data Systems, Vol. 101, No. 7, pp. 329-338.

Yin, R.K. (1994), Case Study Research – Design and Methods, California, Sage Publications.

Critical factor	Key elements
Software process control and its continuous improvement	Continuous software support improvement process, process performance measurement and monitoring including business management; customer satisfaction measurement
Cross-border and intercontinental workflows	Global process visibility and responsibility; end-to-end process performance measurement and continuous improvement processes
Different computer systems and protocols inherited over time	Lack of transparency of the process performance, difficult to follow customer ticket/request from initiation to closing; investment and implementation effort needed to streamline the systems
Collaboration with off-shored units	Language barriers, culture differences, working habit and process differences, creating common team spirit, clarifying responsibilities, different compensation and motivation approaches
Increasingly demanding customers with their unique requests	Customer specific modules, code variants, single or multiple source code system

Table 1. The critical factors related to global software process management and improvement.

Region	Trained support personnel			Installed Systems per Employee		
	L1/L2	L3	L4	L1/L2	L3	L4
1	80	25	20	12	37	46
2	100	21	10	18	85	178
3	108	17	10	10	65	111
4	160	5	0	5	146	N/A
	448	**68**	**40**	**11**	**83**	**112**
	Total			Average		

Table 2. Multitier support organization scattered in 4 global regions and their responsibilities.

Support level	Description
L1	Local language and English call logging and simple (e.g. password) problem resolution
L2	Primarily English problem resolution by trained personnel needing no configuration changes
L3	English only problem resolution by experienced experts needing no source code changes
L4	English only problem resolution requiring source code change leading possibly to a customer specific or general patch or even a new release

Table 3. Support level definitions.

Issue	Proposed solution
Software process control and its continuous improvement	Implement continuous process control metrics, educate people to follow the metrics; link the improvement process with the overall quality management processes of the company; establish regular meeting and review routines
Cross-border and intercontinental workflows	Global process responsibility; process performance measurement like in logistics i.e. information logistics; implement continuous improvement process based on the data from the measurement system
Different computer systems and protocols inherited over time	Either implement global common ticketing/ request monitoring system or as the case company implemented a data warehouse for software support and change request quality
Collaboration with off-shored units	Implement supporting processes including employee circulation between on and off-shore locations; regular video and conference calls between units and global process responsible
Increasingly demanding customers with their unique requests	Improve reporting of customer requests and involve global management in the process; follow number of requests coming by complexity of the request; throughput time; follow-up directly with customer the longest throughput time cases; invest time in learning from the fast solved requests if something good could be generalized

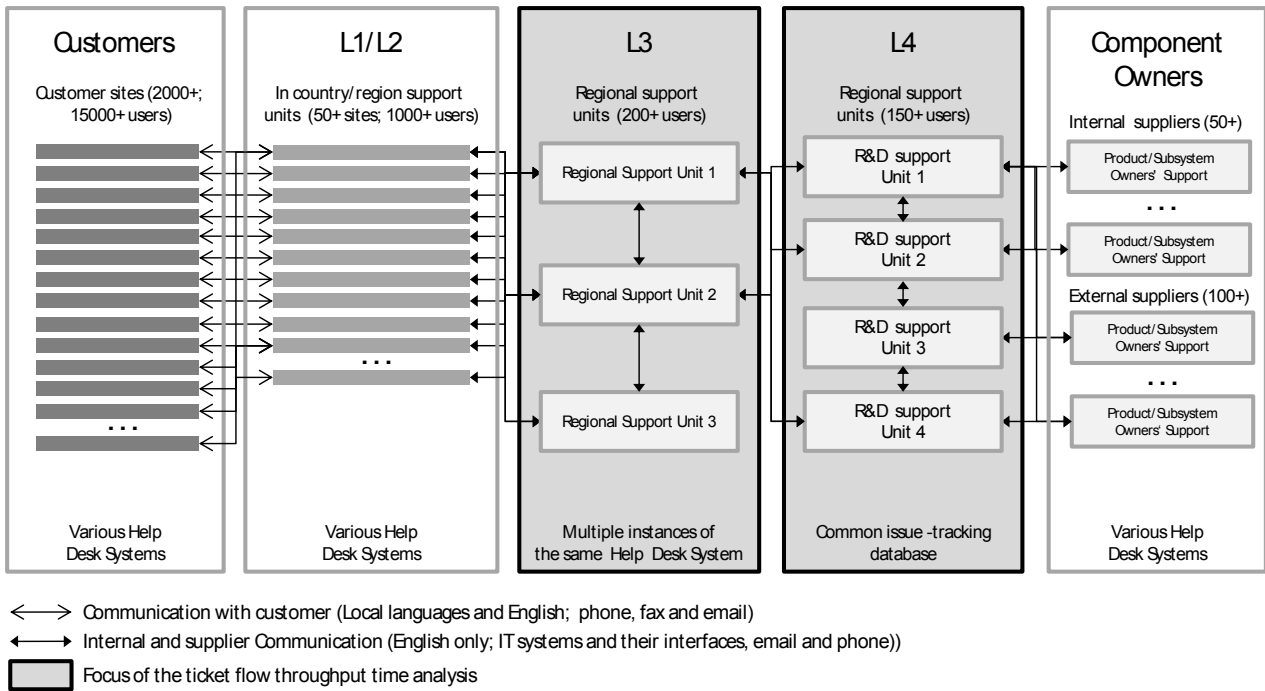Table 4. The challenges and proposed solutions related to global software process improvement

Figure 1. End-to-end overview of the customer request resolution process. The highlighted levels L3 and L4 form the focus of the analysis.
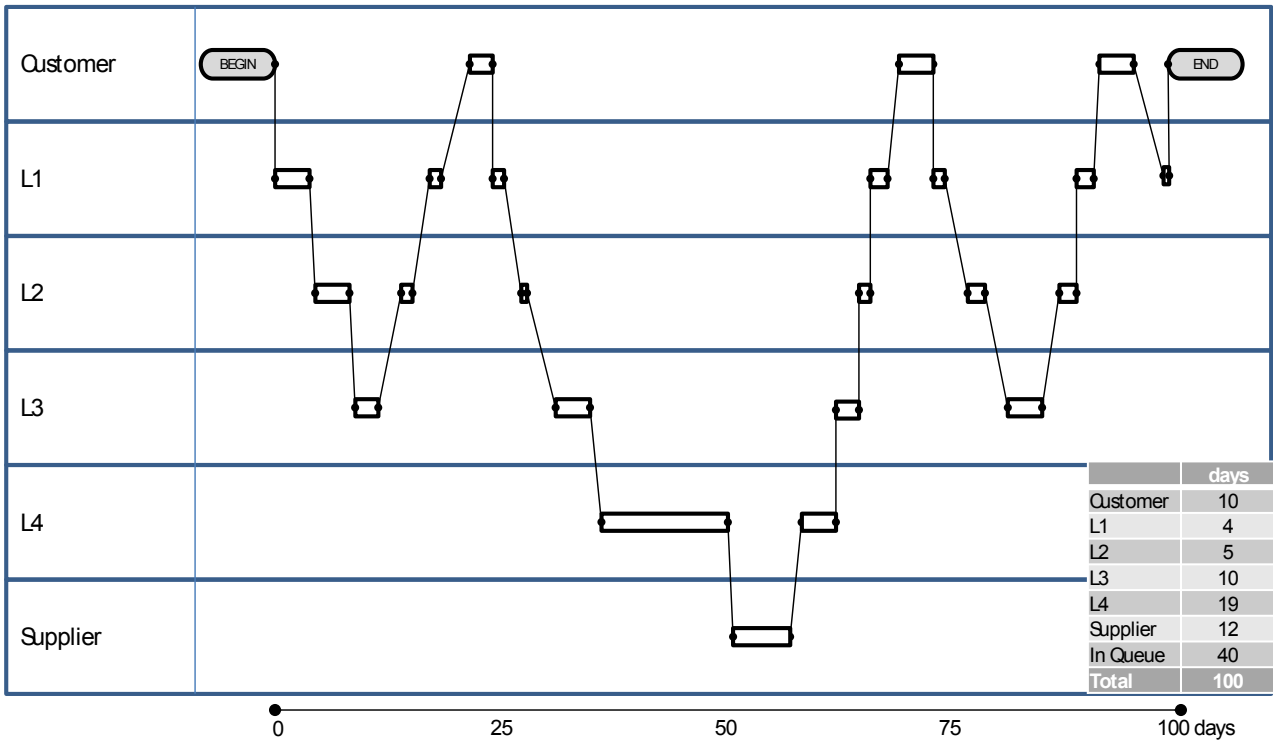
Figure 2. Example of a low performance (multi-hop) issue resolution flow requiring a source code change. Overall lead time is the sum of all lead times at each support level.
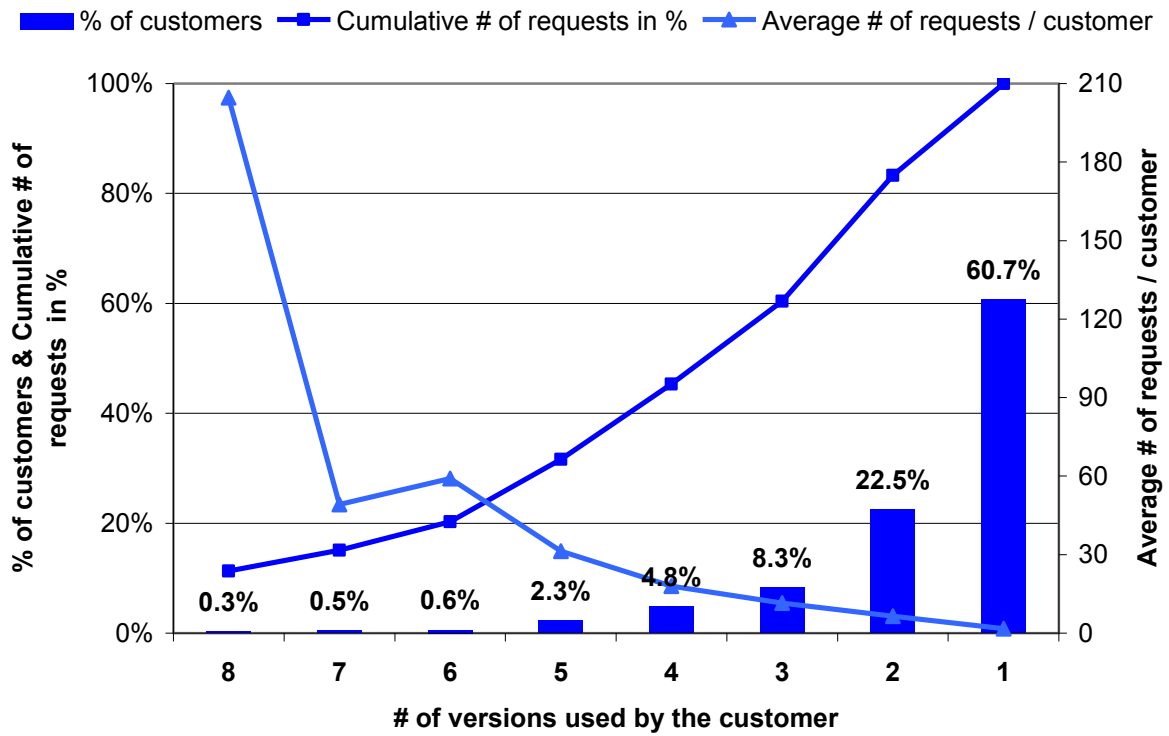
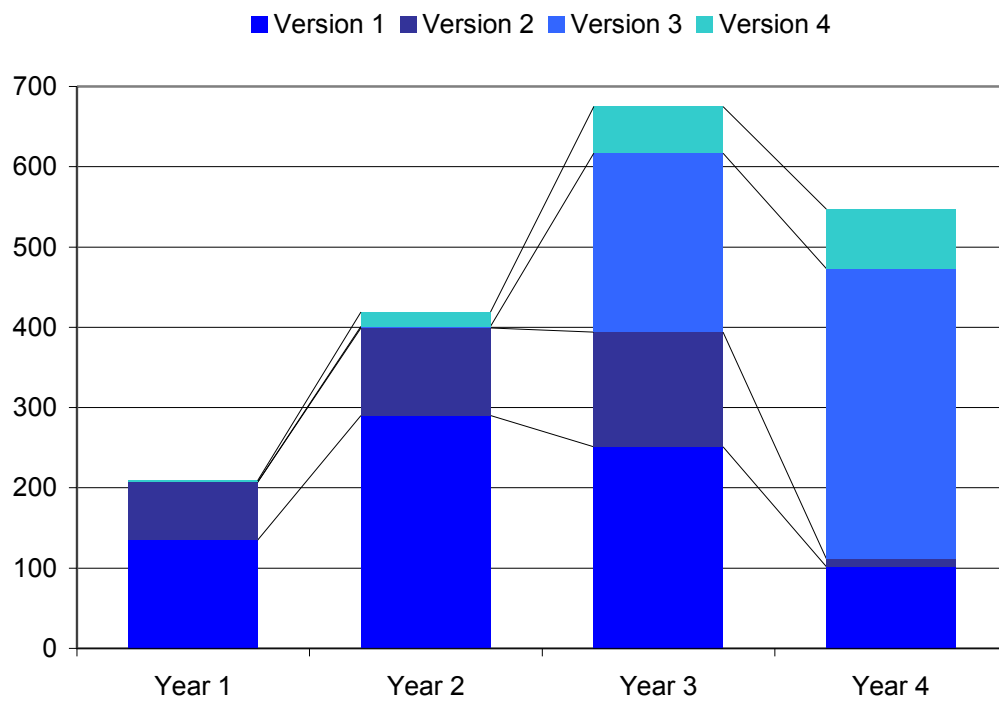Figure 3. Number of different versions used by the customer with their average and cumulative service requests.

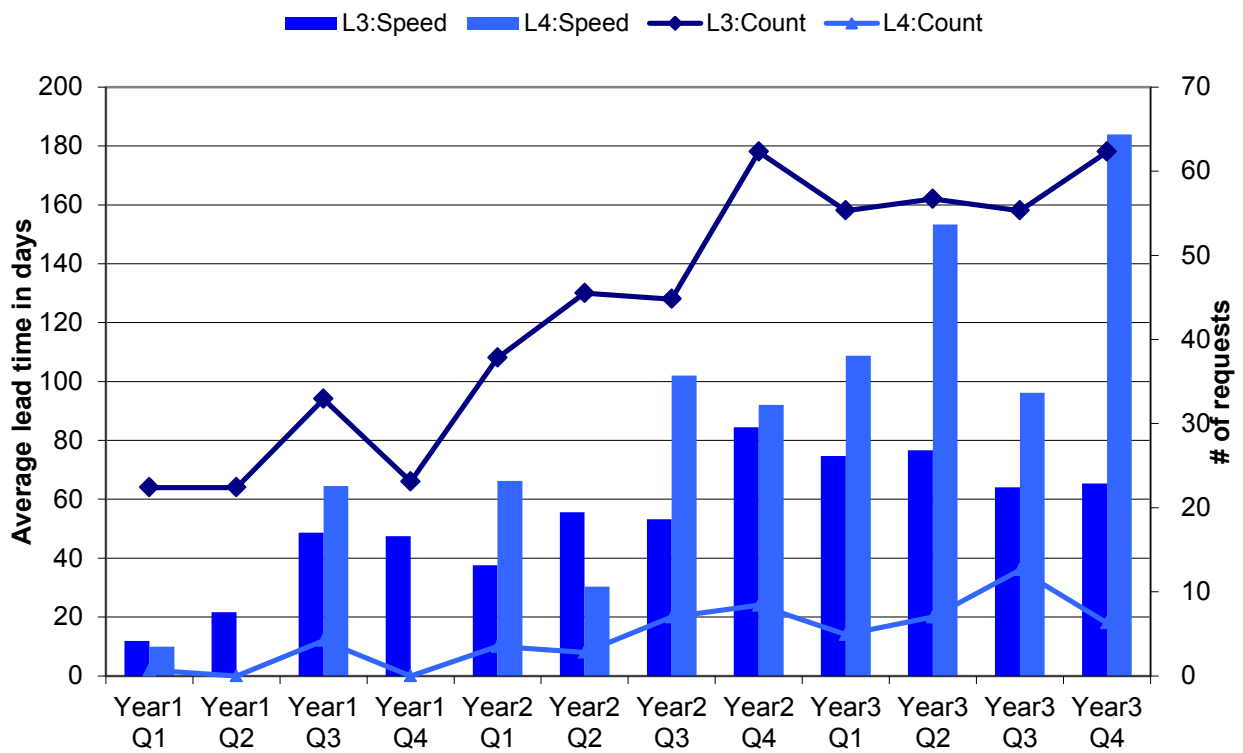Figure 4. Number of requests per software version during the sample period.

Figure 5. Quarterly average request handling lead times in days and request count at levels 3 and 4.
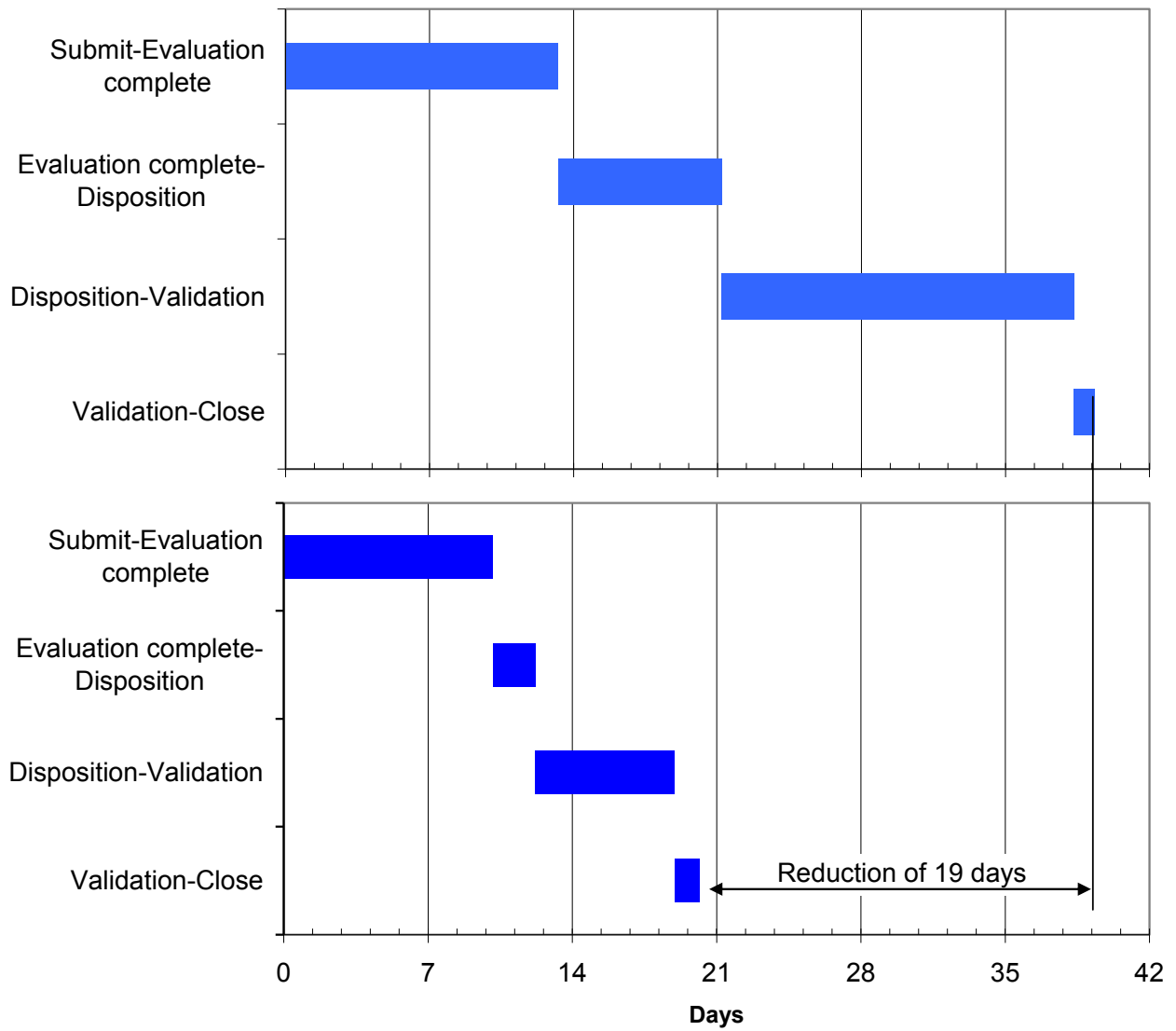
Figure 6. Overall lead time reduction in the processing of the customer requests.