Evolution of a Software Maintenance Organization from Cost Center to Service Center

DOI:
10.1109/ICSM.2003.1235423
http://dx.doi.org/10.1109/ICSM.2003.1235423

Document status and date:
Published: 01/01/2003

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Evolution of a Software Maintenance Organization from Cost Center to Service Center

Sander Smit, Peter H.N. de With, Gert-Jan van Dijk
LogicaCMG, EHAT1, P.O. Box 7089, 5605 JB Eindhoven, The Netherlands.
s.smit@logicacmg.com

Abstract

This paper describes experiences with the evolution of a software maintenance organization for digital set-top boxes of a leading electronics company from a cost center towards a service center. Several years ago a dedicated software maintenance group was constituted. As the costs for software maintenance were not recovered from the customers, the software maintenance group was merely considered a cost center. Through starting a metrics program for software maintenance and defining a service strategy with various service levels, the software maintenance group generated sufficient revenues to become self-supporting. An important conclusion is that the use of ITIL service support has helped to develop a better customer focused approach, which is considered as the most important critical success factor for a professional, self-supporting maintenance organization.

Keywords: software maintenance, practice, software process improvement, service center, organization, management

1. Introduction

This paper presents the successful transition of a software maintenance organization from a cost center towards a service center. At the end of the nineties, a large electronics company finalized the development of their first digital set-top box. This device enabled consumers to watch digital television while employing their existing analog TV via an externally connected digital decoder (set-top box). The customers consisted of television providers who where starting to offer digital television services to their subscribers. The software stack involved in the digital set-top box consisted of an estimated 500K lines of source code. Through the implementation of resident loader software, it was possible to upgrade this software stack via the network (cable or satellite) from the provider. To prevent interruptions in the development process of the engineering department, the company management decided to establish a dedicated software maintenance team. The objective of this team was to solve problems reported by the customers and it was agreed that only in the case of software problems having a large impact on the customer's business, the development team would be involved in removing those problems. The issues involved in software maintenance are well described in [1]. Hardware problems of the product were handled directly by the factory, while the software maintenance team was not initially involved in the (hardware) problem solving.

The software maintenance team became gradually mature but faced internal criticism as they were considered as a cost center inside the company. Gradually, the value of the cost center was discovered through an increased recognition by product management and customers. This resulted in gradually charging the customers for problem resolving so that the maintenance organization further augmented in professionalism. Additionally, installing special tools such as the ITIL [2] resulted in clear improvements. Section 2 explains the various evolution stages of the software maintenance group and highlights the most important learning aspects. Section 3 indicates the special improvement measures that were taken to professionalize towards a service center. In Section 4, we summarize the conclusions and address possible future improvements.

2. Evolution stages

The growth and development of the software maintenance team towards a mature service organization is not the result of a well-defined plan, but merely the outcome of an evolutionary process. Key drivers in this process are the will to improve and believe in the value of the team for the organization. Despite the lack of an initial development plan, four stages in this process can be distinguished (see Figure 1).

- Stage 0, *Initial*: Start of maintenance activities by a group of software engineers.

1063-6773/03 $17.00 © 2003 IEEE

- Stage 1, *Management awareness*: Provide insight in the cost, the value and the potential of the software maintenance activities to the rest of the organization.
- Stage 2, *Internal service agreements*: Defining a service strategy, services and service levels.
- Stage 3, *External service agreements*: Offer services to customers on a level matching optimally with their business.

Besides these stages, software process improvement measures were applied which are discussed later.

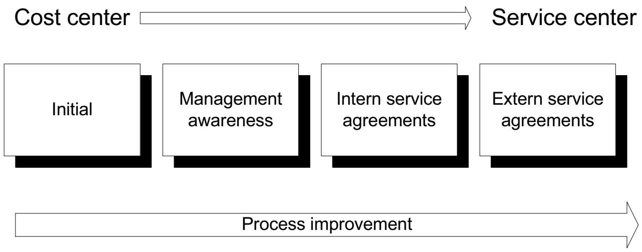

Figure 1. The four stages of evolution.

2.1 Stage 0: Initial

Initially, the software maintenance team started as a group of software and test engineers, recruited from the development department. They had technical knowledge of the released product and maintenance was established. Two problems became apparent. Firstly, new hardware platforms and new software were developed over time, leading to more products. The maintenance team grew in size, because more products had to be supported. This resulted in increasingly difficult budget discussions at the end of each year. Secondly, also the customers became more demanding. Since no clear agreements about the level of support existed, customers expected full support at 24 hours a day. Hence, the position of the maintenance team was squeezed from two sides, so that evolution to another stage of maturity was required.

The lessons learned from this stage are that (1) the company organization asked for cost control in this stage and (2) the expectations to the customers about the support level should have been managed more clearly.

2.2 Stage 1: Management awareness

In this stage, the team started to provide the management with information about the activities of the team. Additionally, metric measurements were established in order to monitor the operation of services that were supported. Most important metrics were: (1) the number of reported problems per product, (2) the number of filed change requests per product, (3) the number of maintenance releases per product, (4) the effect turn around time and (5) the involved effort per customer. These measurements enabled management insight in the maintenance activities and performed tasks per customer in a consistent way.

Prior to the use of metrics, the customer calls for support were considered problems. The measurements showed that 34% of the reported problems were actually change requests. It also showed that one particular demanding customer claimed most of the resources. Sales and account managers acknowledged soon that the organization was providing valuable service without obtaining revenue in return, so that the management agreed to start investigating a business model for software support.

The lessons learned in this stage are that management data is required to initiate actions and have insight in the cost breakdown. Also the sales team has to communicate with the maintenance team in order to incorporate maintenance costs in the total business model.

2.3 Stage 2: Internal service agreements

In this stage, sales managers started charging customers for maintenance releases, which created awareness at the customer side. The software maintenance team identified and defined the services delivered. This resulted in a description of services such as updates, service desk, and so on. Simultaneously, a service strategy was defined. The team discussed with the product management the level and contents of support services to be offered by the organization. The service strategy had to be in line with the business strategy. After establishing a global strategy, each individual customer was evaluated. Customers receiving free support for years were not directly confronted with a service contract. Instead, the team defined the level, the amount of free support, the strategic value and potential of each customer. This resulted in the definition of a level of service for every customer. The outcome was input for the annual budget discussions. Previously, the budget discussions were financially oriented and they were concentrating on satisfying development budgets. Gradually this changed to more content-driven discussions, where the individual needs of the customers were considered. The result was still a number of engineers to form the maintenance team, but the way towards this outcome was far more constructive.

Important lessons learned from this stage are the definition of services and level for individual customers. This resulted in a commercially relevant service strategy fitting to the customer size. This fitting ensured commercial justification of the maintenance costs.

Furthermore, the organization prepared itself for becoming a professional service organization.

2.4 Stage 3: External service agreements

The software maintenance team was now able to offer new customers a professional support package. A white paper describing the different service packages for a well-defined customer choice was included with every proposal. For existing customers, the service level was re-defined to the level determined at the service strategy meetings. Occasional requests for e.g. an extra maintenance release were charged on an individual basis.

The learning aspects of this stage are still under evaluation, because the maintenance team only recently reached this stage and feedback topics did not yet clearly occur. A preliminary conclusion is that the software maintenance activity is becoming an integral part of the business planning and product creation process.

3 Process improvement activities

The software maintenance team inherited all work practices from the software development organization. The development organization was CMM level-2 certified, and has currently CMM level 3. Most of the work processes described in the Product Creation Processes (PCP) could be re-used for maintenance. Particularly, the configuration management was of substantial value for the maintenance group. However, as the evolution from Section 2 proceeded, the work practices gradually transformed to a more efficient software maintenance process.

3.1 Stage 0: Initial

The team first described the processes that are required for maintenance and were not available from the existing PCP. The most important maintenance processes were a Software Transition Process and a Software Release Process. The Software Transition Process guides projects in the transfer of their deliverables to the maintenance organization. The release procedure from the PCP describes the commercial release for a new product. Since the scope of a maintenance release was limited to the software and a maintenance release involved only a limited amount of changes, a release procedure was needed with limited test procedures and shorter time-to-market. The result was a PCP extended with software maintenance-specific processes.

A learned lesson from this stage is that the scope of the PCP is too limited for software maintenance activities.

3.2 Stage 1: Management awareness

In this stage, the management received increasingly detailed information about the activities of the software maintenance team. A frequent question was why it took so long to solve a reported problem. Therefore, the team decided to focus on the efficiency of the processes. One of the results from the metric analysis (Section 2) was that the average analysis throughput time for a problem report was 40 days. Development of a detailed problem breakdown resulted in more control of the analysis process. In addition, the test and verification process was improved. Enhanced guidelines for code review and the delivery of a test plan for every resolved problem resulted in an earlier detection of errors.

The lessons learned are the relevance of performing a structured analysis procedure, and secondly, the usage of a test process aiming at finding errors at the earliest possible state.

3.3 Stage 2: Internal service agreements

The software maintenance team now felt confident about the handling of problem reports, change requests and releases. The focus began to move from the maintained products towards the customers using those products. Instead of writing a new set of processes, the team verified the deviation from the processes as described in IT Infrastructure Library (ITIL). An ITIL self-assessment covering the Service Support Processes was performed. Conclusions from this assessment were as follows:

1. The ITIL process model was useful for the software maintenance team.
2. The gap with ITIL was not as big as expected.
3. The Service Desk Process was poorly implemented (see Figure 2).

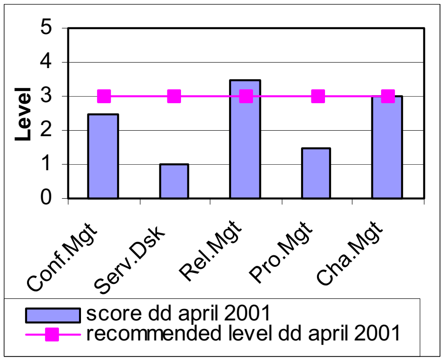

Figure 2 First ITIL self-assessment score.

To improve the Service Desk Process, a call-tracking tool was implemented, a dedicated Service Engineer was assigned, and targets for response times were defined.

The lesson learned is that ITIL is a good process model for software maintenance activities. ITIL appeared to be closer to the Product Creation Process as expected: about 80% of the issues were covered by ITIL.

3.4 Stage 3: External service agreements

Finally, the software maintenance team could rely on processes describing the software maintenance and the service support activities. The implementation of the service delivery processes from ITIL was initiated, starting with the service level management process. Simultaneously, all processes and procedures were organized according to the ITIL model. For example, the Software Transition Process became part of the Change Management Process and the Software Release Process was integrated into the Software Control and Distribution Process. The learning aspects of this stage are still under evaluation.

4 Conclusions and future improvements

We have described how a software maintenance team gradually evolved and became an integral part of the Product Creation Process. It was shown that the transition process can be visualized in four stages: Initial, Management Awareness, Internal Service Agreements and External Service Agreements. Similarly, in parallel the software improvement process went through comparable transition phases, which are outlined separately.

The most relevant lessons from the evolution of the software maintenance organization, are summarized with the following issues:

- *Increased customer focus*: the use of ITIL service support has helped to develop a better customer focused approach, which is considered the most important critical success factor for a professional, self-supporting maintenance organization.
- *Collect metrics*: the selected metrics were crucial in obtaining the necessary management awareness to increase the professionalism of the software maintenance organization.
- *Involve sales and marketing managers*: as software maintenance had always been considered a cost center, the involvement of stakeholders for sales and marketing facilitated the development of a profitable business strategy for self-supporting software maintenance.

For the software maintenance process and business, future improvements should address the following issues. Firstly, increased customer intimacy would further strengthen the customer relationship. Additional business will be developed resulting from better insight in customer demands. Secondly, the maintenance of the system can be more emphasized during the design phase of the system. This is obtained by explicitly defining requirements for software maintenance in the early stages of the development life cycle and through involvement of software maintenance engineers during software development. Consequently, the software maintenance organization will become more efficient, thereby increasing the competitive advantage. Thirdly, the scope of activities can be extended in the area of TV and DVD software. This will increase the necessary critical mass of the maintenance organization. Furthermore, risks will be further spread, which will increase stability and viability of the maintenance organization.

The presented concept and stages can be re-used for developing connected consumer electronics devices in a business-to-business oriented market. The aspects such as cost, awareness and service agreements, are relevant to all cases where software maintenance activities are performed.

References

[1] Thomas M. Pigoski, "*Practical Software Maintenance*", Wiley Computer Publishing, 1996.
[2] CCTA, "IT Infrastructure Library", 1993, (see http://www.itil.co.uk).