Chapter 7.13
Integrated Product Life Cycle Management for Software:
CMMI[1], SPICE, and ISO/IEC 20000

Dirk Malzahn
OrgaTech GmbH, Germany

ABSTRACT

This chapter describes how models for software development and service delivery can be integrated into a common approach to reach an integrated product life cycle for software. The models covered by this chapter are the capability maturity model integration (CMMI), SPICE (software process improvement and capability determination, ISO 15504) and ISO 20000 (service management). Whilst the CMMI constellation approach delivers an integration perspective defined in three models (development, acquisition and services), SPICE and ISO 20000 need additional alignment to be usable in an integrated approach.

INTRODUCTION

The focus of the market for IT solutions has changed. Whilst many companies and organizations followed the latest "hype" several years ago, they now trust in reliable and sustainable solutions.

To ensure this, standardization of quality evaluation becomes more and more important. For supplier selection, make-or-buy decisions and outsourcing strategies, a powerful set of procedures, that can help to assess the capability of internal and external software processes, is required. These procedures have to be based on best practices and must be widely accepted.

On this basis, standards offer the best possibilities: they are usually defined by a wide group of experts, which all contribute their experiences and best practices. Standards are either sponsored by an industry or by national bodies—therefore making these standards de facto mandatory for an industry, nation, or combination of both enforces the acceptance. If a significant group uses a standard, market dynamics have an additional impact. Official certificates, levels, and so forth can be and are used for marketing activities.

In the field of software related standards, lots of different standards have been defined for special topics, but one standard is still missing: a standard that covers a software product from the very beginning—the first idea—up to the very end—the retirement of the software.

On the one hand powerful standards, for example the capability maturity model integration (CMMI) or SPICE (ISO 15504), have been defined for software development. On the other hand, standards for service delivery, for example ITIL or ISO 20000, have been well established; but there is still a wall between the worlds of software development and service delivery. Even though some standards – like SPICE – take a look over the wall, an integrated approach has not been delivered yet.

The need for this integration is obvious. A customer is not interested in having some quality for development and some other quality for service delivery—the customer needs one quality approach that covers the full life cycle of a software product.

BACKGROUND

The Wall Between Software Development and Service Delivery

When IT systems are planned, the focus of the planning is mostly restricted to software development. Topics like operation environment or data management are discussed, but the strategy usually ends with the delivery of the software product.

On the other hand, service-delivering organizations mostly just provide "services" and are not really interested in the software development process.

This behavior leads to multiple difficulties and inefficiencies:

- Software developers and service people do not understand each other. They work in different worlds and have their own "language" and processes.
- The efficiency and effectiveness of service delivery highly depends on the architecture of and assumptions for the software, therefore the service organization has to be integrated early into the software development.
- Service level agreements can be optimized, when both sides reach a common understanding. The development of service level agreements is often based on the "what we need" position of both sides and not on the "what will be best for the customer" position.
- Problem Management is not transparent to the customer. The customer is not interested whether he has a service problem or a software problem—the customer wants a quick and reliable solution. If the software side does not understand the service side, problems often become ping-pong balls.
- Software usually lives longer than the original developer intends. Systems often have to be enhanced just to fulfil the requirements of a new service platform. If this is not taken into account when the software is developed, the effort for updating software may become enormous. Sometimes software has to be retired, just because it is not executable on the new platform!
- New approaches like service oriented architectures (SOA) demand the high integration of software and service elements. Future trends will rather lead to small combined software/service environments than to big software solutions operated by massive computer environments.

Just to ensure that I am not misunderstood: software developing and service delivering organizations will still deliver and operate solutions