



Software.Process.Management
software processes that fit your business needs

Establishing the Software Engineering Processes

Whitepaper



Whitepaper: Establishing the Software Engineering Processes

0701-1.0.0

June 2007

Software.Process.Management

Dr. Andreas Birk

Gutenbergstrasse 99

70197 Stuttgart

Germany

<http://www.swpm.de>

Copyright © 2007, Software.Process.Management, Dr. Andreas Birk

All rights reserved.



Abstract

Software project performance and product quality are often widely determined by the design of the software processes and by the degree at which the defined processes are actually established throughout the project or organization. This whitepaper describes the benefits of processes, gives an overview of software engineering life cycle processes, and explains how processes can be established efficiently.



Contents

What Is <i>Process</i> ?	5
Benefits of Establishing the Software Engineering Processes	6
Overview of Software Engineering Processes	7
Define, Establish, and Evolve the Software Engineering Processes	11
Other Processes that Interface with the Software Engineering Processes	13

What Is *Process*?

A widely accepted definition of *Process* in software engineering is¹:

A process is a set of interrelated activities, which transform inputs into outputs.

Accordingly, Software Engineering Process can be defined as:

A software engineering process is a process, which transforms business goals and requirements into operational software systems that meet the business goals and requirements.

A process serves a purpose, i.e., a high level objective of performing the process. It provides an outcome in the form of specific benefits to the process's stakeholders.

Processes are used to describe, think, and talk about what happens or is supposed to happen in software projects. They are an important tool for planning and management, ease collaboration within teams, and are the basis for continuous improvement and the identification of best practices.

A software process is usually performed by human individuals or groups, sometimes its execution is automated using some software solution.

Defined vs. actually performed processes

It is useful to distinguish between *defined processes* and *actually performed processes*:

The **defined processes** are the processes prescribed and documented in a project handbook, the project plan, job descriptions, or the like.

The **actually performed** processes are the processes that the project participants can be observed to perform, irrespective of what has been described in the process definitions.

Sometimes, the actually performed processes deviate significantly from the defined processes. There are various possible reasons for that, ranging from over-engineered and inflexible process definitions to insufficient process awareness throughout the project teams.

Software engineering management will usually be most effective and efficient, when (1) the process definitions are appropriate and (2) the projects actually operate according to these definitions.

A wide variety of software engineering processes

It has become common to speak of the *software engineering processes (in plural)* instead of the *process* (in singular), because a software project involves the performance of very many different processes. Each of these can be defined and performed in many different ways.

References

- ¹ International Standards Organization (ISO). *Quality management systems – Fundamentals and vocabulary (ISO 9000:2005)*, December 2005.

Whitepaper: Establishing the Software Engineering Processes

Copyright © 2007, Software.Process.Management, Dr. Andreas Birk



Benefits of Establishing the Software Engineering Processes

Having effective software engineering processes in place can be crucial for the success of a project or organization. The following list contains important benefits that can be achieved when software engineering processes are established:

- Processes enable effective project control and manageability
- Processes are an important tool for project planning and estimation
- Processes help coordinating the interaction between project participants and stakeholders
- Processes foster the quick integration of new team members
- Processes are a prerequisite for achieving higher levels of process maturity ratings
- Processes facilitate evolutionary improvement of software development practice
- Processes ease the fast adoption of new technologies and development approaches

Overview of Software Engineering Processes

There is a variety of different software engineering processes. Each of which providing its specific contribution to software development projects and organizations. It is useful to start the explanation of the software engineering processes with the core processes that create and deliver the technical solution to the identified business goals and requirements. The additional processes will be explained afterwards.

Development and Delivery of Technical Solution

The core processes of a software project are those that create and deliver the technical solution to the identified business goals and requirements. Figure 1 shows these life cycle processes of solution development and delivery. It is a prototypical process framework whose scope is single-product, on-site development of IT systems. Process frameworks for other project characteristics—such as product line development, embedded systems development, service-oriented architectures, or distributed off-shore development—require specific adaptations of the base model.

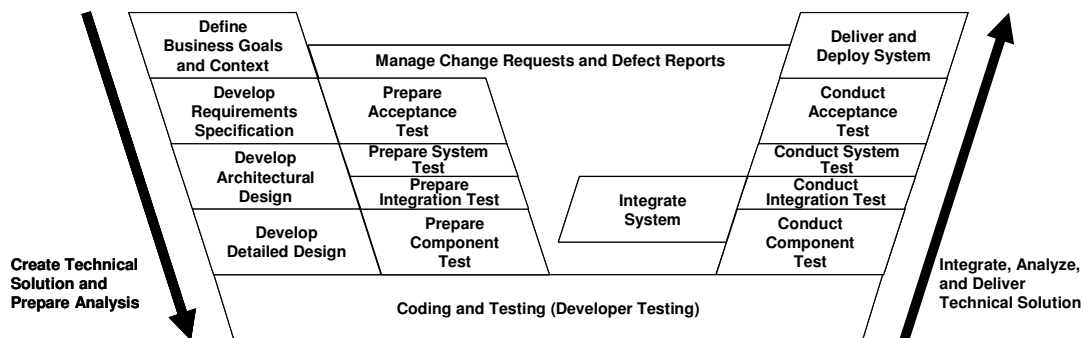


Figure 1: An overview of the processes for development and delivery of technical solution.

The processes can be grouped into three process threads and one additional cross-sectional process:

- (1) Creation of the technical solution,
- (2) Preparation of the analysis procedures (i.e., typically various kinds of testing), and
- (3) Integration, analysis, and delivery of the technical solution.

An additional process manages change requests and problem reports.

Creation of the technical solution includes the processes from definition of business goals and context via requirements specification and design to coding and (developer) testing.

Preparation of analysis procedures contains the preparation of acceptance, system, integration, and component tests. It is closely related to the creation of the technical solution. It builds on

the intermediate results of the technical solution, such as the requirements specification documents.

The third process thread covers the various testing processes, integration of the technical solution from its components, and delivery and deployment of the developed software system.

Change request and problem report management is a cross-sectional process, which is tightly interrelated with creation and delivery of the technical solution. It is triggered by every external and internal change request, involves problem reporting and resolution, and it interacts with possibly all other engineering processes.

Comprehensive Set of Process Categories

In addition to the development and delivery of technical solution described above, which is also called Engineering Process Group, software projects require a variety of other processes. Figure 2 presents an overview of these additional processes, grouped into several process categories.

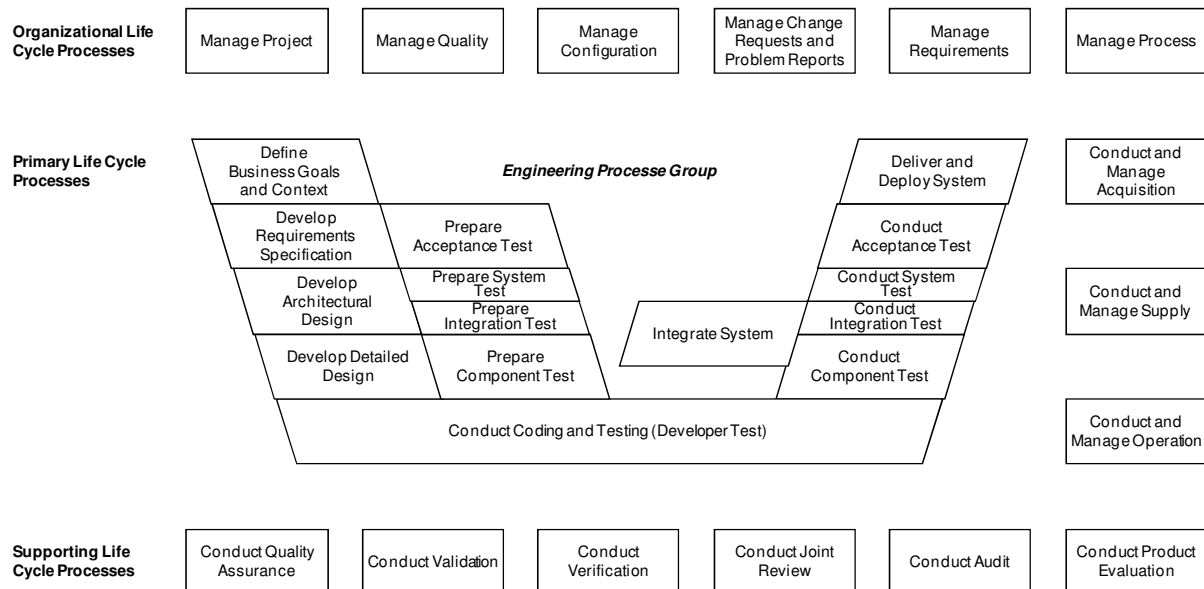


Figure 2: Extended set of software engineering processes including all process categories and selected organizational and supporting life cycle processes.

The process categories and process groups are¹:

- Organizational life cycle process category
- Primary life cycle process category
 - Engineering process group
 - Acquisition process group
 - Supply process group

- Operation process group
- Supporting life cycle process category

The presented process model is specifically designed to meet the various process requirements of industrial software projects. It combines elements from international standard ISO/IEC 12207¹, the Rational Unified Process (RUP)², the V-Model XT³, and the W-Model of software testing⁴. By blending these different models, the process model overcomes shortcomings of its source models. It provides effective guidance for defining and establishing a tailored process model for a specific software project or organization.

The particular strengths of the presented process model are:

- The processes contained in the process model can be mapped to clear milestones and associated tangible results, such as documents, development artefacts, and executable software code.
- The process model provides an operational framework for defining the work breakdown structure of a project.
- The process model establishes a basis for defining role and competence models.
- The process model places particular emphasis on effective testing, including the explicit test preparation and problem report management processes.
- The process model includes processes that establish important interfaces of software development. Among those are:
 - Business goals and requirements, which includes interfaces to project sponsors, product management, IT operations, and users.
 - System handover (delivery), which includes close interaction with IT operations.
 - Change request and problem report management, which involves interaction with product management, users, and IT operations.

Note that—despite the quite large number of processes—the presented model is not yet fully comprehensive. Several processes have been omitted in order to ease the initial understanding of the model. Most of those omitted processes belong to the organizational and supporting life cycle process categories. Also within the engineering process group, system development processes required for embedded systems development are not addressed.

However, the set of overall process categories in the presented model is complete, and the listed processes apply to virtually every kind of development project. This allows for effectively using the process model as a guide for project reviews, and for the development of tailored organization-specific process models.

References

- ^{1.} International Standards Organization (ISO) / International Electrotechnical Commission (IEC). Information technology — Software life cycle processes AMENDMENT 2 (ISO/IEC 12207:1995/Amd.1:2004(E)), May 2004.
- ^{2.} P. Kroll and Ph. Kruchten. The Rational Unified Process made easy: A practitioner's guide to the RUP. Addison Wesley, Boston, MA, 2003.
- ^{3.} V-Modell XT, Release 1.2 - Der Entwicklungsstandard für IT-Systeme des Bundes, <http://www.v-modell-xt.de>
- ^{4.} A. Spillner. *Management des Testprozesses von Anfang an - Das W-Modell*. In Th. Spitta, J. Borchers, H.M. Sneed (Eds.): *Proceedings of Software Management 2002 - Progress through Constancy*, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik (GI), Bonn, Germany, 2002, pp. 65—76.

Define, Establish, and Evolve the Software Engineering Processes

The effective implementation of software engineering processes in a specific project or organization will always involve the three steps of defining, establishing, and evolving the processes. Defining the software engineering processes includes transforming, tailoring, and fleshing out the processes of a general process life cycle model to the specific needs and constraints of the project or organization.

Establishing the processes aims at making them actual daily practice of the project teams. Process evolution is required, first, to improve and further mature the software engineering practices where needed, and second, to keep them up to date with changing context situations, like new development tools or organizational restructuring.

Define the Software Engineering Processes

Defining concrete software engineering processes for a specific project or organization can start with a comprehensive and well-balanced life cycle model like the one presented in the previous section.

Process definition requires tailoring such a generic process model so that it fits the specific goals and needs of the project or organization. It does not necessarily involve a formal definition of activities to be performed. Also alternative means of process definition, like job descriptions, milestone planning, or a certain organizational setup can be appropriate, too (cf. Figure 3).

Process definition involves the following decisions:

- Decide about the scope of the process model (i.e., whether it should be applicable to the entire software organization, to some specific parts of it, or to an individual project only)
- Decide about the process development strategy (i.e., either a top-down strategy that introduces new processes to the project or organization, or a bottom-up strategy that picks up current development practices and documents them explicitly).
- Decide which processes of the overall life cycle model shall be included into the new process model.
- Decide about the appropriate level of detail of each process's definition.
- Decide about the life cycle methodology to be applied for the process definition (i.e., sequential, incremental, or iterative development).
- Decide about how to document and how to make available the process definition (e.g., as process handbooks, work policies, or job descriptions).

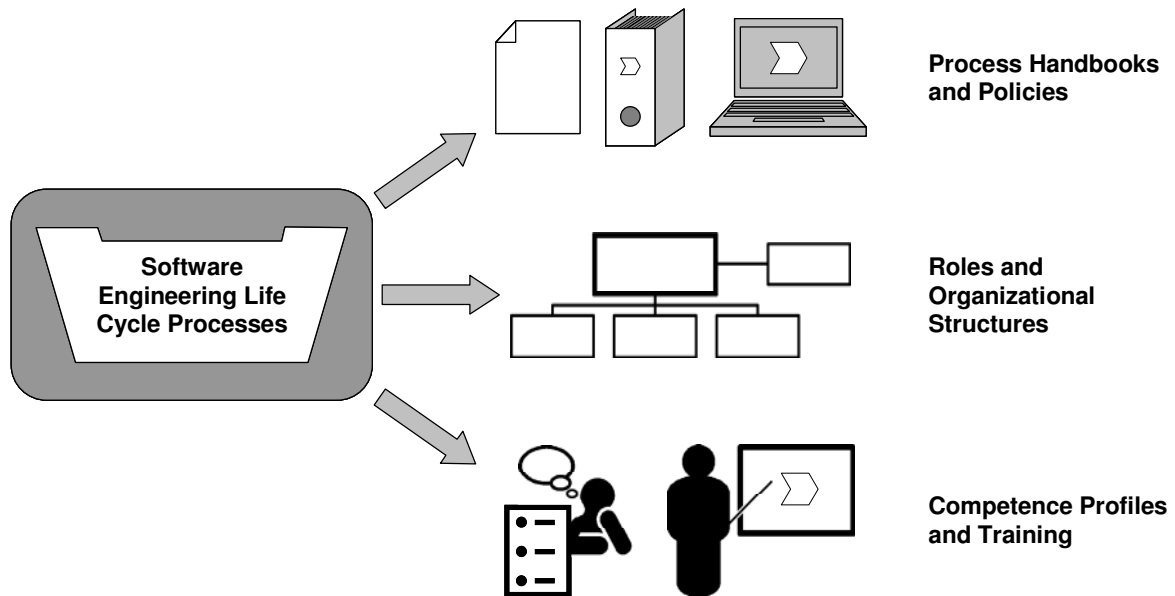


Figure 3: Examples of possible approaches to process definition.

Establish the Software Engineering Processes

A process is established, if it has been accepted by the project staff and can be observed as daily practice throughout the project and organization. Establishing a process means bringing it to the various stakeholders, and making them understand and agree upon the common work practices.

The following practices and measures can ease and support process establishment:

- Involve important stakeholders early during process definition. Address their goals, needs, concerns, and experiences.
- Choose an appropriate and easy-to-understand way of process definition.
- Provide customized training on the new processes and work procedures. This training should be tightly related to real-life work tasks of the training participants.
- Provide process coaching to the team during the first period of working according to the new processes. Such process coaching should also be available later when particularly important or challenging project phases are to be performed.
- Adjust the process definition, if experiences during process establishment suggest so.
- Check from time to time whether the processes are established sufficiently well, and whether the process definition might need to be updated or refined. This can be performed using project reviews or quality audits.

Evolve the Software Engineering Processes

Process evolution involves continuous improvement activities as well as adopting the processes to changing context situations. Project management and quality management will usually recognize early when the processes should be developed further. Nevertheless, it is recommended to conduct process evolution measures on a regular basis.

Regular process analysis—e.g., through project reviews, project retrospectives, and quality audits—will demonstrate to the team that processes matter. It will also create the competence throughout the organization to conduct process changes in a fast and efficient manner.

Other Processes that Interface with the Software Engineering Processes

Software engineering life cycle processes interface to various other kinds of processes of a software organization. They also must be adapted to overall development strategies such as product line development or agile methods. Such process adoption usually lets the overall set of software engineering processes unchanged. Rather, this adoption calls for modification of *how* the processes are to be performed: What work products are to be developed, which stakeholders are to be involved at which points in time, and what activities are to be conducted within the processes.

Figure 4 gives an overview of important processes that interface with or impact the software engineering processes. Project management, quality management, and product management are processes already included into the presented process model (see above). However, there exist detailed and specialized process models for these processes that stem from subject matter areas outside software engineering. Examples are the Project Management Body of Knowledge (PMBOK) and the ISO 9000 quality management standard. These process models should be integrated into the respective software engineering processes.

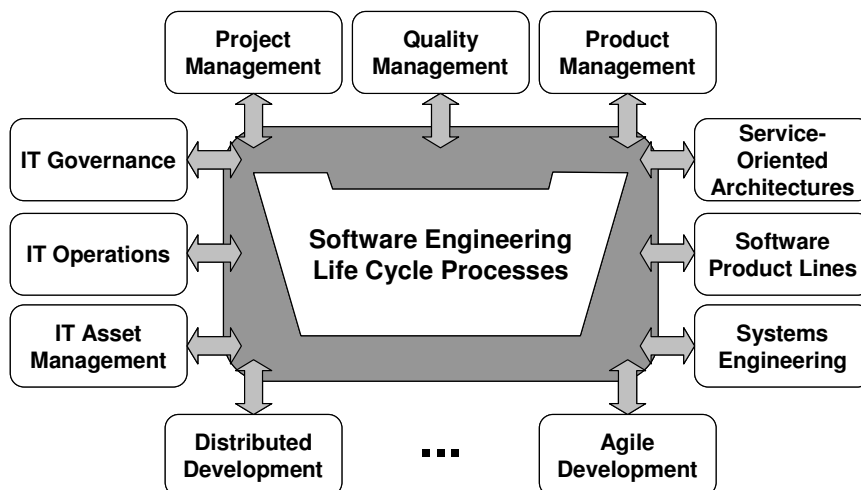


Figure 4: Examples of other processes that interface with the software engineering processes.



IT governance, IT operations, and IT asset management are important processes for every organization that develops or applies software. Sometimes, the links between these processes and software engineering are still much too weak, causing misunderstandings, quality problems, time delays, and cost overruns. The software engineering processes must be designed to interact early and effectively with the various IT processes.

Service-oriented architecture, software product line development, systems engineering, distributed development, agile development, and the like are process paradigms that require the software engineering processes be performed in a specific manner. These requirements must be addressed when defining and establishing the software engineering processes.