

INF5180: Software Product- and Process Improvement in Systems Development

Part 02:
Processes and Process Modeling
(Section A)



Dr. Dietmar Pfahl
email: dietmarp@simula.no

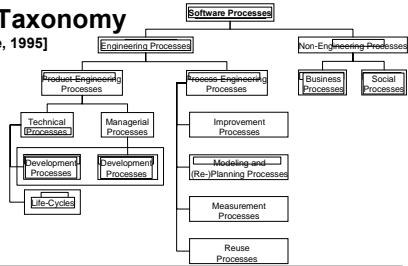
Spring 2010

INF5180 – Spring 2009

Part 02: Processes and Process Modeling

A Process Taxonomy

[Rombach & Verlage, 1995]



H. Dieter Rombach, Martin Verlage,
Directions in Software Process Research,
Advances in Computers, Volume 41,
Marvin V. Zelkowitz (Ed.), Pages 1-63,
Academic Press, Boston, MA, 1995.

Page 2

Copyright 2010 © Dietmar Pfahl



INF5180 – Spring 2009

Part 02: Processes and Process Modeling

What is a (Software Development) Process?

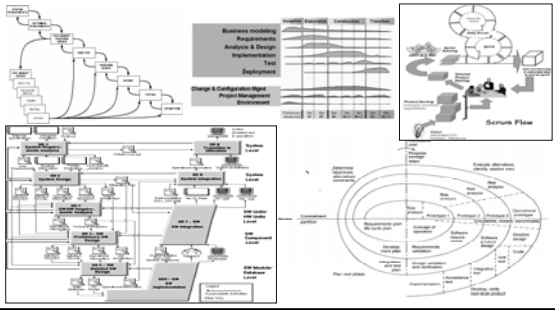


Page 3

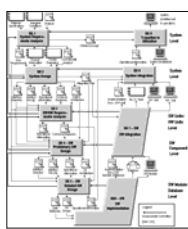
Copyright 2010 © Dietmar Pfahl



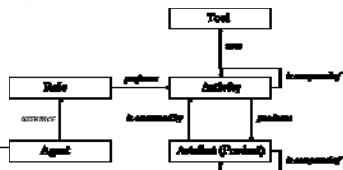
SW (Product-)Engineering Process (Model) Examples



What is a (Software) Process Model?



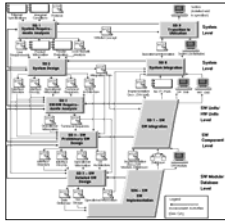
- "Software Process Model: An abstract software process description. It can be more or less formal." [Lonchamp 93]
- Key elements:



What are the Goals of Process Modeling?

- To enable effective understanding and communication
 - At one development site (developers, teams, ...)
 - Between development sites (distributed development, outsourcing, contractor-supplier relations, ...)
- To improve software development activities
 - Improving real processes requires measurement and measurement requires defined processes
 - Evolving processes
- To support project management
 - Transparency, tracking, ...
- To guide the developers
 - Incorporating new employees
- To support reuse of process knowledge
- To support automatic process enactment
 - Workflow support
 - CASE tools

Characterization of Process Models



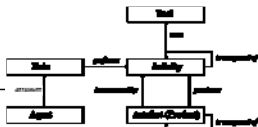
A *Process Model* defines:

- an identifiable activity or a group of activities
- a hierarchy of activities
- the control flow between activities
- the input/output products of activities
- the product flow
- the relations between activities and techniques, methods, tools, and roles

The Role Concept

- *Role*
 - A role is in charge of one or more activities defined in one or more processes
 - A role has defined responsibilities
 - Possible relationships between agents and roles

- 1 : 1
- 1 : m
- n : 1
- n : m



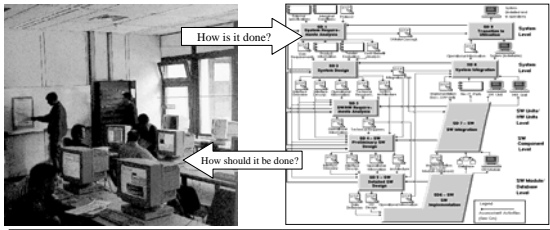
Role Responsibilities

RASCI Matrix

Roles \ Activities	Module developer	Moderator	Tester	Quality assurer
Module design	R			
Module coding	R			
Module review	S, R	S		A
Module testing			R	I

- R = Responsible
- A = Approve
- S = Support
- C = Consult
- I = Inform

Descriptive vs. Prescriptive Process Models

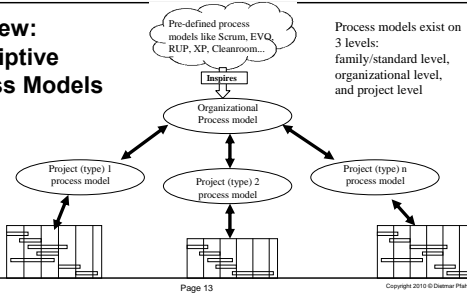


Prescriptive vs. Descriptive Process Models

- Prescriptive Models (theoretical)
 - “Ideal” Process
 - (Assumed) best practice
 - Often requires instantiation and detailing
 - Deviations from real processes are likely
 - Examples: waterfall, V-model, spiral model, incremental, iterative, evolutionary, agile process models
- Descriptive Models (empirical)
 - Accurate elicitation of actual, real processes
 - Basis for the revision of existing (prescriptive) process models based on observation and experience

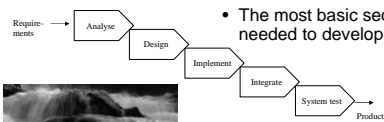
Prescriptive Process Modeling

Overview: Prescriptive Process Models



Process models exist on 3 levels: family/standard level, organizational level, and project level

An Example Process: The Waterfall Model

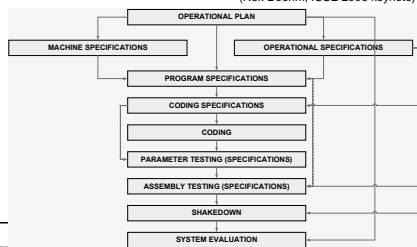


• The most basic sequence of activities needed to develop sizable software

- Well tried, well documented model
- Well suited to successive breakdown
- Is simple to manage
- Requires stability
- Requires careful analysis and planning
- Results in "big-bang integration"

Early Waterfall Model (1956)

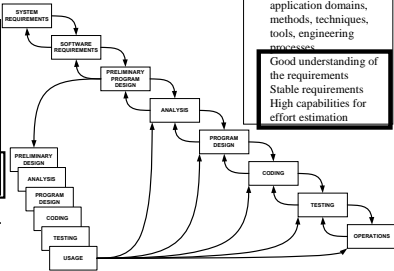
(Ref: Boehm, ICSE 2006 keynote)



The SAGE Software Development Process (Benington, 1956)

Waterfall: Royce Model (1970)

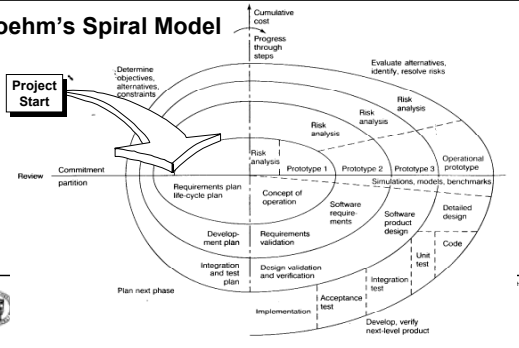
Idea:
 Sequential creation of products on different levels of abstraction (e.g., precede code by design, precede design by requirements) and integration in reverse direction
 Strictly sequential control flow can be weakened by controlled iterations



Prerequisites:
 Familiarity with application domains, methods, techniques, tools, engineering processes
 Good understanding of the requirements
 Stable requirements
 High capabilities for effort estimation



Boehm's Spiral Model



Iterative Enhancement: Overview

- **Origin:** Basili und Turner, 1975
- **Idea:**
 - Develop each increment (i.e., a product part that fulfills a subset of requirements) in a Waterfall style; integrate increment by increment into the product until delivery
 - The focus of the development of an increment might be completion of functionality or structure, but it can also be refinement and improvement
 - Strictly sequential control flow can be weakened by controlled iterations
- **Prerequisites:**
 - Structure of the problem permits incremental development



Iterative Enhancement (or Incremental Development)

Advantages:

- Efficient learning during the project; thus, experience level can be low
- Early availability of a product, with the essential properties of the final product.
- Allows for early customer involvement and feedback
- Applicable when parts of requirements are unclear or unstable
- Supports integration testing
- Good applicability in case of fixed delivery dates (→ prioritize requirements with the customer)

Disadvantages:

- Risk that, by ignoring specific requirements, the product will be designed in such a way that fulfilling future requirements becomes difficult/expensive
 - particularly problematic are non-functional requirements
- Comprehensive version and configuration management is necessary

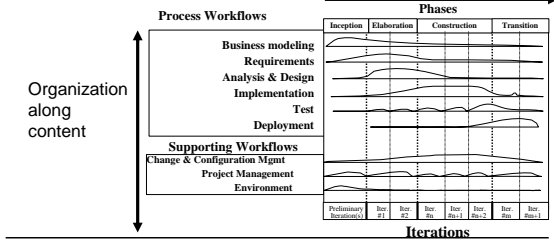
Unified Process

- Family: Iterative Enhancement
- Origin:
 - Ivar Jacobson, James Rumbaugh, Grady Booch, 1998
- Defines process framework that is adaptable to
 - various application domains
 - different organizations
 - different competence levels
 - different project sizes

- Characteristics:
 - use case driven
 - architecture-centric
- Provides only rudimentary instructions
- Refined version:
 - Rational Unified Process (Ph. Kruchten)

Rational Unified Process

Organization along time →



RUP Phases and Iterations — The Time Dimension

- This is the dynamic organization of the process along time.
- The software lifecycle is broken into cycles, each cycle working on a new generation of the product. The Rational Unified Process divides one development cycle in four consecutive phases.
 - Inception phase
 - Elaboration phase
 - Construction phase
 - Transition phase
- Each phase is concluded with a well-defined *milestone*—a point in time at which certain critical decisions must be made, and therefore key goals must have been achieved.



RUP – Static Process

Static Structure of the Process

- A process describes who is doing what, how, and when.
- The Rational Unified Process is represented using four primary modeling elements:
 - Workers (Roles), the "who"
 - Activities, the "how"
 - Artifacts, the "what"
 - Workflows, the "when"

Activities, Artifacts, and Workers



RUP – Resources and Workers (Roles)

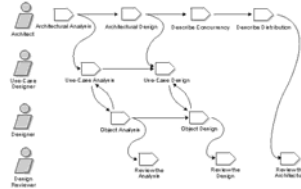
Resource	Worker	Activities
Faust	Designer	Object Design
Mary	Use Case Author	Detail a Use Case
Joe	Use Case Designer	Use Case Design
Bylina	Design Reviewer	Review the Design
Elmer	Architect	Architectural Analysis Architectural Design

• A worker defines the behavior and responsibilities of an individual, or a group of individuals working together as a team.
 • You could regard a worker as a "hat" an individual can wear in the project.
 • One individual may wear many different hats. This is an important distinction because it is natural to think of a worker as the individual or team itself, but in the Unified Process the worker is more the role defining how the individuals should carry out the work.
 • The responsibilities we assign to a worker include both to perform a certain set of activities as well as being owner of a set of artifacts.

RUP Workflow – Example: Analysis & Design

Workflows

- A mere enumeration of all workers, activities and artifacts does not quite constitute a process. We need to describe meaningful sequences of activities that produce some valuable result, and to show interactions between workers.
- A workflow is a sequence of activities that produces a result of observable value.
- In UML terms, a workflow can be expressed as a sequence diagram, a collaboration diagram, or an activity diagram (cf. activity diagram on the right hand side).



RUP Tools

<http://www-01.ibm.com/software/rational/sw-bycategory/subcategory/SW720.html>



Rational Method Composer

- A flexible process platform that includes the Rational Unified Process ([Learn about](#))

Rational Portfolio Manager

- Aligns priorities, projects, and people ([Learn about](#))

Rational SoDA

- Automates software project documentation throughout the entire life cycle ([Learn about](#))

Rational Suite

- Provides a complete, integrated lifecycle solution of best practices, tools, and services ([Learn about](#))

Rational Team Unifying Platform

- Provides common access to development assets, requirements, and process guidance ([Learn about](#))



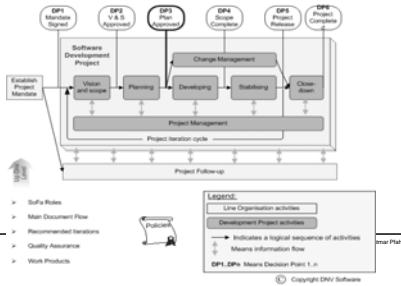
MSF (Microsoft Solution Framework)

For details pls. refer to the White Paper in p02 of the Reading Materials



MSF-Inspired Process Model (at DNV)

For sub-contractor management processes and agile development processes using the Norwegian PS2000 process standard pls. refer to the reports in p02 of the reading materials



Process Families/Standards

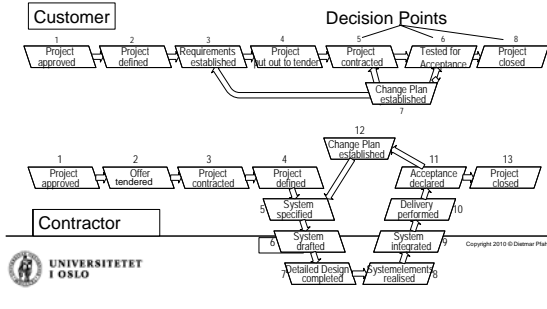
V-Model® XT (XT = Extreme Tailoring)



- Published in January 2005
- Predecessor: V-Model (1997) for military authorities in Germany
- Structured in a modular way
- Mandatory for IT projects in public and military domains in Germany
- Goals:
 - Enhance support for adaptability, scalability, changeability, and expandability of V-Model 97
 - Consider state of the art and adapt to current regulations and standards
 - Expand application range considering the complete system lifecycle of development projects
 - Introduce a process of organizational process improvement

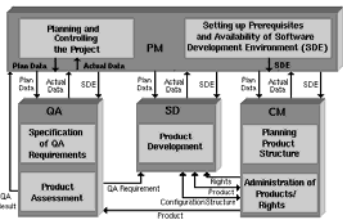
Somewhat Comparable to the role of PS 2000 in Norway

Customer vs. Contractor View



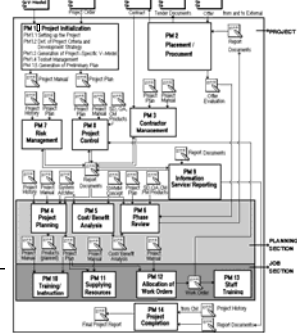
V-Model: The Big Picture

The V-Model comprises four sub-models:
 System Development (SD)
 Quality Assurance (QA)
 Configuration Management (CM)
 Project Management (PM)



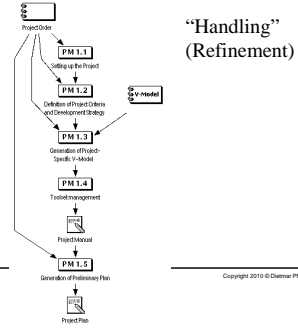
V-Model: Project Management (PM) Sub-Model

- Activity Types:
- Management-related
 - Initialization/Finalization
 - Periodically Required
 - Placement/Procurement-related
 - Planning-related
 - Resource-related

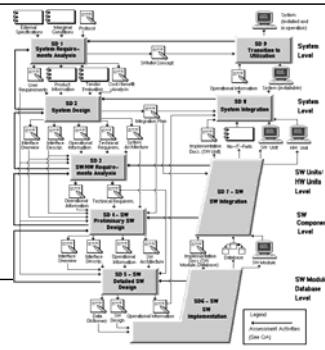


**V-Model:
Project
Management (PM)
Sub-Model**

PM1:
Project
Initialization



**V-Model:
System
Development (SD)
Sub-Model**



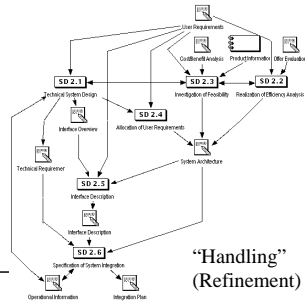
**V-Model:
System
Development (SD)
Sub-Model**

SD2:
System
Design

Phase	Product	to	from	Method	Tool	Ext. Name
System Analysis	System Requirements	System Design	System Analysis			
System Design	System Architecture	System Design	System Design			
SW Requirements Analysis	SW Requirements	SW Design	SW Requirements Analysis			
SW Preliminary Design	SW Preliminary Design	SW Design	SW Preliminary Design			
SW Design	SW Design	SW Integration	SW Design			
System Integration	System Integration	System Verification	System Integration			
System Verification	System Verification	System Verification	System Verification			
SW Integration	SW Integration	SW Integration	SW Integration			
SW Verification	SW Verification	SW Verification	SW Verification			

V-Model: System Development (SD) Sub-Model

SD2:
System
Design



“Handling”
(Refinement)

Descriptive Process Modeling

Goals of Descriptive Process Modeling

- Understand the process
 - Explicit documentation
 - Analyses (consistency, completeness, complexity)
- Support measurement
 - Describe, who can measure what and when
 - Collect quantitative information about processes, products and resources
- Communicate (about) the process
 - Find agreement in case of conflicting opinions
 - Propagation of 'Best Practices'
- Manage the process (and products)
 - Define goals (target values) and control the adherence to these goals.

Process Analysis

- The number of products is higher (approx. twice as high) than the number of processes.
- The complexity of product flow interfaces of processes is relatively high (most of the processes access more than a dozen of products).
- Most of processes are undertaken by several roles (partly over five roles).
- Most of roles are involved in execution of more than a third of the whole process.

30 Processes
66 Products
42 Resources

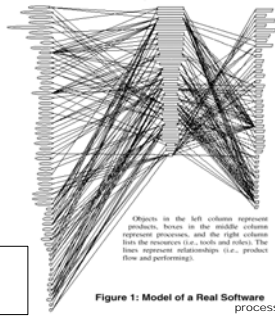


Figure 1: Model of a Real Software process

Modeling Languages (suitable for PM)

- **Business Process Modeling Notation (BPMN)**, and the **XML** form BPML is an example of a **Process Modeling** language.
- **Extended Enterprise Modeling Language (EEML)** is commonly used for business process modeling across a number of layers.
- **Flowchart** is a schematic representation of an algorithm or a stepwise process.
- **IDEF** is a family of modeling languages, the most notable of which include **IDEF0** for functional modeling, **IDEF1X** for information modeling, and **IDEF5** for modeling ontologies.
- **Unified Modeling Language (UML)** is a general modeling language to describe software both structurally and behaviorally. It has a graphical notation and allow for extension with a **Profile (UML)**.

Process Modeling Tools

- **Commercial tools not dedicated to process modeling**
 - E.g. ABC Flowcharter, Microsoft Visio, Statemate
- **Workflow Management Systems**
 - E.g. ARIS Toolset (event-process chains)
- **Research prototypes**
 - E.g., Spearmint
