Technische Universität
Kaiserslautern

o1o1seda o1o1oo

software engineering dependability

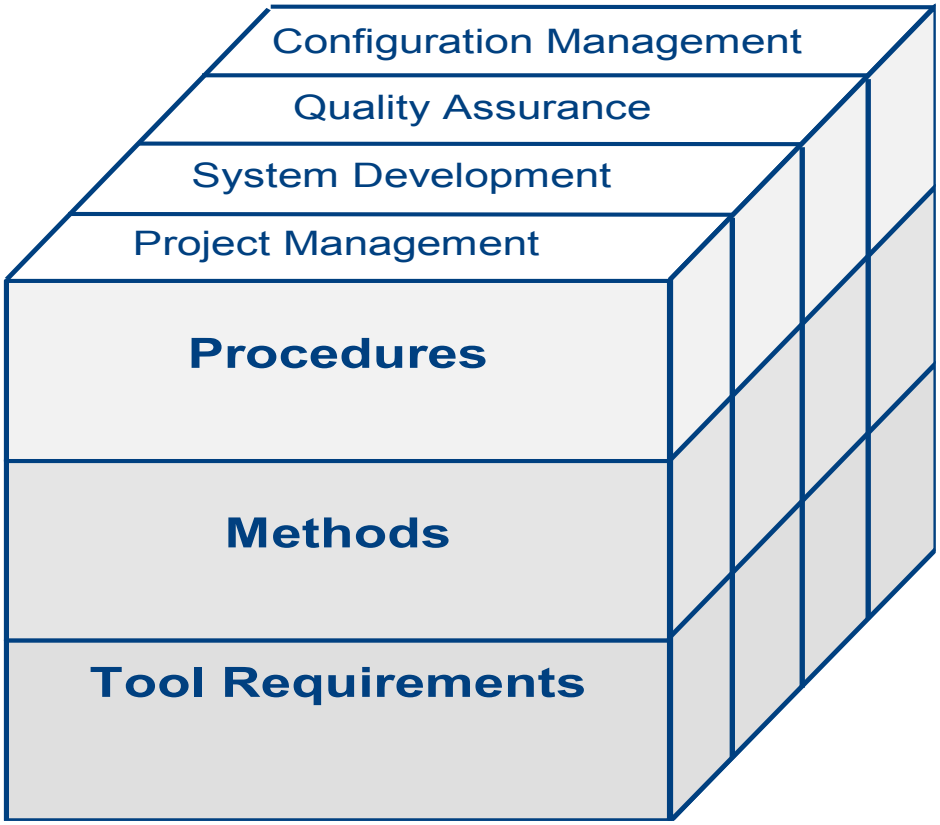
Quality Management of Software and Systems:
Processes and QM

Contents

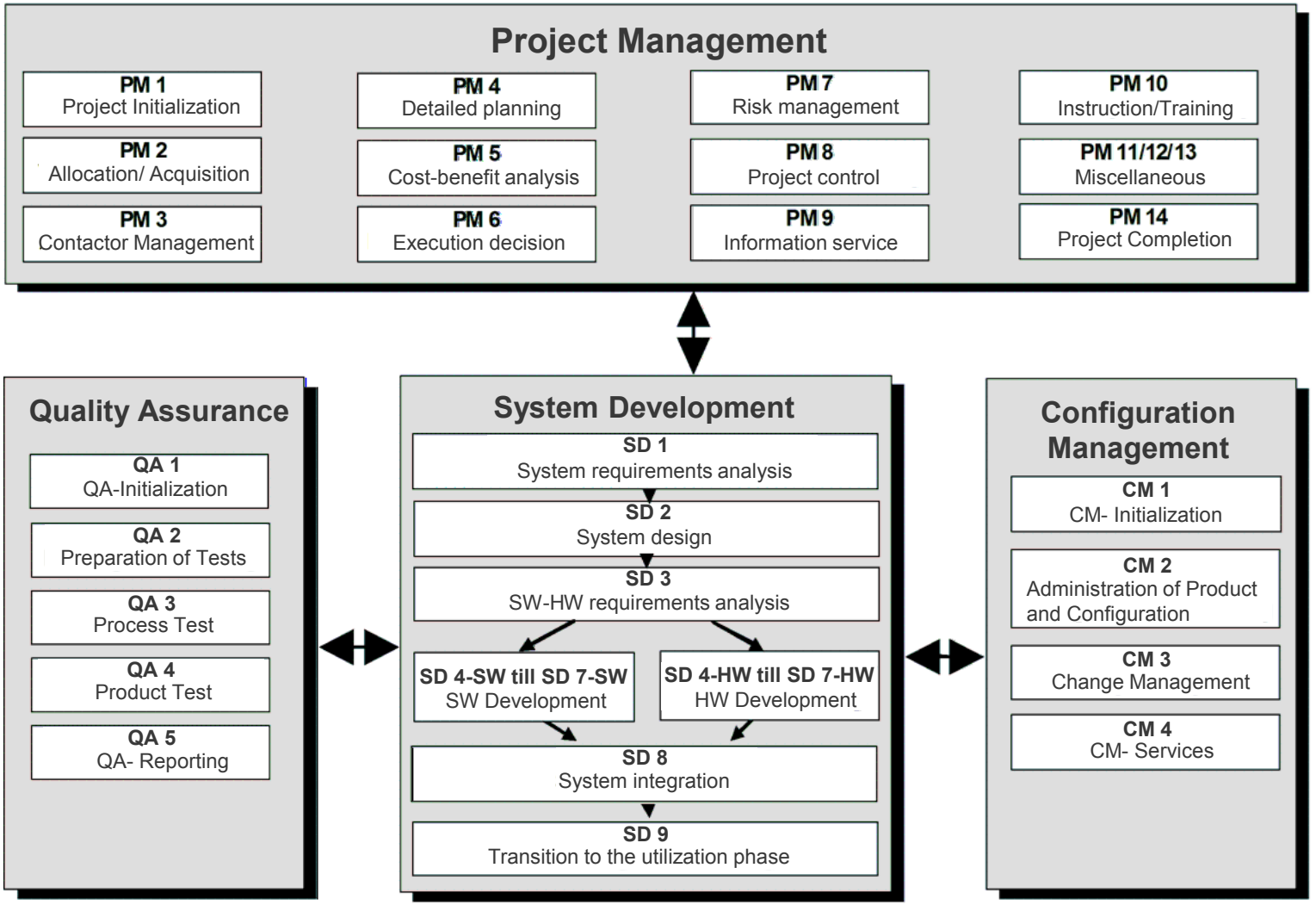
- V-Model XT
- Rational Unified Process (RUP)
- Extreme Programming (XP)
- Processes

V-Model XT
Starting point: V-Model 97

- Broadened guideline for performing IT-projects
  - Generally binding for IT-projects in public and military domains
  - Increasingly applied in business, partially in SMBs, too
- 07/1997: update and release of V-Model ´97
  - No further development since that time
  - V-Model ´97 is not state of the art in all fields



V-Model XT
Starting point: V-Model 97



V-Model XT
Goals of V-Model XT development

- Enhance support for adaptability, practicability, scalability, changeability and expandability of V-Model
- Consider state of the art and adapt current regulations and standards
- Expand application range with respect to consider the whole system lifecycle in scope of development projects
- Introduce a process of organizational improvements for process models

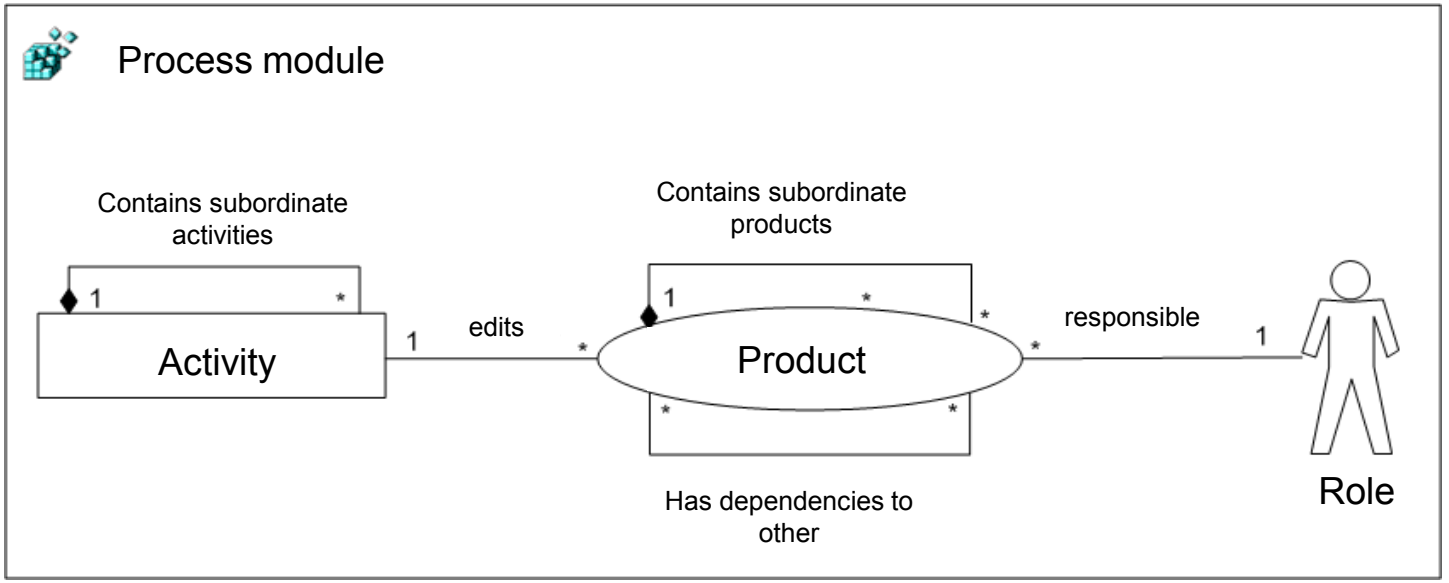
V-Model XT
Process model and objectives

- V-Model XT is a process model
  - Development model for the customer
  - Development model for the contractor
  - Quality model for companies

- Objectives of the V-Model XT
  - Minimizing project risks
  - Quality improvement and quality guarantees
  - Budget containment for the whole project and system life-cycle
  - Communication improvements between all participants

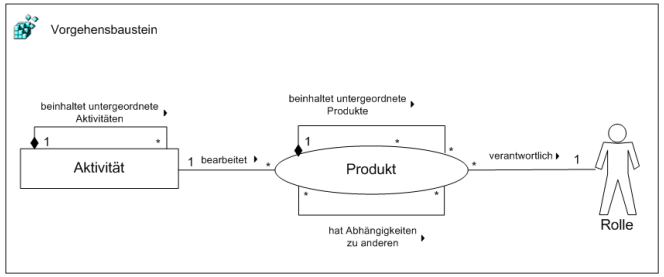
Process modules as modular elements

- The V-Model is composed of modular blocks, so-called process modules



- A process module
  - encapsulates roles, products and activities
  - is a unit, which can be independently used
  - is a unit, which can be updated or extended independently

V-Model XT
Model element dependencies



Product Group

Activity Group

Role

responsible

Product

creates

Activity

Role

Role

collaborate

Subject

Subject

Subject

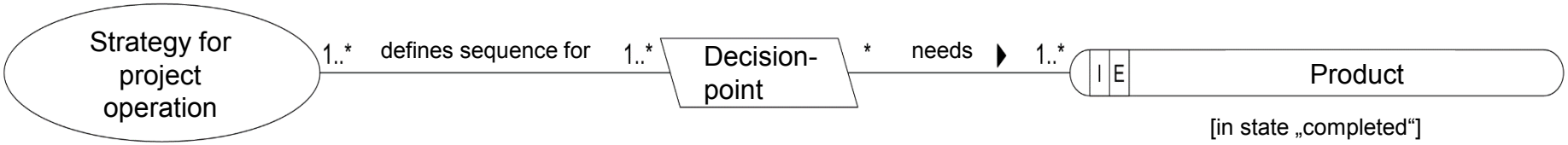
edit

Step

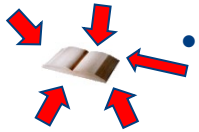
Step

V-Model XT: Project Execution Strategies and Decision Points

- Process components, products and activities do NOT constrain or suggest any order of execution
- A strategy for project operation defines the sequence in which the project-progress-levels have to be reached
- A decision-point
  - Defines a date, which is determined by the project plan, at which a “progress-decision” (GO/NOGO) will be made
  - Defines a set of products, which have to be completed at the decision-point. such that the “progress-decision” can be made.



V-Model XT: Philosophy - Goal and result oriented approach



- Products take center stage as they are the (intermediate) results of a project

- Strategies for project operation and decision-points define the sequence of product completion and thus the elementary structure of the project‘s progress

- Detailed planning and controlling will be performed based on development and completion of products

- One role is responsible for each product.

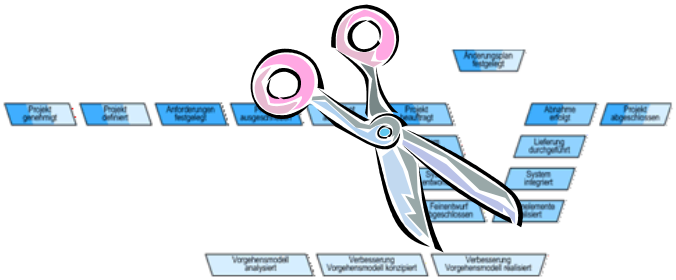
- The quality of products is checkable by using:
  - Product Requirements
  - Existing dependencies with other products

V-Model XT
Types of projects and tailoring

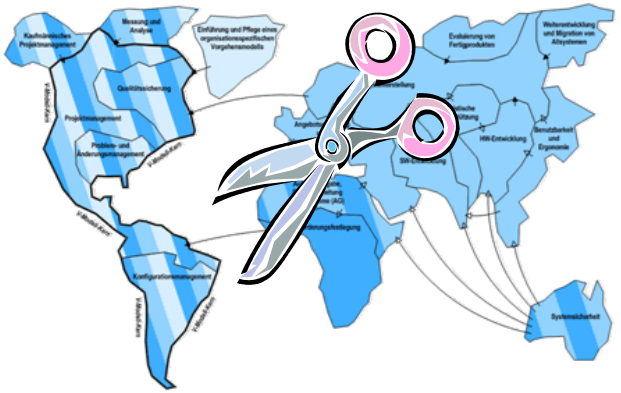
Project types

- System development project of a client
- System development project of a contractor
- Development of an organization-specific process model

Decision-points & Strategies for project operation

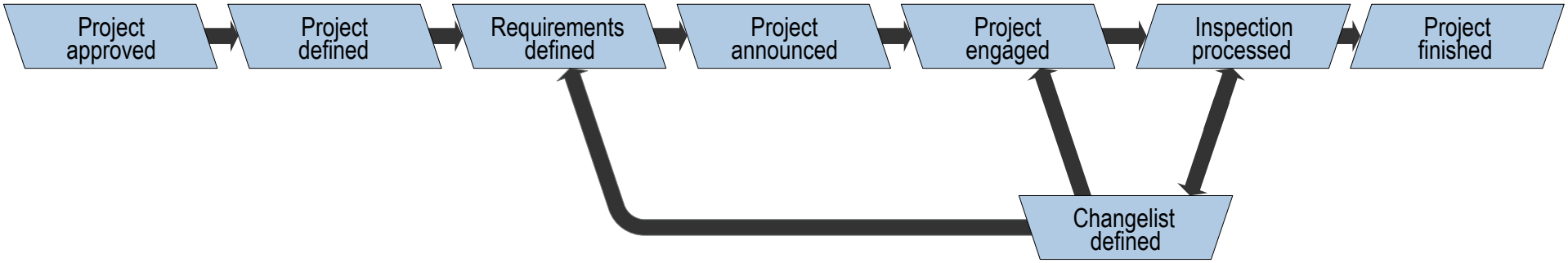


Map of process modules with V-Model core



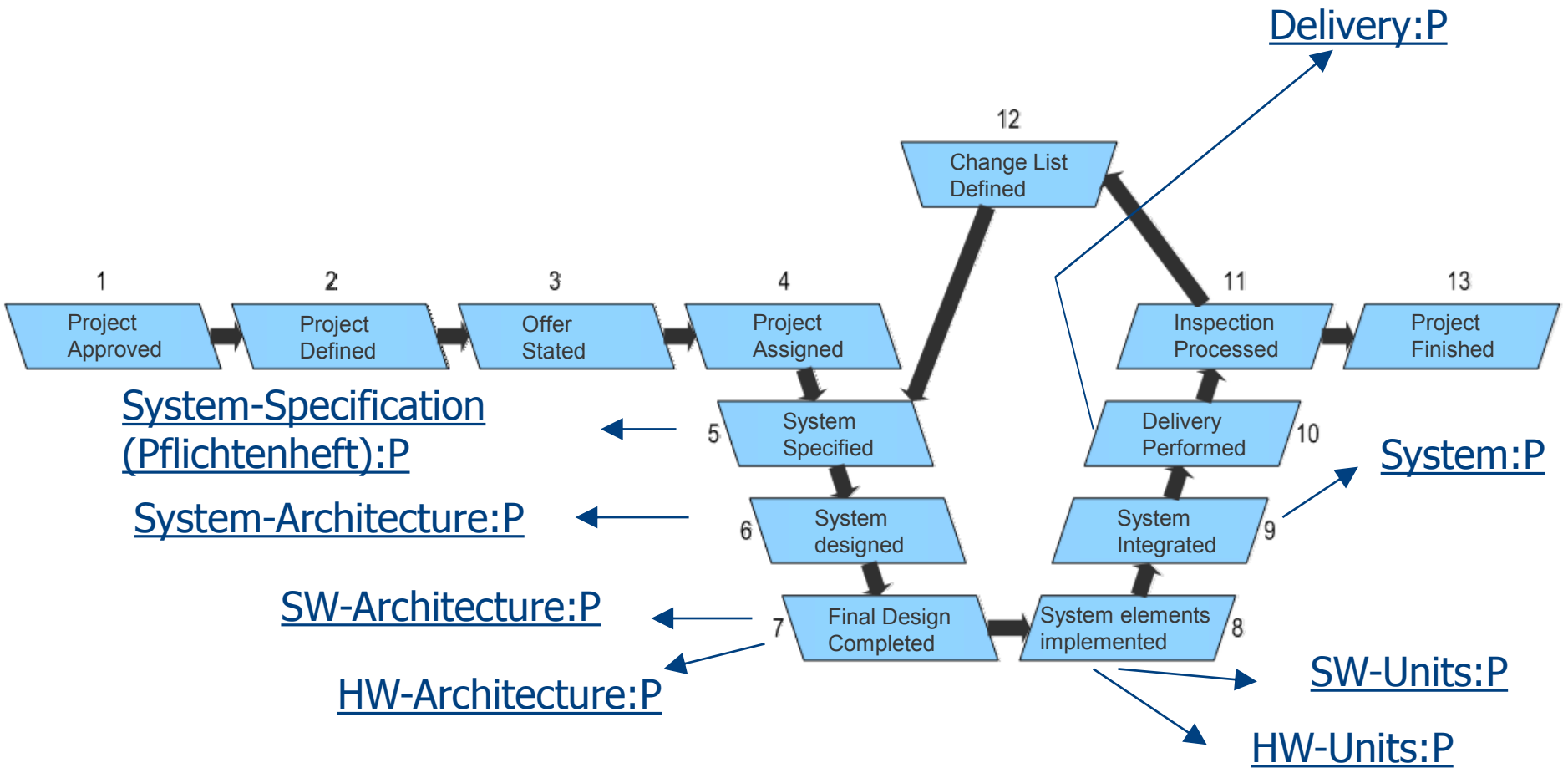
V-Model XT
Project Execution Strategy for Client

- Tailoring delivers
  - Strategy for project operation
  - Process modules (if necessary supplemented)

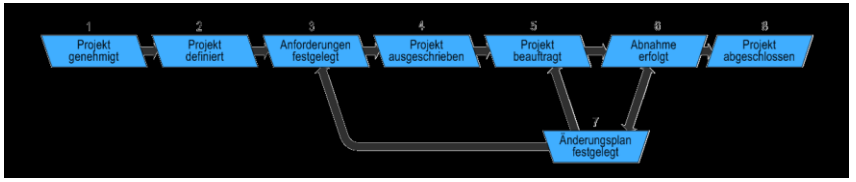


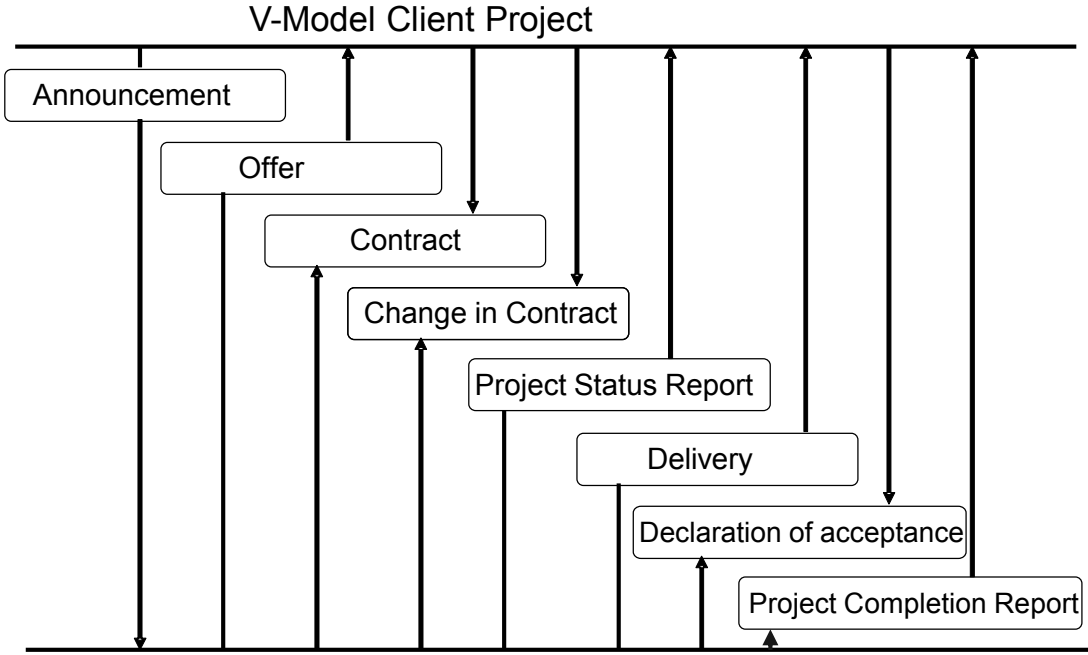
- Process modules define the project's activities and products
- The strategy for project operation has to be instantiated concretely for a specific project

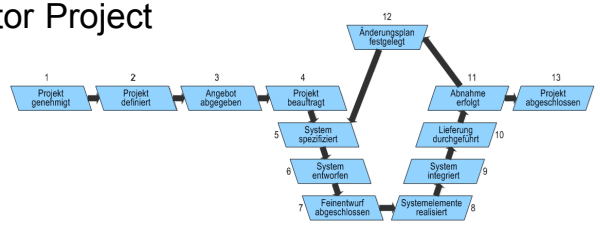
V-Model XT
Project Execution Strategy for Contractor



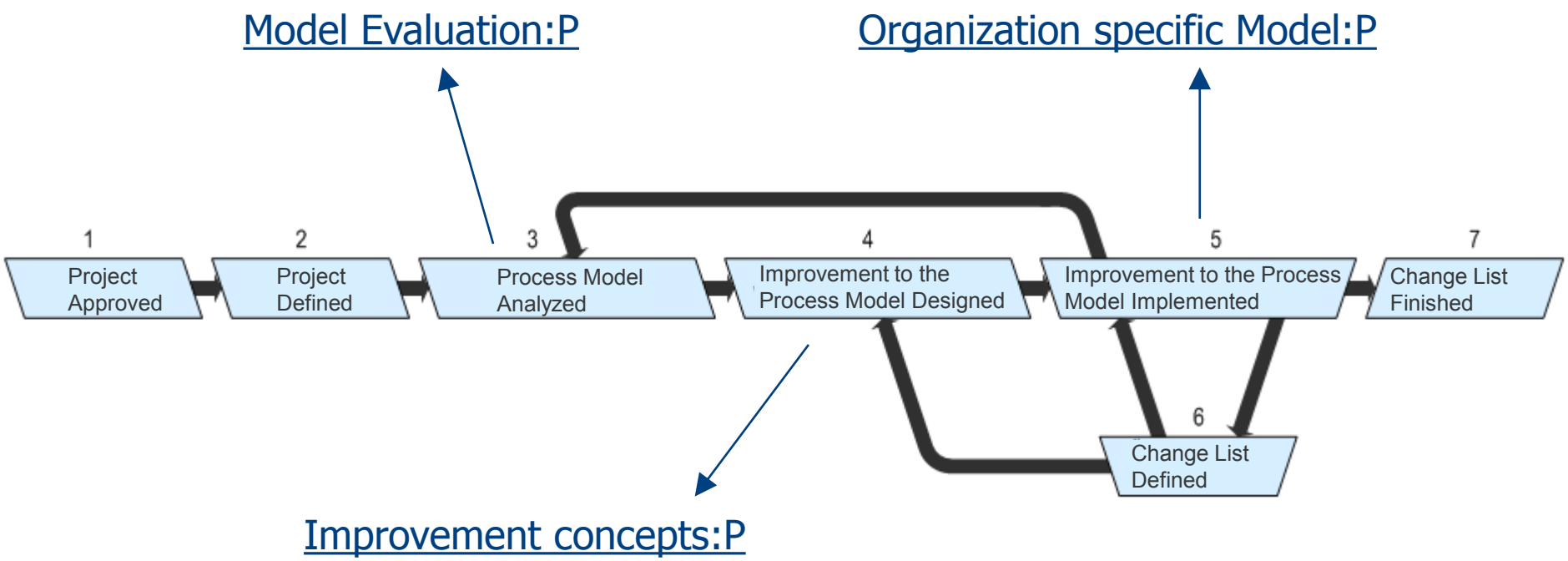
V-Model XT
Interface between Client and Contractor



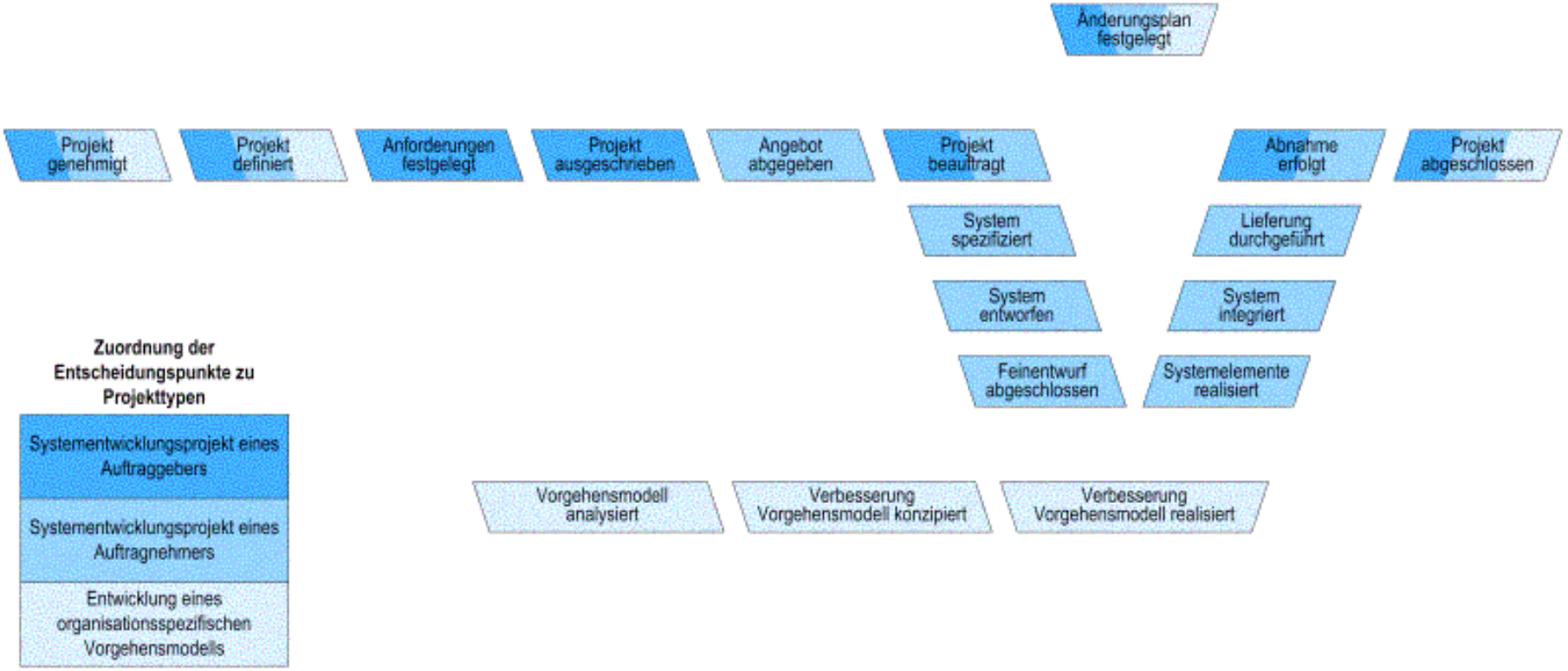




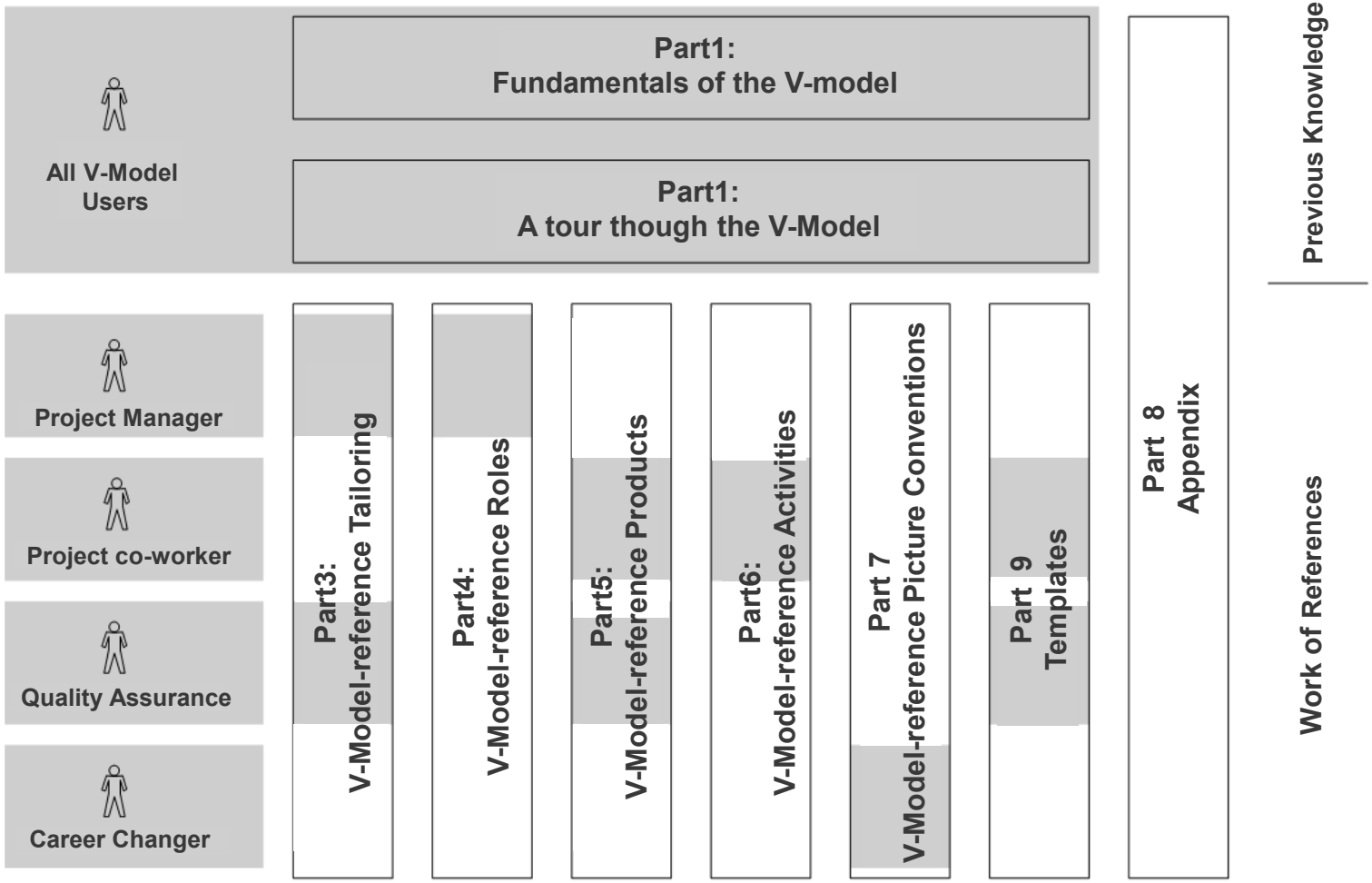
V-Model XT: Project Execution Strategy – Organization Specific Model



V-Model XT
Decision Points: Overview

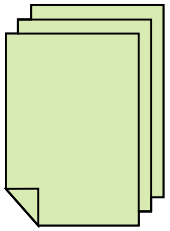


V-Model XT
Document Size



V-Model XT Availability

- V-Model
  - PDF, Word und HTML, (XML)
  - Training material
  - Tutorial
  - Example Projects



- Product Templates (RTF)

- V-Model XT Editor: Open Source Tool for editing and enhancing V-Model XT
- V-Model XT Project wizard: Open Source Tool for Tailoring of V-Model XT
- Open Source: http://fourever.sourceforge.net
- Binary: http://www.v-modell-xt.de

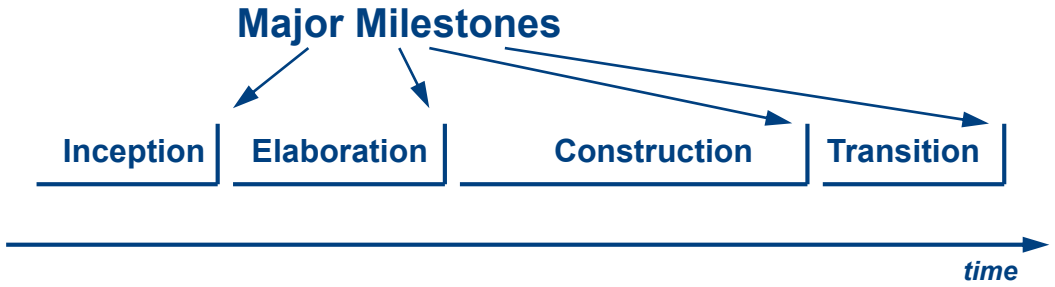
For more information visit
http://www.v-modell-xt.de

Rational Unified Process (RUP)

- Software development process
- Customizable and extensible framework
- Language used is UML
- Use-Case driven
  - Use-cases are the starting point and the base for the development
- Architecture centered
  - The System is divided in components und subsystems through the architecture
- Iterative and incremental process
  - Segmentation in smaller projects
  - Iterations are steps within the workflow
  - Increments are extensions and improvements of the product

Rational Unified Process (RUP) Overview

- Development consists of multiple cycles
- Each cycle finishes with a product release, i.e. after each cycle a product is delivered to the customer
- Each cycle consists of four phases
  - Inception
  - Elaboration
  - Construction
  - Transition



- Each of these phases in divided in nine workflows

Rational Unified Process (RUP)
Best Practices

- Iterative development
- Requirements management
- Architectural centered development
- Visual modeling (with UML)
- Quality assurance
- Change management (configuration management)

- The „Best Practices“ are the design principles for RUP and can be found within the workflows

Rational Unified Process (RUP)
Inception Phase - Conceptualization

- Formulation of the product idea, the vision
- Specification of essential business use cases
- Definition of project size
- Prediction of costs and risks
  - Simplified cost estimate

- **Life Cycle Objective Milestone**

Rational Unified Process (RUP)
Elaboration Phase – Analysis/Design

- Specification of product features
- Architectural design
- Scheduling of necessary activities and resources

- **Life Cycle Architecture Milestone**

Rational Unified Process (RUP)
Construction phase - Implementation

- Product creation
- Development of the architecture
- Result: finished product

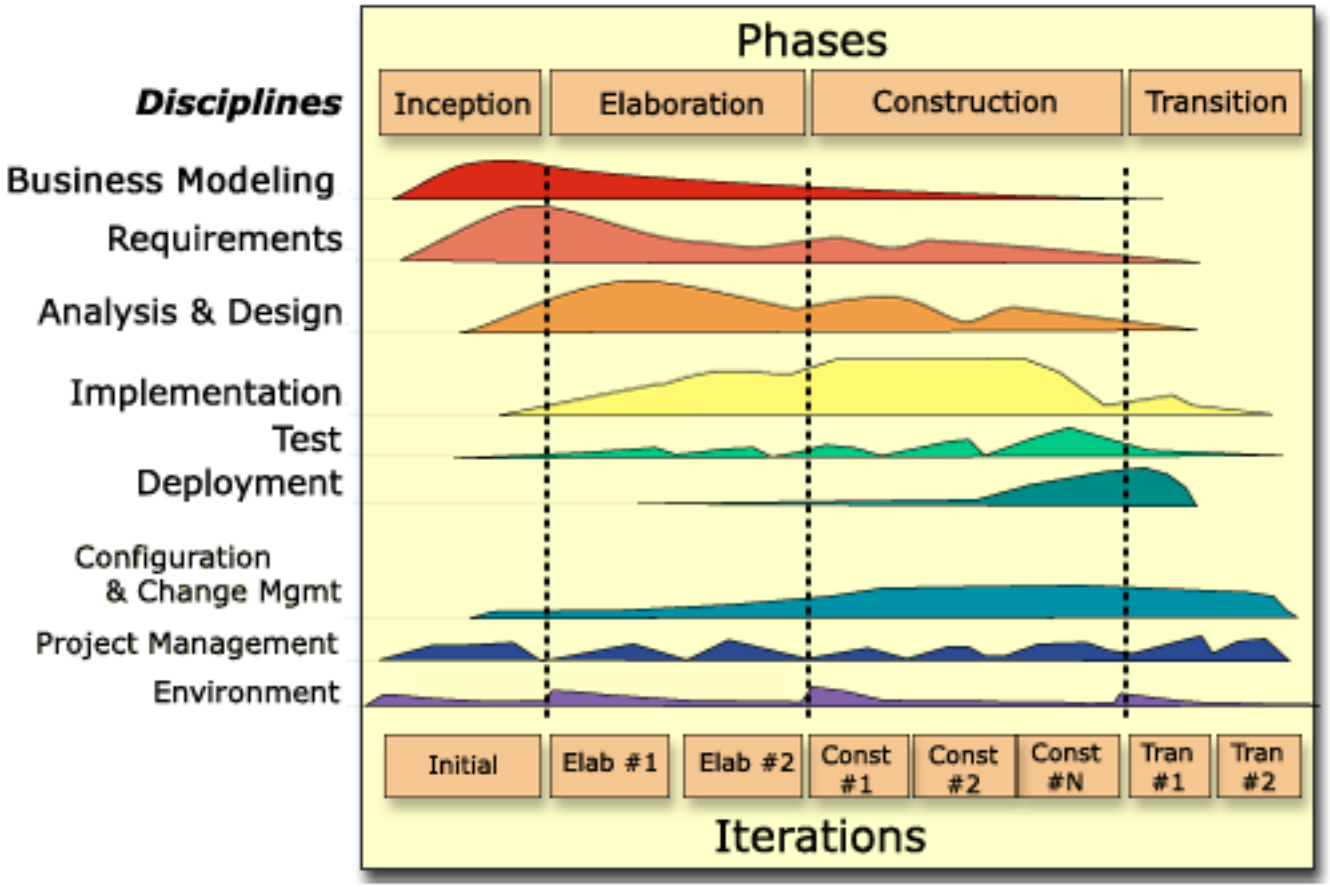
- **Initial Operational Capability Milestone**

Rational Unified Process (RUP)
Transition phase – Market release

- Product release to the customers
- Examination of quality level
- Delivery, training, service support, maintenance

- **Release Milestone**

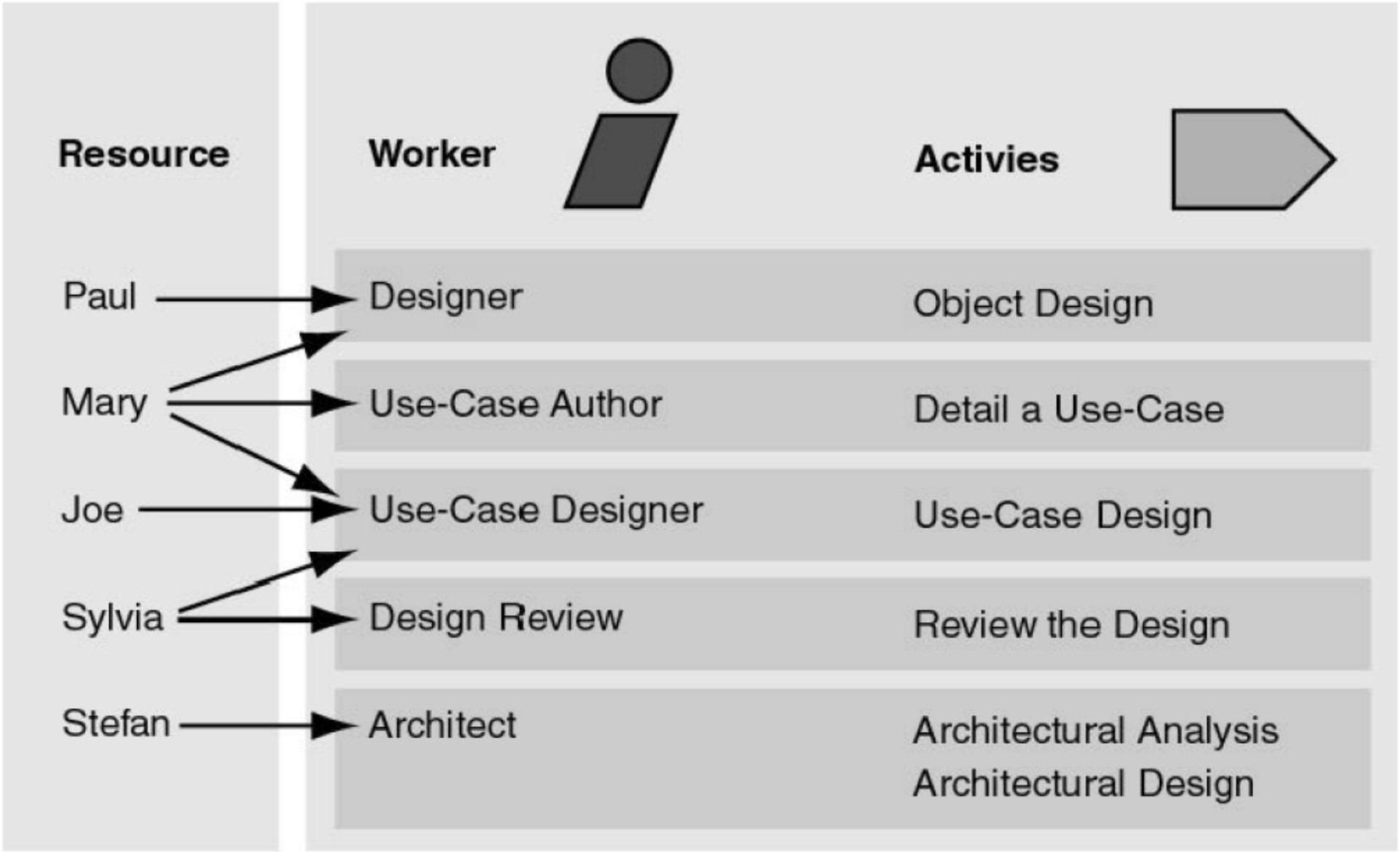
Rational Unified Process (RUP)
Process structure



Rational Unified Process (RUP)
Process structure

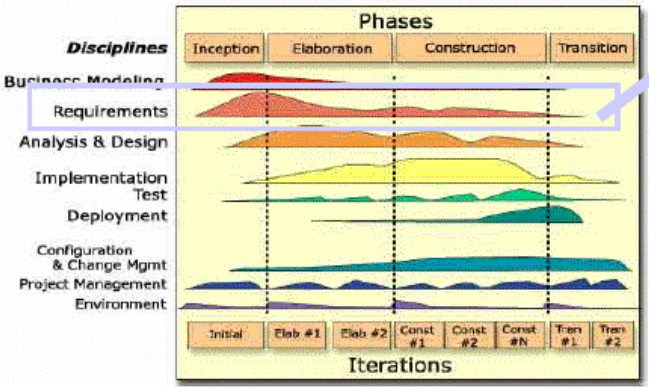
- Each phase consists of at least one iteration
- Each iteration is composed of workflows
- Workflow elements are roles („*Workers*“), activities, and artifacts
  - Worker: „who“
  - Artifact: „what“
  - Activities: „how“
  - Workflows: „when“
- Thus, it is specified who does what, when and how for the whole process

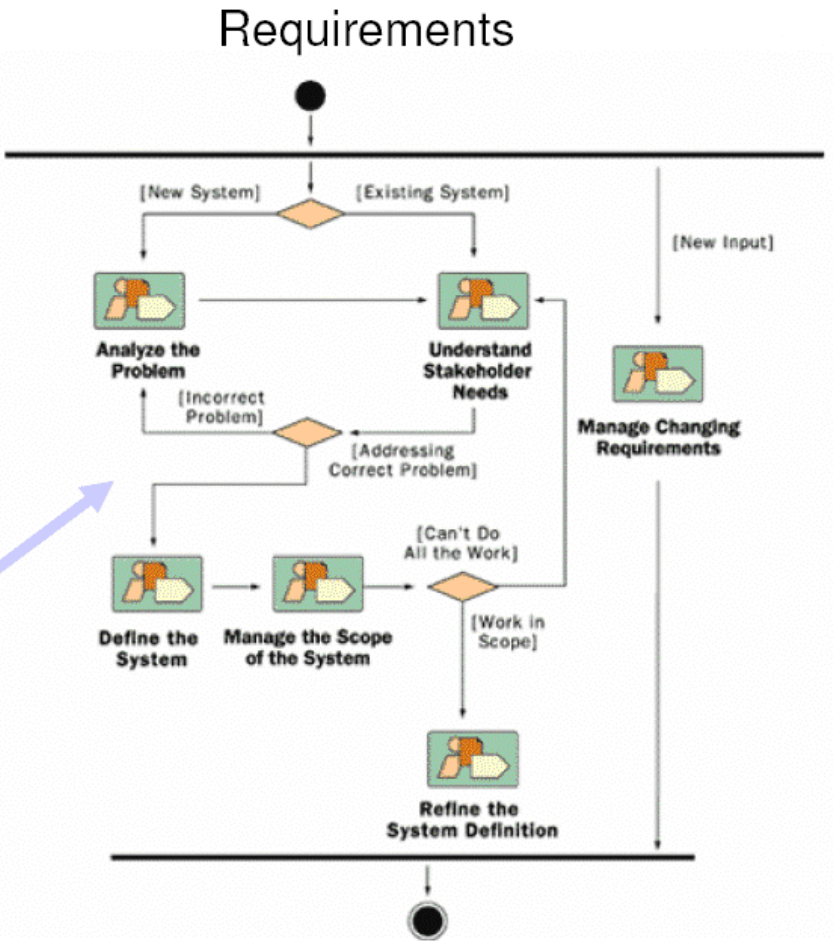
Rational Unified Process (RUP)
Persons and Workers



Rational Unified Process (RUP) Workflows

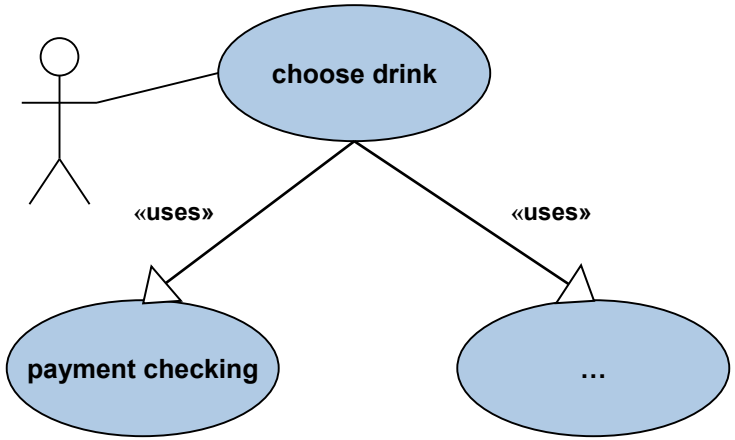
- For each workflow, starting from business modeling, the implementation, up to the project management, RUP provides tool supported procedures





Rational Unified Process (RUP)
Use-case based

- User interacts with system, system executes a series of activities
- A use-case is the description of an interaction and specifies the **functional requirements the users have**
- Initiated through an actor and consists of several activities
- A set of use-cases specifies the requirements for the whole system
- Use-cases are modeled using UML
- Use-cases are the basis for all subsequent parts of RUP

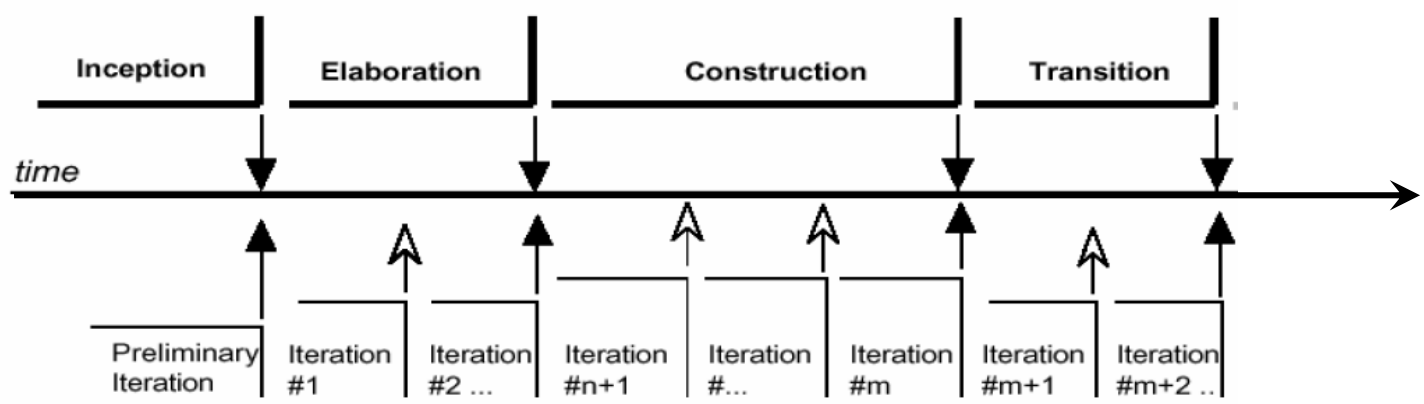


Rational Unified Process (RUP)
Architecture centered

- The architecture structures the system, using components and subsystems
- Provides ‘views’ for the static and dynamic system aspects
  - Logical view
  - Implementation view
  - Process view
  - Distribution view
  - Use-case view
- Affected by
  - Important use-cases (functional requirements)
  - Platform (OS, …)
  - Reusable components (Frameworks,…)
  - Existing applications (Integration of Legacy Systems,…)
  - Non-functional requirements (Performance, reliability, …)
- The most important use-cases constitute subsystems, classes, or components

Rational Unified Process (RUP)
Iterative and incremental

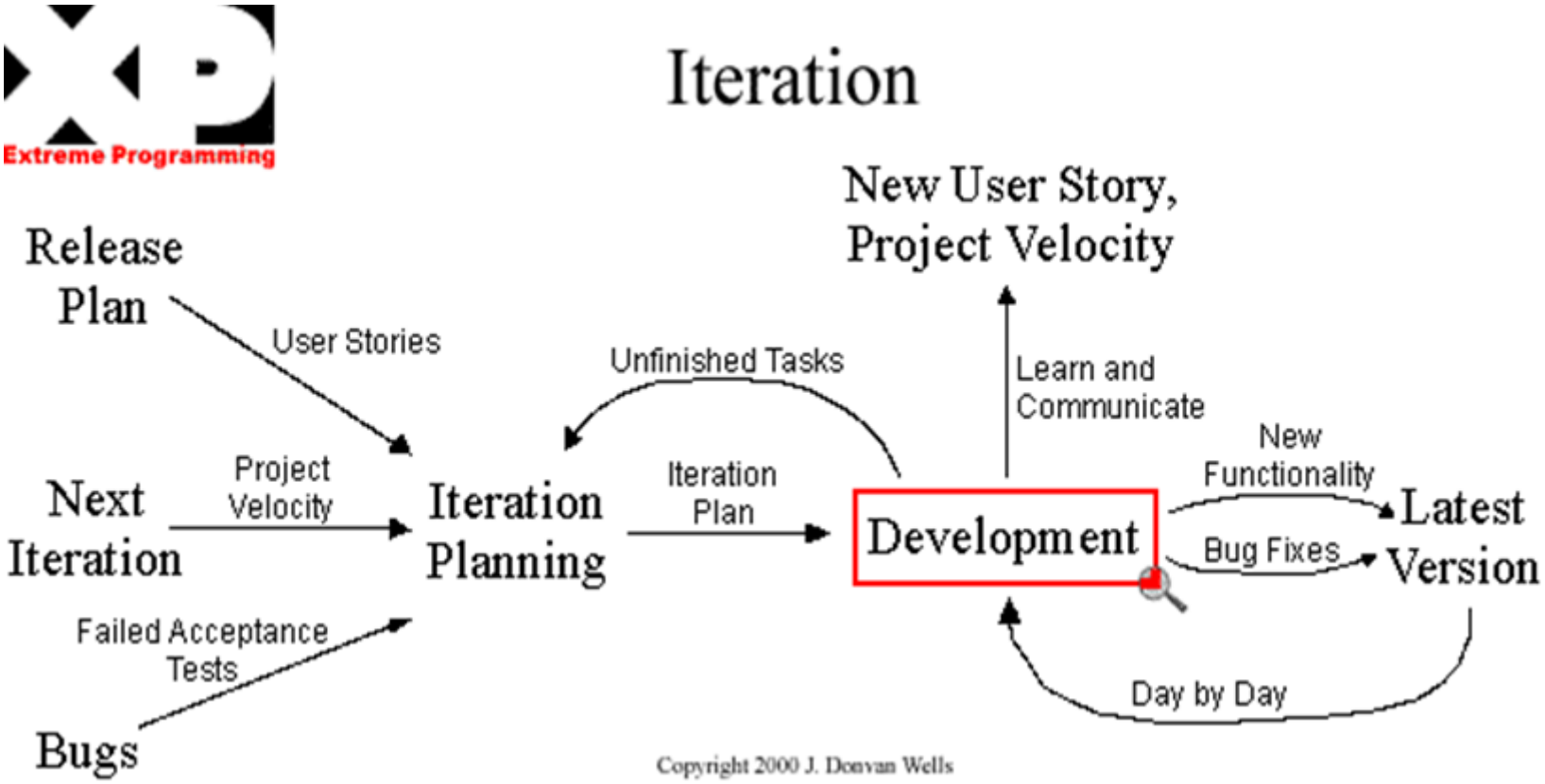
- Project is splitted in several mini projects
- Each mini project is an iteration
- Iterations are steps within the workflows
- Each iteration leads to a product growth
- Each phase consists of at least one iteration

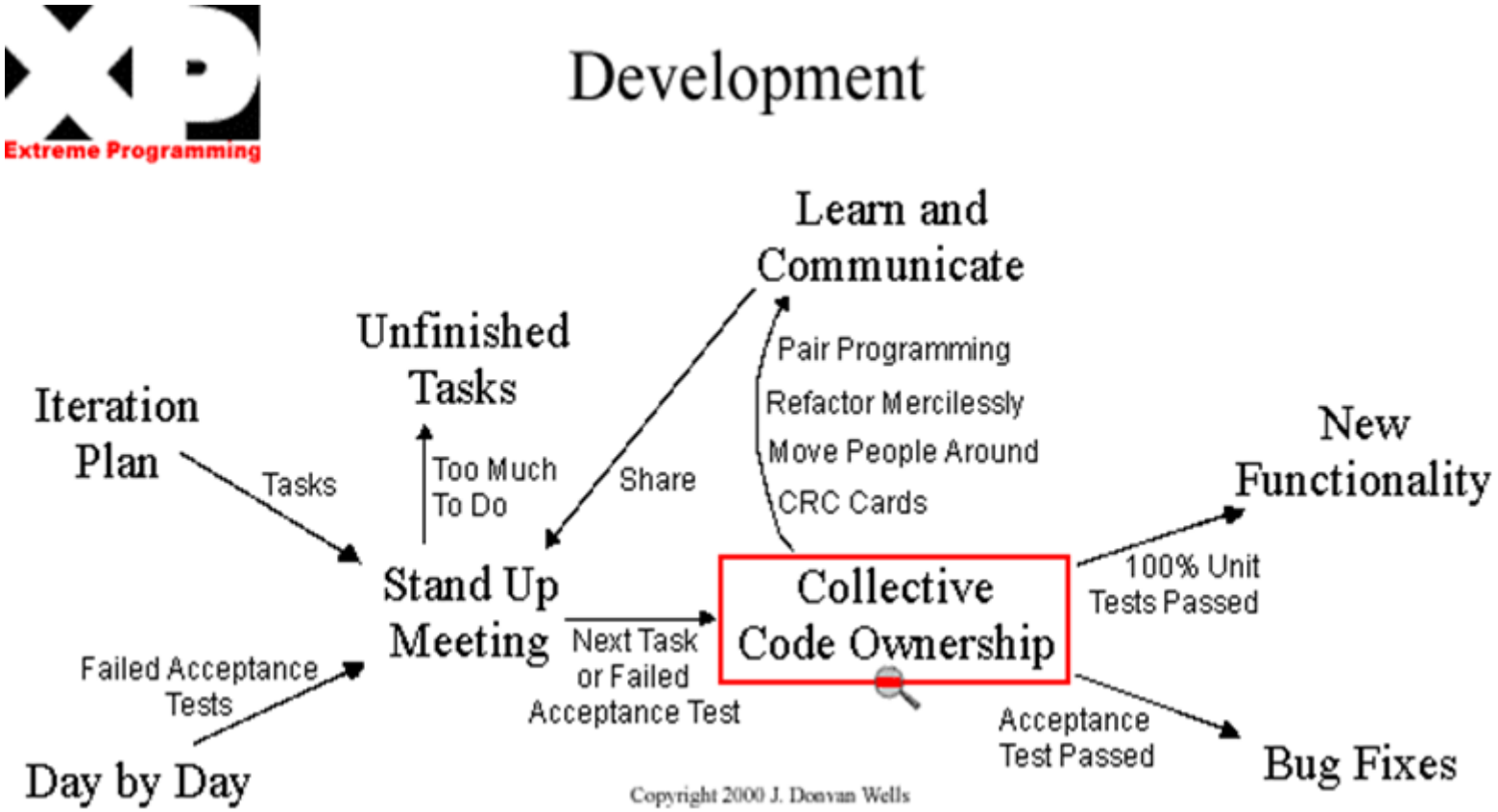


Rational Unified Process (RUP) Adaptable Framework

- Realizing RUP is very complex
  - > 30 roles
  - > 130 activities
  - > 100 result types (artifact types)
- But RUP can be adapted to a company's or project's needs
- Workflows can be shortened or left out, if they are not required

Extreme Programming (XP)



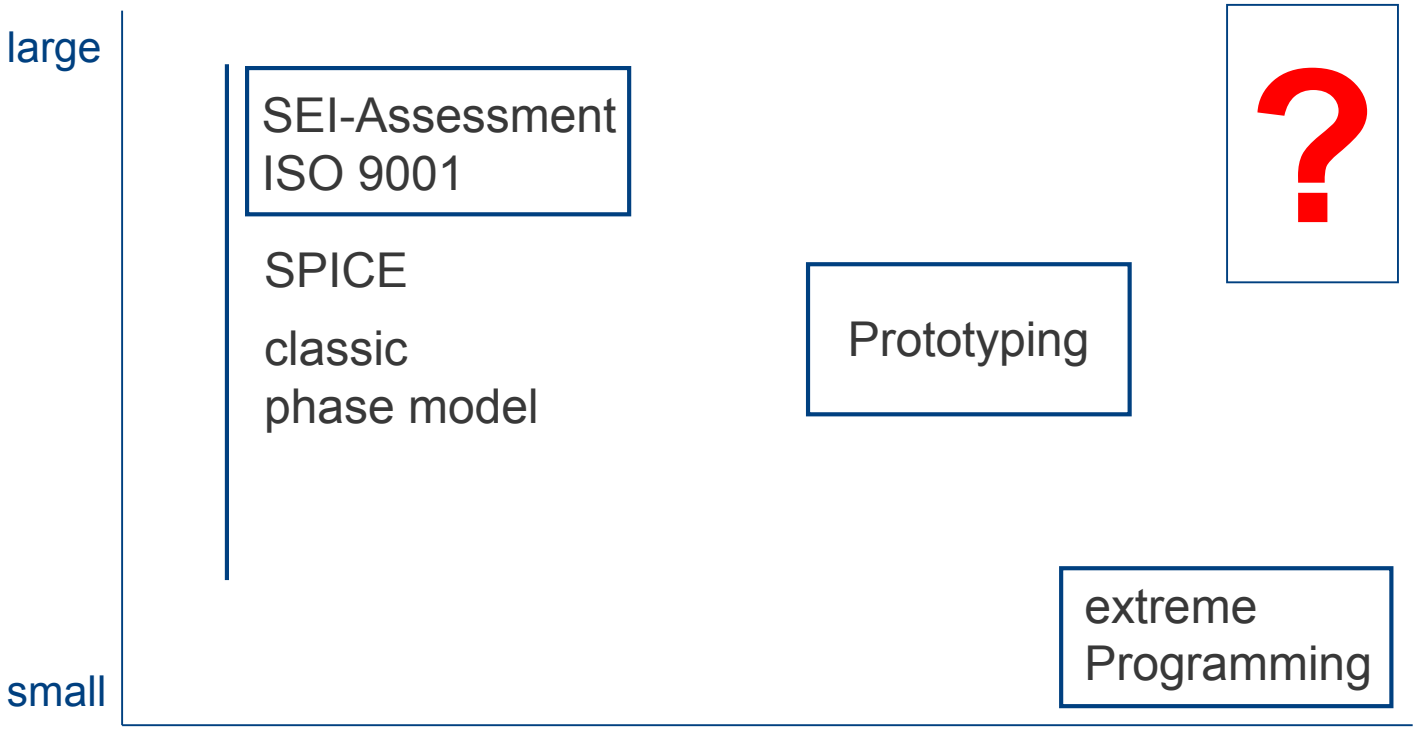


Extreme Programming (XP)

- Small projects (approx. 10 collaborators)
- Unstable or unknown requirements
- Contributory customers
- Strong focus on the customer
- Strong focus on quality

- Danger of leading to chaos (legitimating ad-hoc working procedures)

Processes


large
small
SEI-Assessment
ISO 9001
SPICE
classic
phase model
Prototyping
?
extreme
Programming
stable requirements
known requirements
customer interface
unstable requirements
unknown requirements
customer involvement

Processes Prediction

- Assessments will play a major role in large companies
- The DIN ISO 9001 certificate will be considered necessary, but not sufficient
- Waterfall models will remain
- Waterfall models will be supported by prototyping, to deal with unclear requirements
- Extreme Programming can be used for small projects, if the customer is willing to collaborate and if certain documents are not necessary