

Refactoring the Unified Process



Refactoring the Unified Software Development Process



Jutta Eckstein
jeckstein@acm.org





Overview

- Motivation
- USDP vs. RUP
- Key Aspects
- Refactoring
- Conclusion





Motivation

- Standardization of UML (and USDP??)
- Faster Progress with lighter Process
- Better Applicability
 - also in training environment





Unified Process vs. RUP

- RUP = commercial product
- Unified Process (or USDP) = framework
 - closer to formal level of XP





Key Aspects

- “People are crucial”
- Use Case Driven
- Architecture Centric
- Incremental / Iterative





Use Case - Driven



- no formal standard for textual specification
- priorities according to Risk and Functionality



BUT:



Prioritizing by Developer vs. Business People





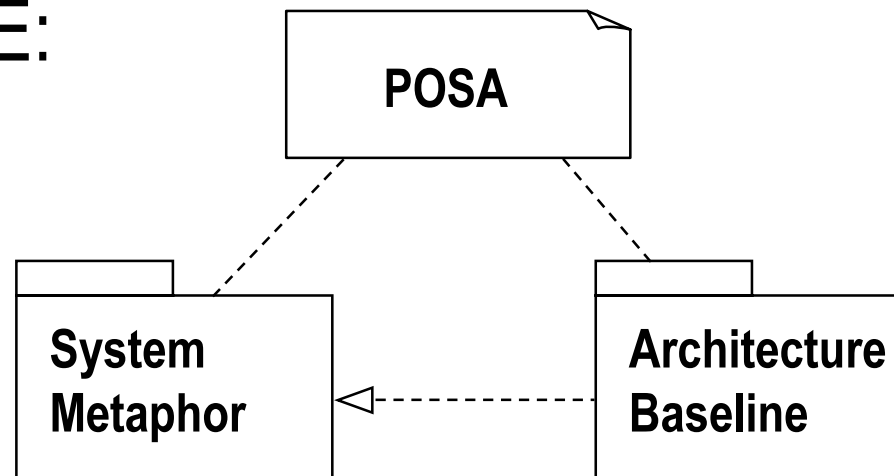
Architecture - Centric



- Architecture Baseline = architectural view on different models
- Metaphor is very high-level



THEREFORE:





Incremental / Iterative

- Short time spans result in a milestone



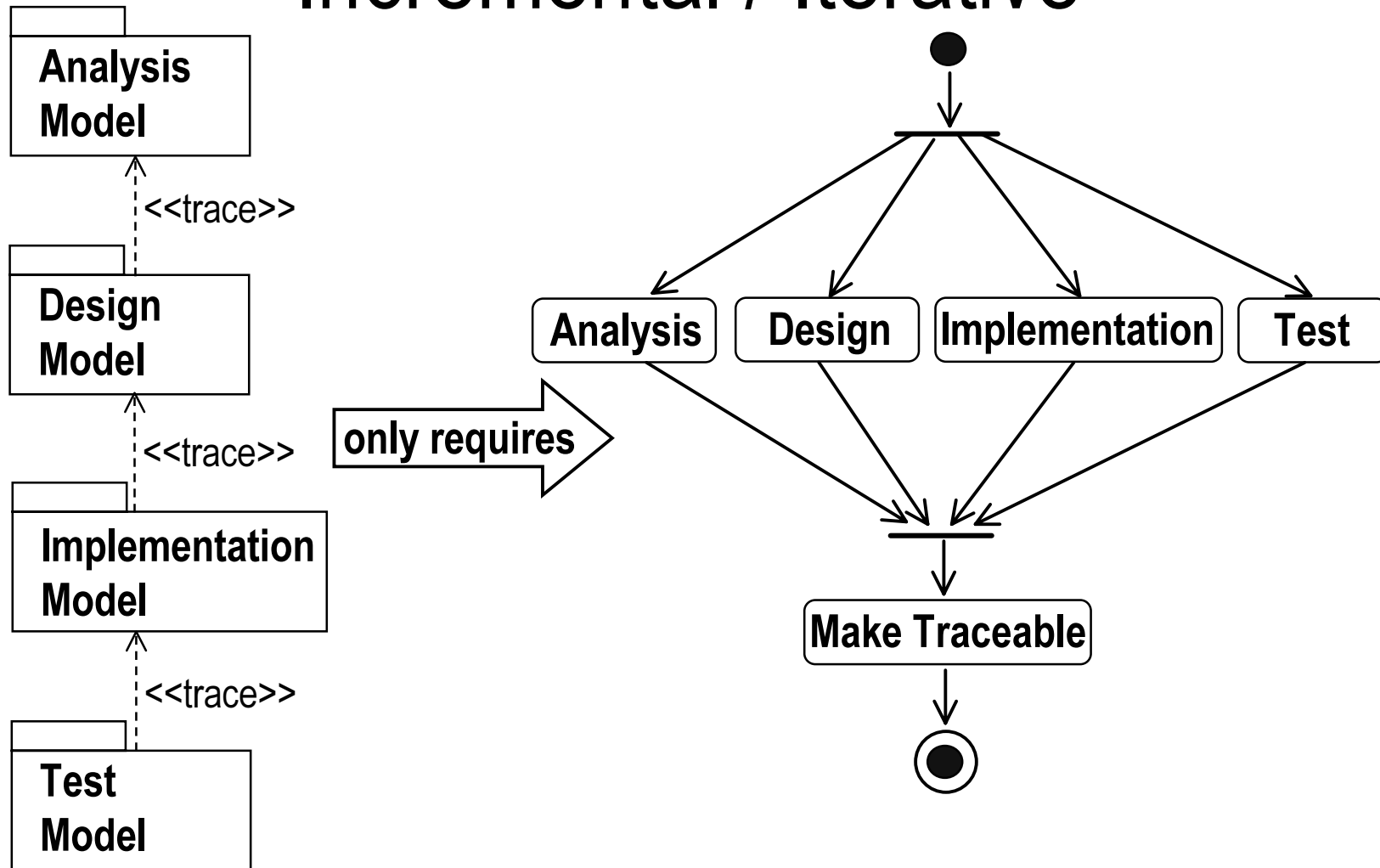
BUT:

Often interpreted as a miniature waterfall





Incremental / Iterative





Refactoring

- Flattening the Workers
- Dissolving the Workflows
- Sorting out the Artifacts





Flattening the Workers

- USDP Roles = Responsibilities
- USDP Workers = Position where specific responsibilities are required



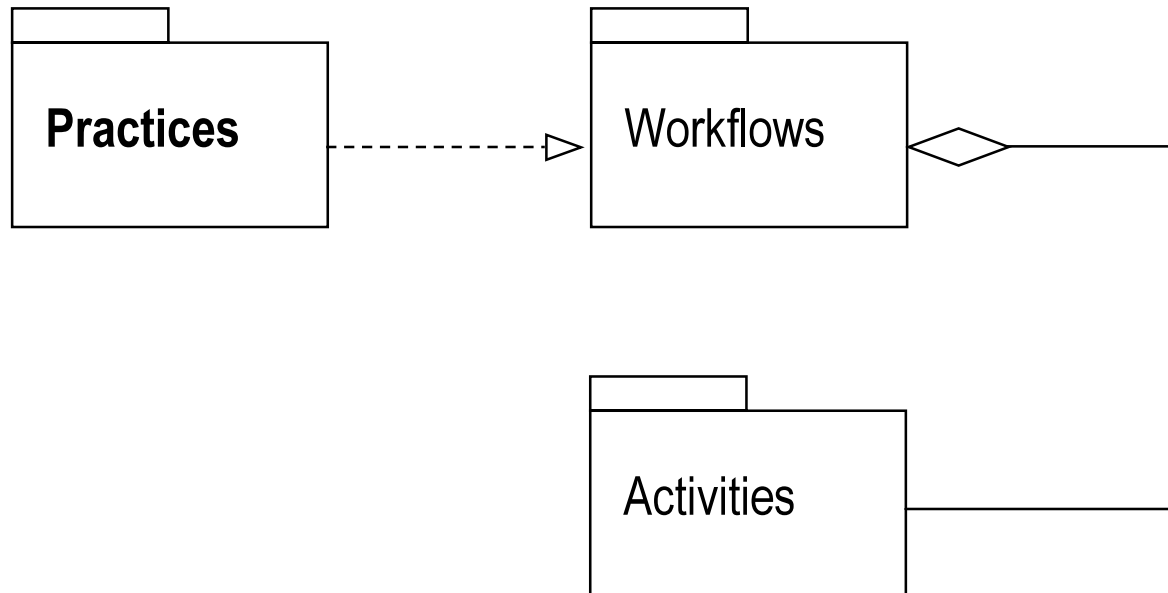
THEREFORE:

Most of XP roles equal USDP workers, e.g. Customer, Tracker, Coach, Big Boss





Dissolving the Workflows





Sorting out the Artifacts

- USDP artifacts: 6 high level models
 - Use Case, Analysis, Design, Deployment, Implementation, Test

vs.

- Travel Light

BUT:

Each artifact can be produced





Conclusion



USDP provides a framework in which XP can live

– Commonalities:

- people are crucial
- Use Case centered
- incremental/iterative



– No (real) contradiction

- phases
- artifacts



BUT: Different Driving Forces!