

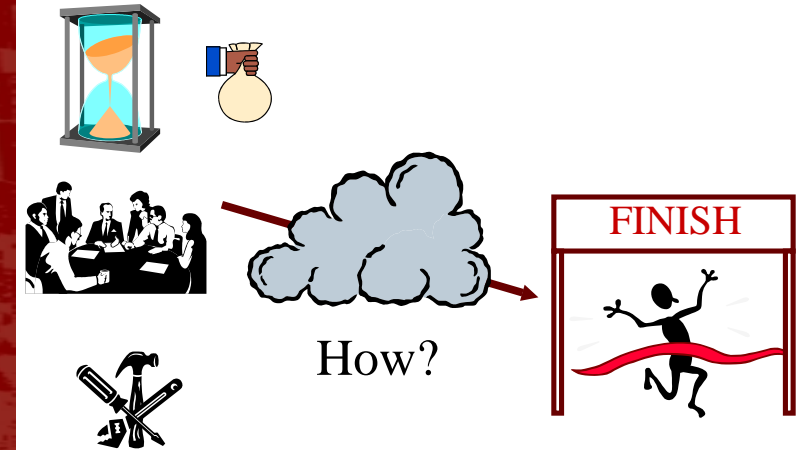


## Work Models

*Solving Problems Using People*



## Achieving a goal



PVK--HT00

Copyright © 1997-1999, jubo@cs.umu.se/epltos@epl.ericsson.se

2



## Elements of Software Engineering

- ◆ **Methods**  
Technical “how tos” to support software development tasks
  - ◆ **Languages**  
Notations to support methods
  - ◆ **Tools**  
(Semi-) automated support for (the usage of) methods and languages
  - ◆ **Processes**  
Coordination and management of software development tasks supported by methods, languages, and tools
- ➔ Economically produce quality software

PVK--HT00

Copyright © 1997-1999, jubo@cs.umu.se/epltos@epl.ericsson.se

3



## The Work Model: A pattern of steps that are used to approach a goal.



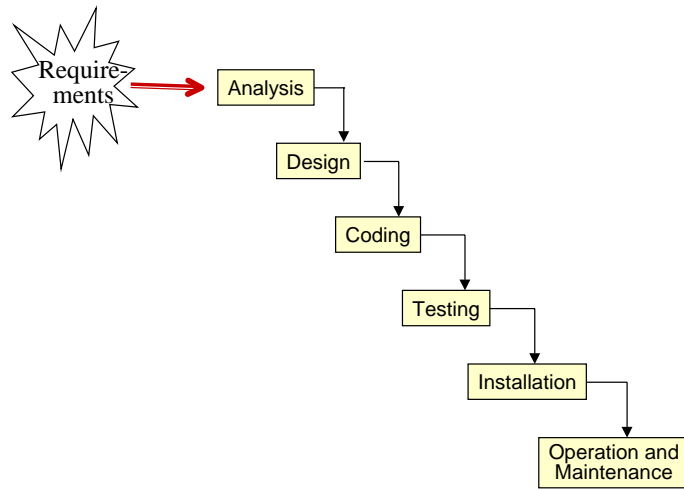
PVK--HT00

Copyright © 1997-1999, jubo@cs.umu.se/epltos@epl.ericsson.se

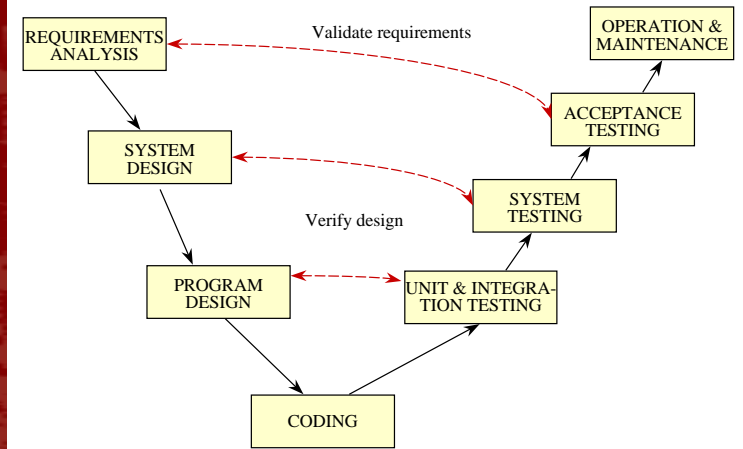
4



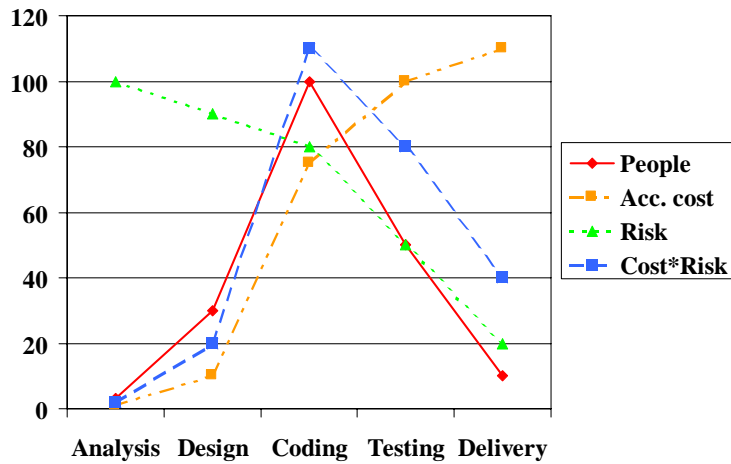
# The Waterfall Model ('70)



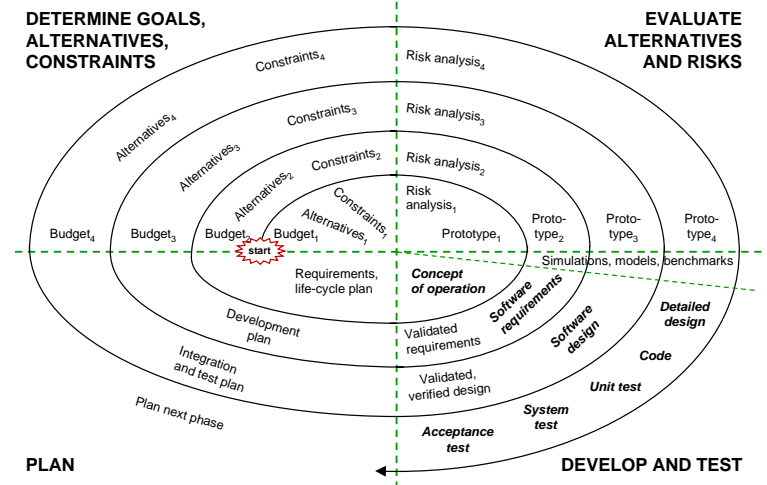
# V Model



# The Waterfall Model



# The Spiral Model ('88)



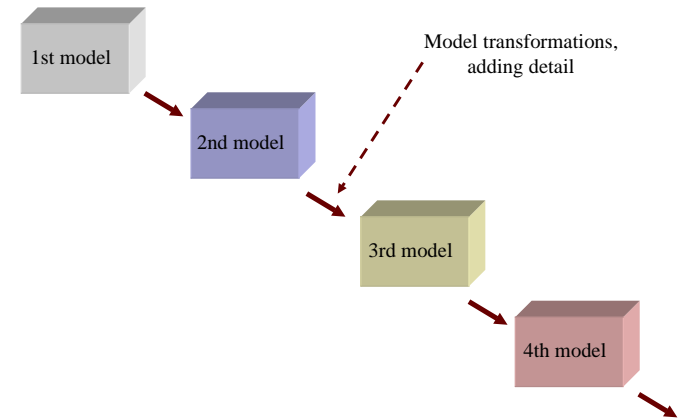


# Waterfall vs. Spiral Model

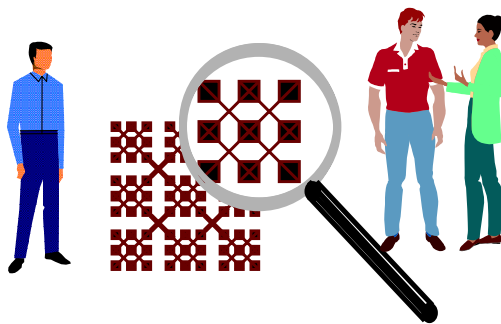
	Waterfall Model	Spiral Model
<b>Model Complexity</b>	Simple, linear sequence of phases	Complex, iterative model; many integrated tasks
<b>Management</b>	Document driven	Risk driven
<b>Quality Control</b>	Natural milestone after each phase	Continuous evaluation, integrated into the model
<b>Customer interaction</b>	No	Prototypes are built and evaluated by customers in every iteration
<b>Risk</b>	High (late feedback)	Low (risk analysis is integrated in the model)
<b>Usability</b>	Small and/or low risk projects	Large projects



# Stepwise Refinement



# Different Focus



# Roles

- ◆ Requirements analyst
- ◆ Architect
- ◆ Function designer
- ◆ Unit designer
- ◆ Unit tester
- ◆ .....



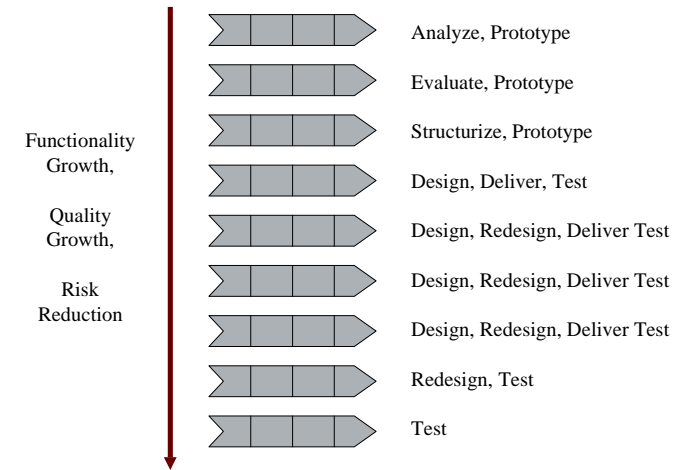
# The Development Cycle



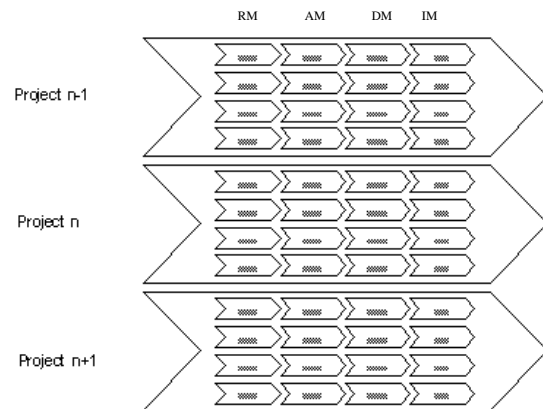
Different phases used to focus on different types of problems



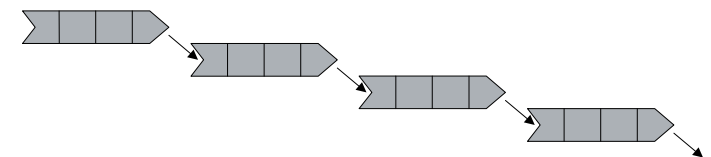
# Incremental Development



# Projects vs Iterations



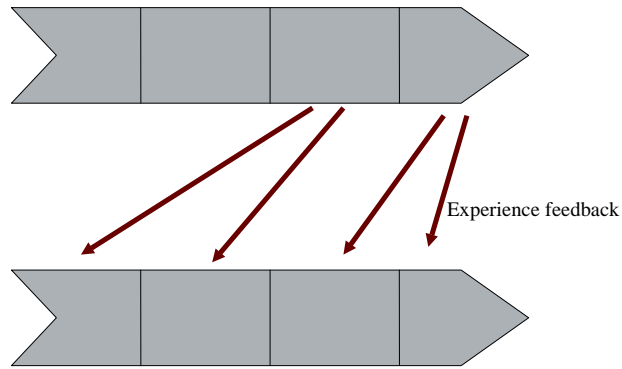
# Small Projects: One Phase at a Time



How to keep people busy?



# Evolutionary Development

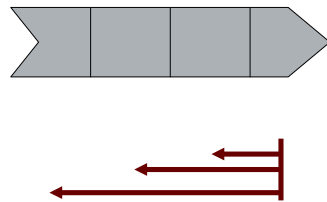


# eXtreme Programming

- ◆ One room
- ◆ On-site customer
- ◆ Write tests before programs
- ◆ Collective ownership
- ◆ Continuous refactoring
- ◆ Pair programming
- ◆ 24 hour development cycles
- ◆ .....



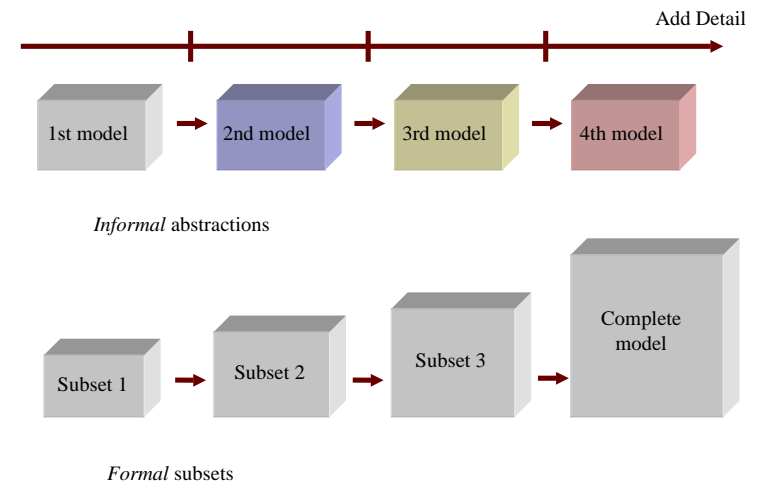
# Roundtrip Engineering



Refinement decision "back pressure".  
*Decisions in later phases replace previous decisions.*

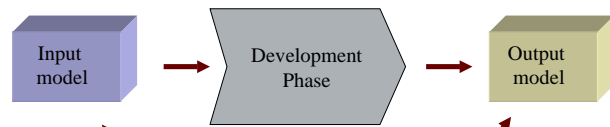


# Abstraction vs Information Hiding





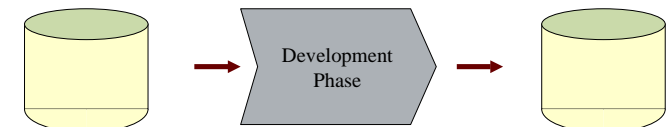
# “Document Based” Development



Discrete sets of atomic configuration items (documents), modeling the system at different abstraction levels



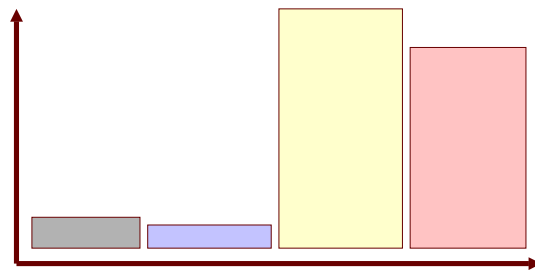
# “Model Based” Development



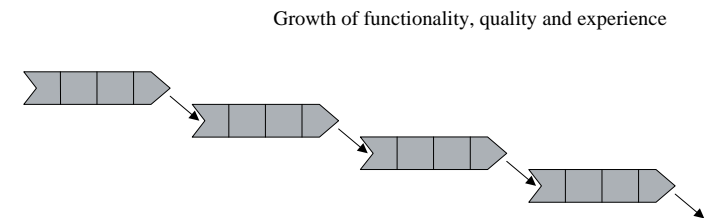
Two revisions of the same information set, where input is a subset of output



# Larger Projects: Phases and People

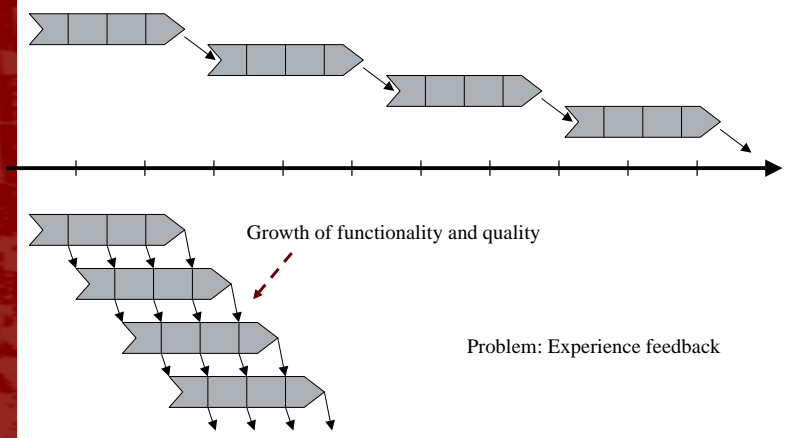


# Sequential Increments

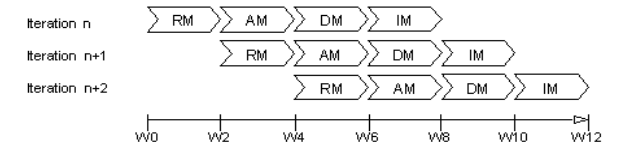




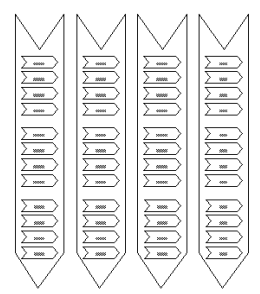
# Shortening the TTM



# Overlapping Iterations



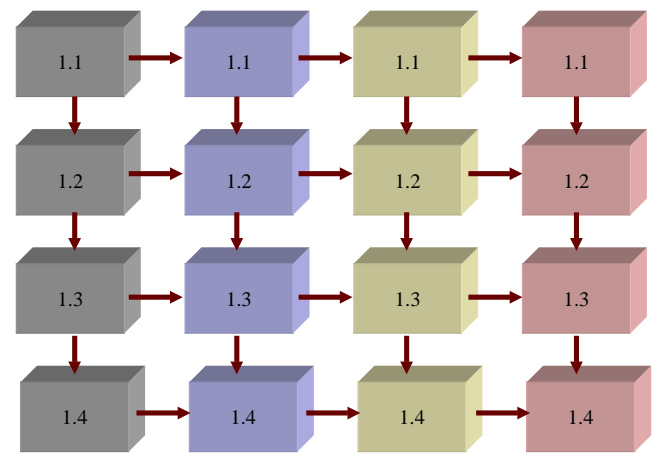
# Model Process Roles



- Analyst
- Architect
- Designer



# “Document Based” Development with Overlapping Iterations





# “Model Based” Development with Overlapping Iterations ??

