

# The Unified Process

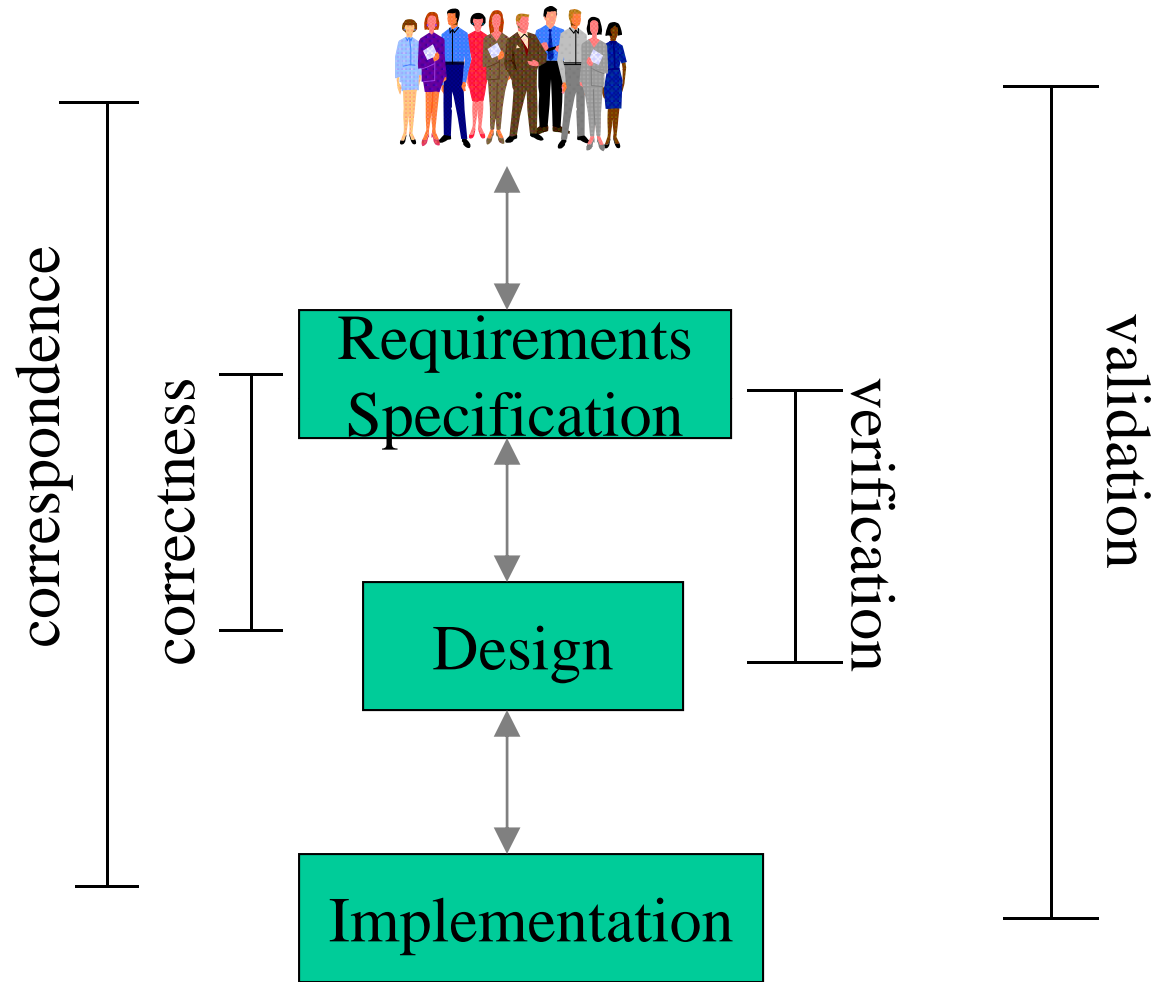
Robert B. France

Dept. of Computer Science

Colorado State University

# **PROCESS REVIEW**

# Essence of Software Development



# Validation vs. Verification

- Validation is concerned with establishing that a design or an implementation satisfies users
  - Are we building the right thing?
- Verification is concerned with establishing that a development artifact (e.g., design or code) satisfies formal specifications
  - Did we build it right?

# What is a Process and why do we need a systematic process?

- A software process is a sequence of steps required to develop or maintain software (Watts Humphrey)
- A process is a series of steps involving activities, constraints, and resources that produce an intended output
- Good people + good process = lower risk of project failure

# Evolution of Software Process Models

- “Code and Fix” Model
- Waterfall model
- Spiral model
- Incremental/iterative agile processes
- Unified process

# Choosing a Model

- Choice depends on nature of project:
  - Are requirements clearly defined and stable?
  - Is there pressure to produce a working product quickly?
  - Are the consequences of operational errors serious?

# Radical vs. Conservative Models

- More radical models suitable when:
  - quick results are needed
  - requirements are fuzzy or unstable
- More conservative models suitable when:
  - consequences of errors are very serious
  - requirements are well-understood and stable

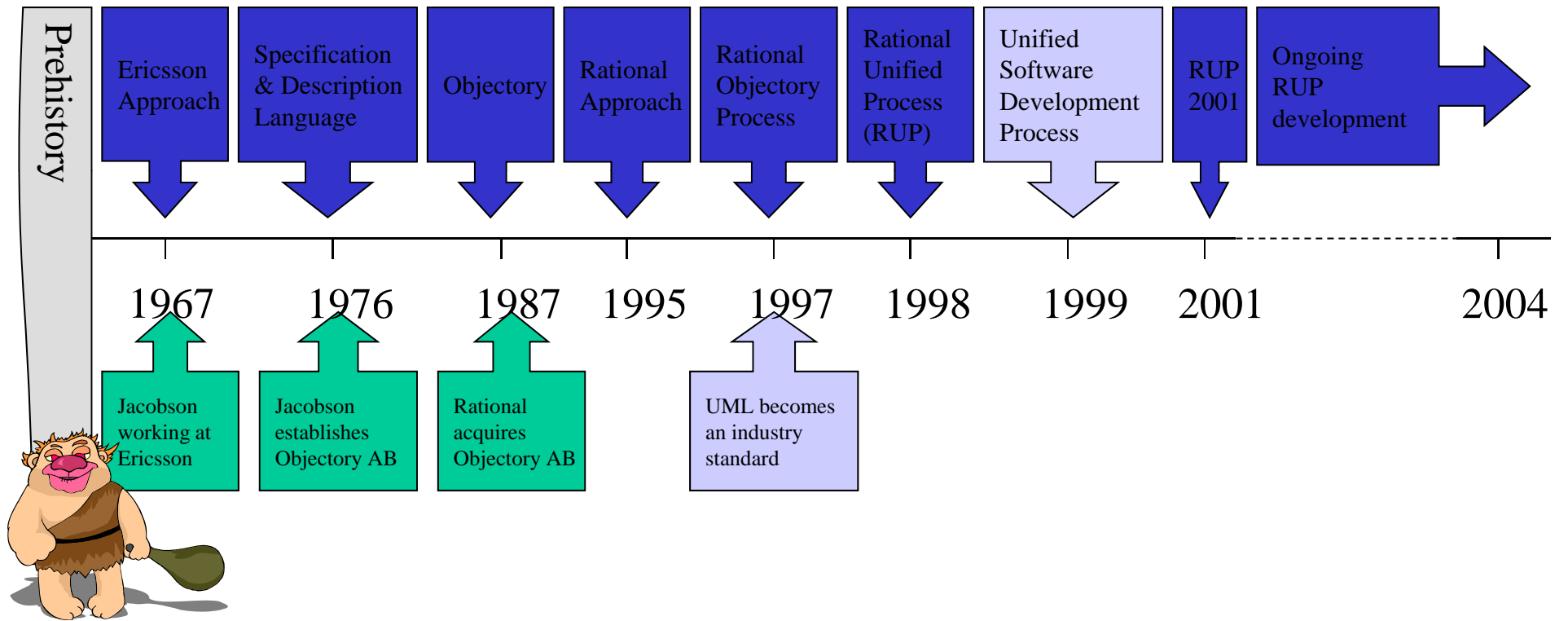


Unified Process:  
Introduction to an Iterative,  
Incremental OOA/D Process

# The Unified Process (UP)

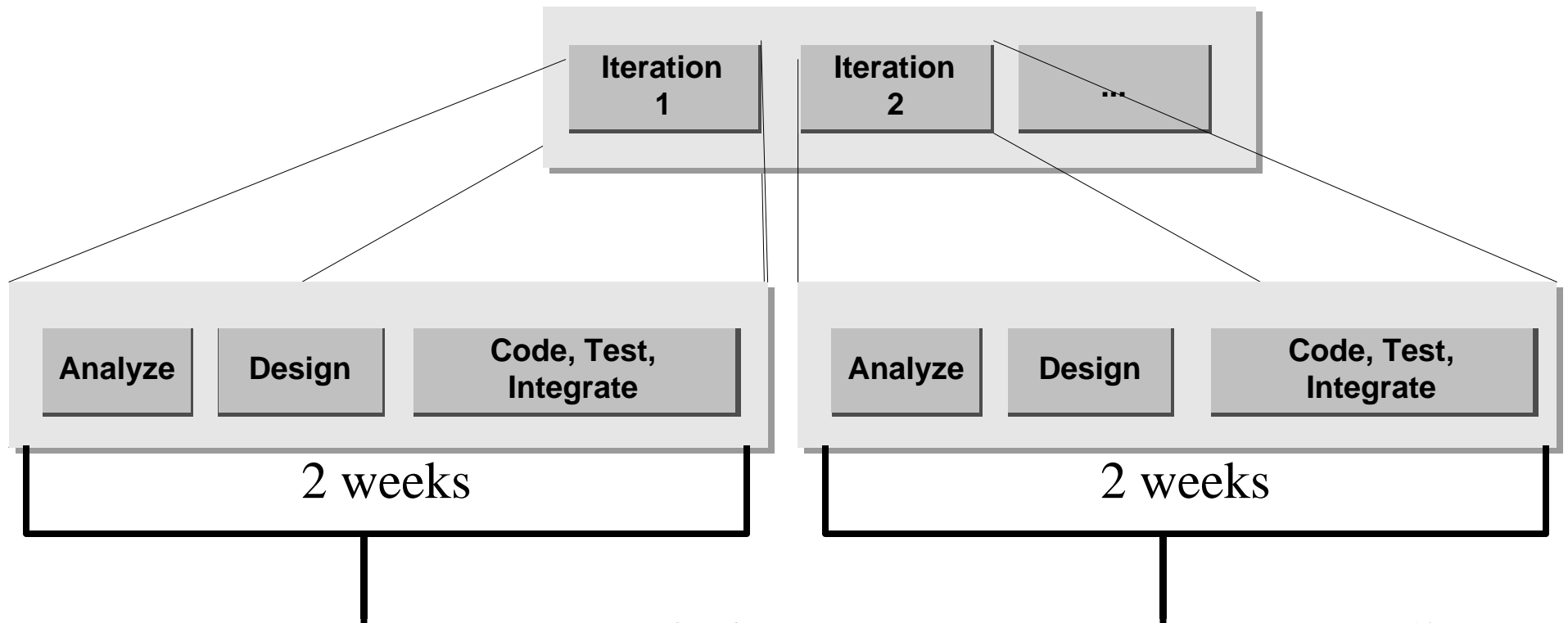
- The Unified Process is an industry standard software engineering process
  - It is the generic process for the UML
  - It is free - described in "The Unified Software Development Process", ISBN:0201571692"
- UP is:
  - Use case (requirements) driven
  - Iterative and incremental
- UP is a generic software engineering process. It has to be customised (instantiated) for your project
  - In house standards, document templates, tools, databases, lifecycle modifications, ...
- Rational Unified Process (RUP) is an instantiation of UP
  - RUP is a product marketed and owned by IBM
  - RUP also has to be instantiated for your project

# UP history



# UP Basics

- Small steps, feedback and evolution
- Iterative, incremental, time-boxed
- Risk-driven



# Key Practices

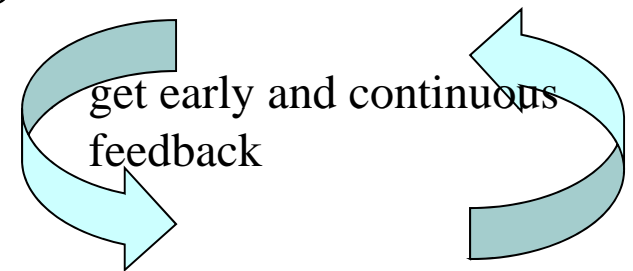
- Deliver product in increments developed in iterations
- Iterations are carried out in a fixed time
  - Developers can choose to drop features but should not extend iteration
- High risk and high value aspects tackled in early iterations
  - Cohesive architecture implemented in early iterations
- Customers continuously engaged in evaluation, feedback and requirements elicitation
- Continuously verify quality; test-driven code development
- Model software

# Motivating Time-boxing

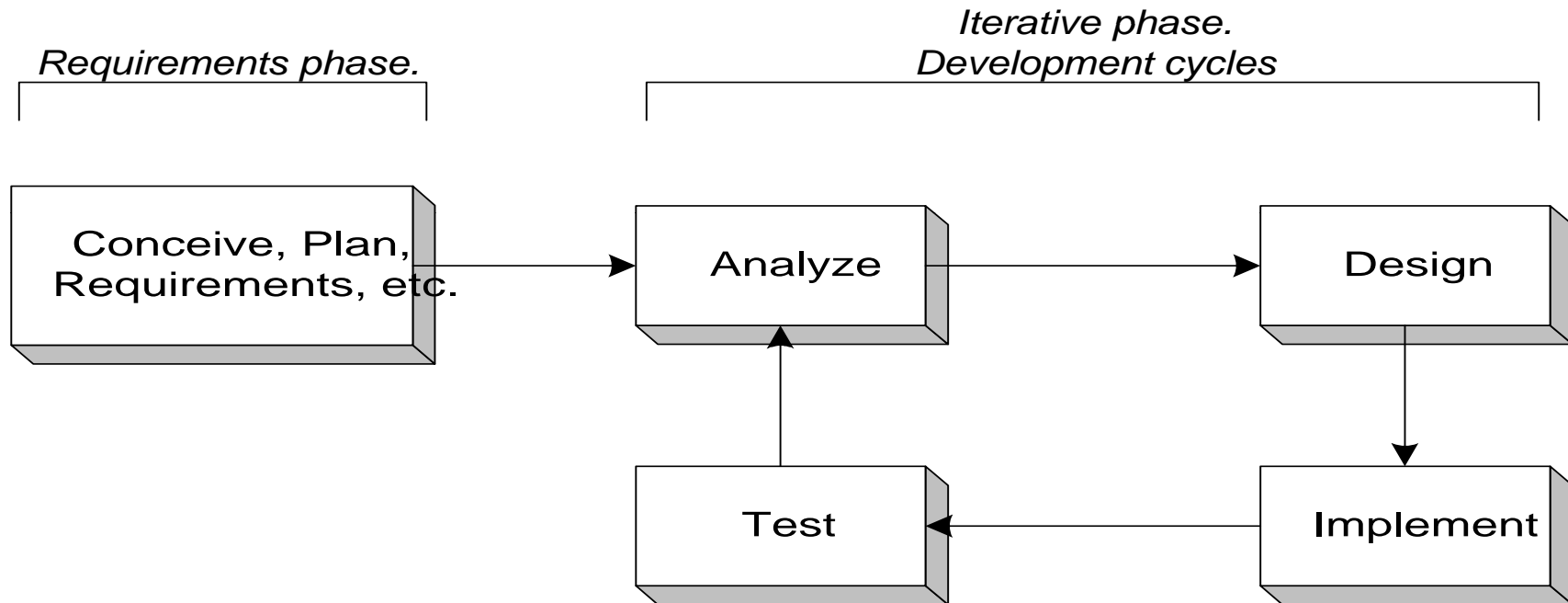
- “Work expands so as to fill the time available for completion” (Parkinson’s Law)
- Forces prioritization of tasks and risks
- Gain confidence of customers
- Build team confidence/satisfaction

# Iterations

- Iterations are the key to the UP
- Each iteration is like a mini-project including:
  - Planning
  - Analysis and design
  - Integration and test
  - An internal or external release
- We arrive at a final product release through a sequence of iterations
- Iterations can overlap - this allows parallel development and flexible working in large teams
  - Requires careful planning
- Iterations are organised into phases

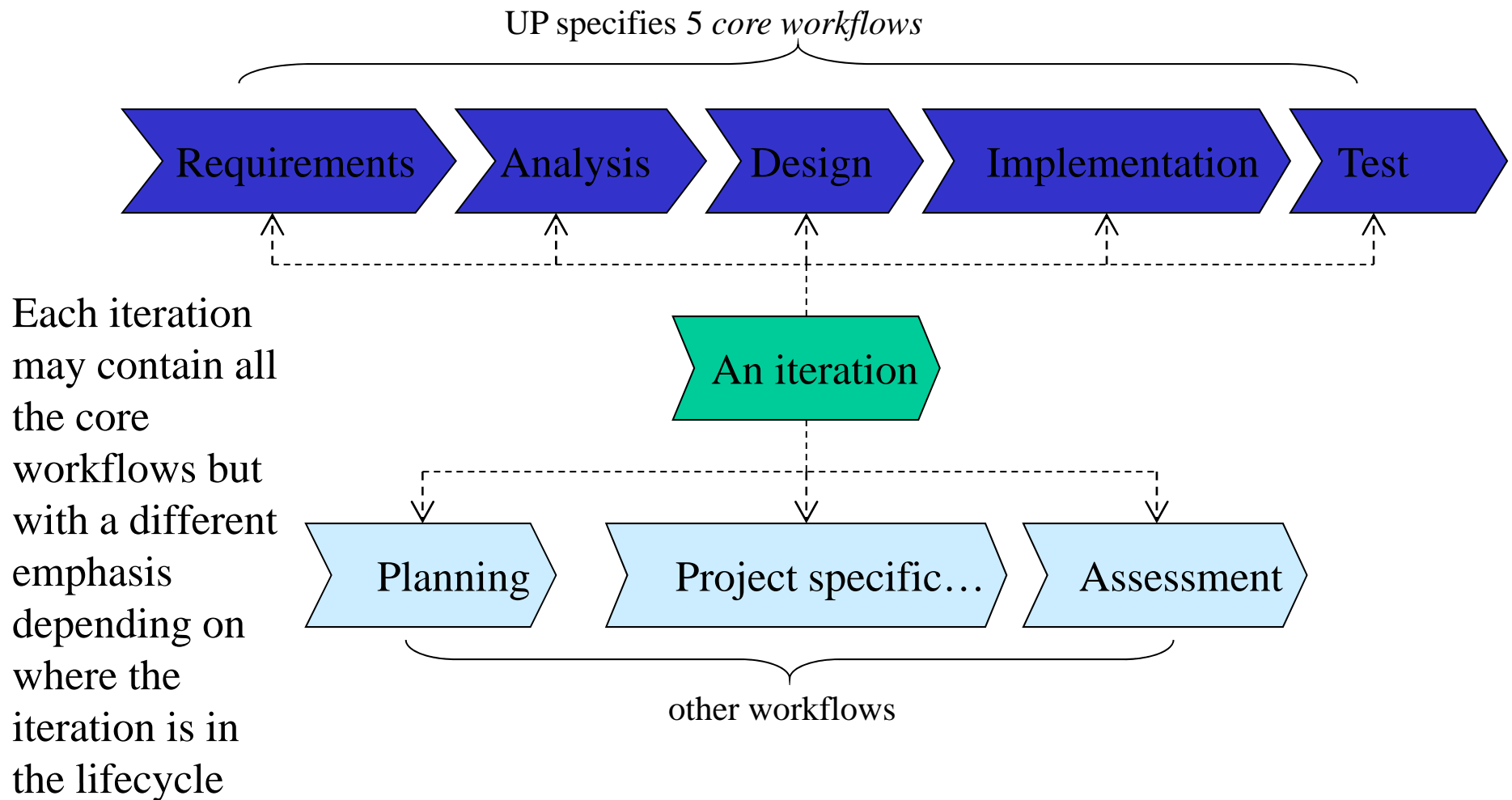


# Iteration Structure





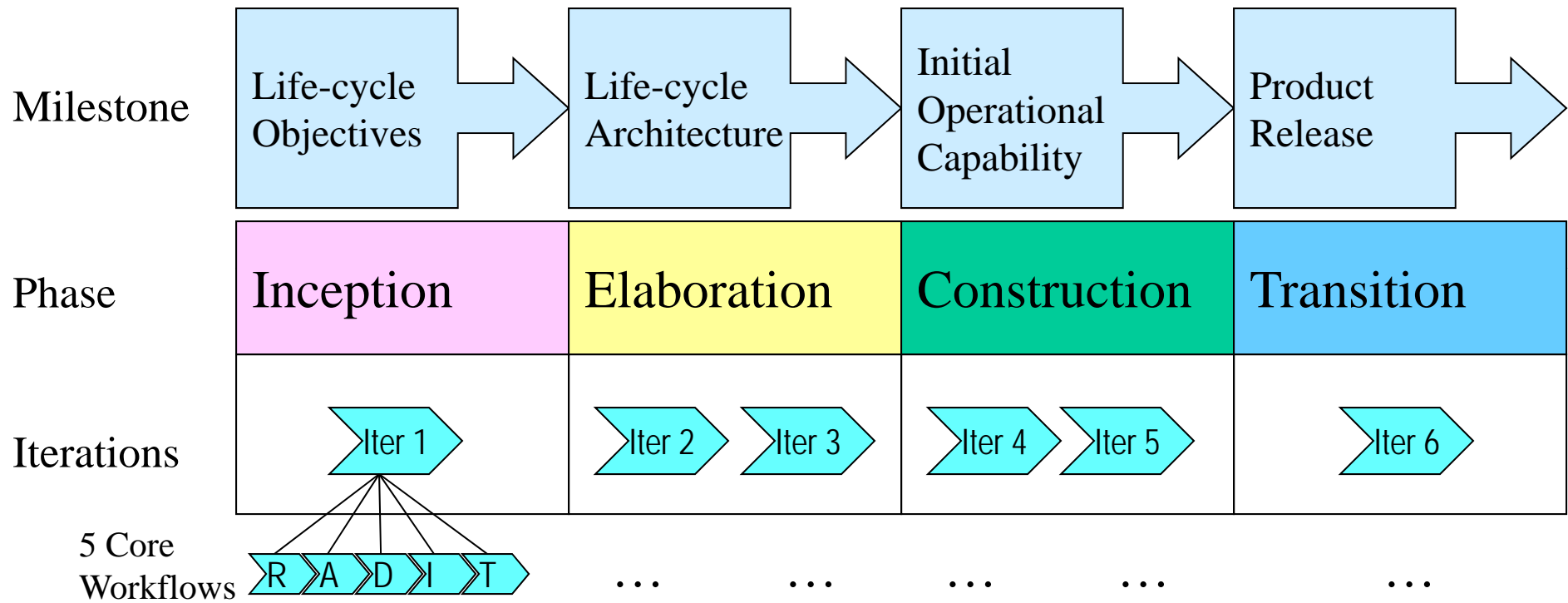
# Iteration workflows



# Baselines and increments

- Each iteration generates a baseline
- A baseline is a set of reviewed and approved artefacts that:
  - Provide an agreed basis for further review and development
  - Can be changed only through formal procedures such as configuration and change management
- An *increment* is the difference between the baseline generated by one iteration and the baseline generated by the next iteration
  - This is why the UP is called “iterative and incremental”

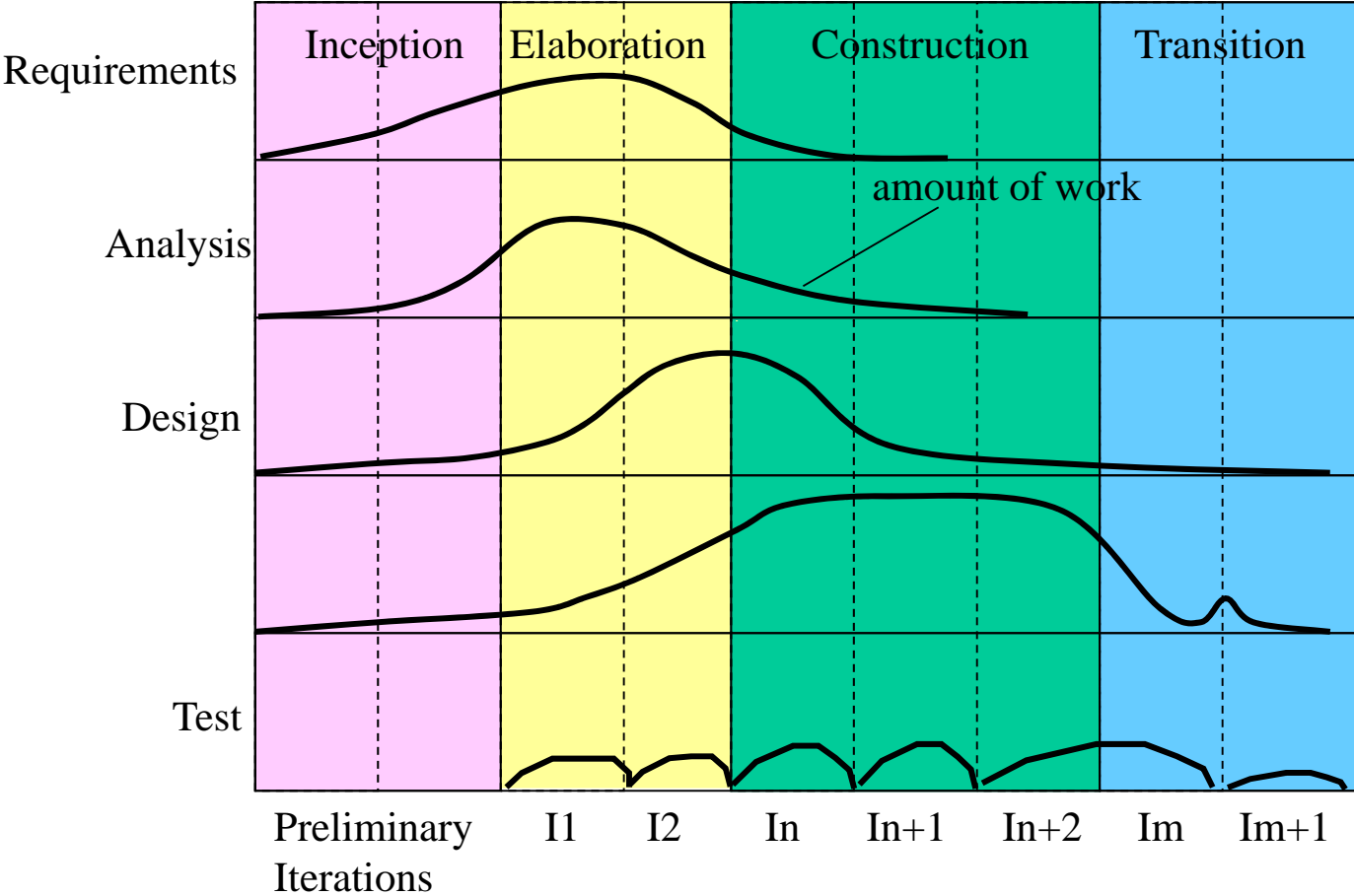
# UP Structure



- Each phase can include several iterations
  - The exact number of iterations per phase depends on the size of the project! e.g. one iteration per phase for small projects
- Each phase concludes with a major milestone

# Phases and Workflows

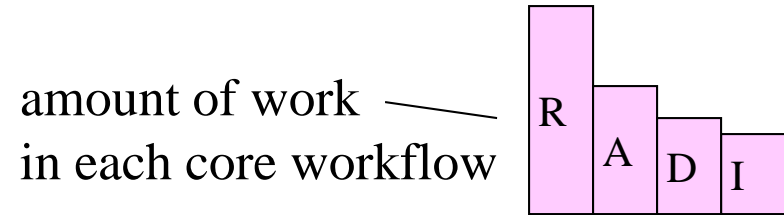
- For each phase we will consider:
- The focus in terms of the core workflows
- The goal for the phase
- The milestone at the end of the phase



# Phases

- Inception
  - Early exploration of problem to determine project feasibility
  - What's the perceived business value?
  - What are the risks?
- Elaboration
  - Requirements detailing (major requirements identified)
  - Iterative implementation of “core” architecture
- Construction
  - Iterative development of remaining low-risk elements
  - Prepare for deployment
- Transition
  - Beta tests
  - Deployment

# Inception

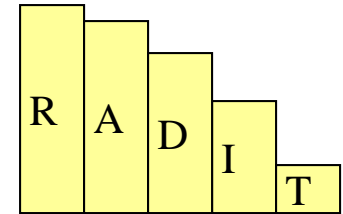


		Inception	Elaboration	Construction	Transition
Focus	Requirements – establish business case and scope. Capture core requirements	[Graph showing high activity in Inception, tapering off in subsequent phases]			
	Analysis – establish feasibility	[Graph showing activity starting in Inception, peaking in Elaboration, and tapering off]			
	Design – design proof of concept or technical prototypes	[Graph showing activity starting in Elaboration, peaking in Construction, and tapering off]			
	Implementation – build proof of concept or technical prototype	[Graph showing activity starting in Construction, peaking in Transition, and tapering off]			
	Test – not generally applicable	[Graph showing low, sporadic activity across all phases]			
Goals	Establish feasibility of the project - create proof of concept/technical prototypes Create a business case Scope the system - capture key requirements Identify critical risks				

## Inception - milestone

- Life Cycle Objectives - conditions of satisfaction:
  - System scope has been defined
  - Key requirements for the system have been captured. These have been defined and agreed with the stakeholders
  - An architectural vision exists. This is just a sketch at this stage
  - A Risk Assessment
  - A Business Case
  - Project feasibility is confirmed
  - The stakeholders agree on the objectives of the project

# Elaboration



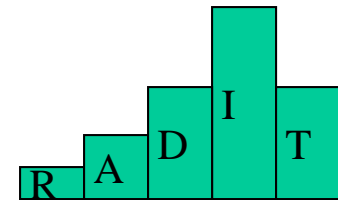
		Inception	Elaboration	Construction	Transition
Focus	Requirements – refine system scope and requirements				
	Analysis – establish what to build				
	Design – create a stable architectural baseline				
	Implementation – build the architectural baseline				
	Test – test the architectural baseline				
Goals	Create an executable architectural baseline				
	Refine Risk Assessment and define quality attributes (defect rates etc.)				
	Capture use cases to 80% of the functional requirements				
	Create a plan with sufficient detail for the construction phase				
	Formulate a bid which includes resources, time, equipment, staff, cost				



## Elaboration - milestone

- Lifecycle Architecture - conditions of satisfaction:
  - A resilient, robust executable architectural baseline has been created
  - The Risk Assessment has been updated
  - A project plan has been created to enable a realistic bid to be formulated
  - The business case has been verified against the plan
  - The stakeholders agree to continue

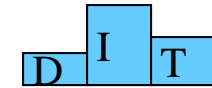
# Construction



	Inception	Elaboration	Construction	Transition	
Focus	Requirements – uncover any requirements that had been missed	[Line graph showing a curve that rises in Inception, peaks in Elaboration, and then gradually declines through Construction and Transition]			
	Analysis – finish the analysis model	[Line graph showing a curve that rises in Inception, peaks in Elaboration, and then declines through Construction and Transition]			
	Design – finish the design model	[Line graph showing a curve that rises in Inception, peaks in Elaboration, and then declines through Construction and Transition]			
	Implementation – build the Initial Operational Capability	[Line graph showing a curve that rises in Inception, peaks in Elaboration, and then declines through Construction and Transition]			
	Test – test the Initial Operational Capability	[Line graph showing a curve that rises in Inception, peaks in Elaboration, and then declines through Construction and Transition]			
Goals	Complete use case identification, description and realization Finish analysis, design, implementation and test Maintain the integrity of the system architecture Revise the Risk Assessment				

# Construction - milestone

- Initial Operational Capability - conditions of satisfaction:
  - The product is ready for beta testing in the user environment



# Transition

	Inception	Elaboration	Construction	Transition
Focus	Requirements – not applicable			
	Analysis – not applicable			
	Design – modify the design if problems emerge in beta testing			
	Implementation – tailor the software for the user site. Fix bugs uncovered in beta testing			
	Test – perform beta testing and acceptance testing at the user site			
Goals	Correct defects			
	Prepare the user site for the new software and tailor the software to operate at the user site			
	Modify software if unforeseen problems arise			
	Create user manuals and other documentation			
	Provide customer consultancy			
	Conduct post project review			

# Transition – milestone

- Product Release - conditions of satisfaction:
  - Beta testing, acceptance testing and defect repair are finished
  - The product is released into the user community

# Summary

- UP is a risk and use case driven, architecture centric, iterative and incremental software development process
- UP has four phases:
  - Inception
  - Elaboration
  - Construction
  - Transition
- Each iteration has five core workflows:
  - Requirements
  - Analysis
  - Design
  - Implementation
  - Test