OO Analysis and Design with UML 2 and UP

Dr. Jim Arlow,

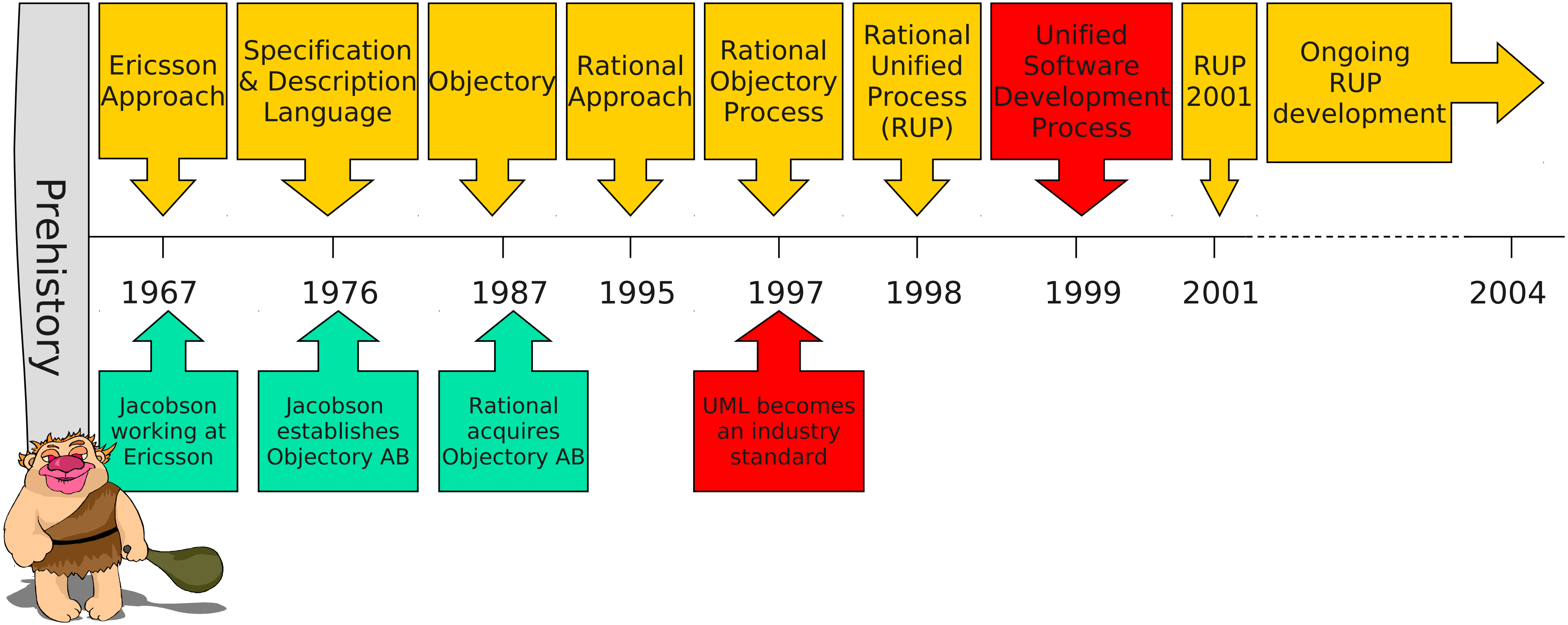
Zuhlke Engineering Limited

Introduction to the Unified Process

The Unified Process (UP)

- The Unified Software Development Process is an industry standard software engineering process
  - It is commonly referred to as the "Unified Process" or UP
  - 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
  - Risk driven
  - Architecture centric
  - 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 Rational Corporation
  - RUP also has to be instantiated for your project

UP history



Prehistory

Ericsson Approach

Specification & Description Language

Objectory

Rational Approach

Rational Objectory Process

Rational Unified Process (RUP)

Unified Software Development Process

RUP 2001

Ongoing RUP development

1967

1976

1987

1995

1997

1998

1999

2001

2004

Jacobson working at Ericsson

Jacobson establishes Objectory AB

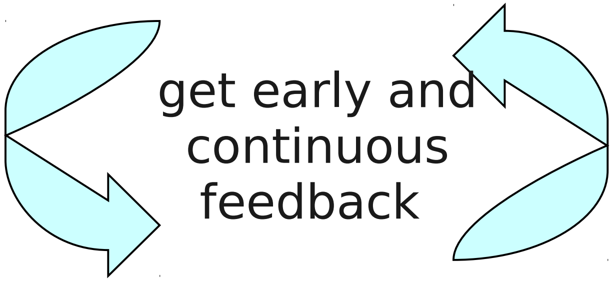
Rational acquires Objectory AB

UML becomes an industry standard

2.7

Iterations

- Iterations are the key to the UP
- Each iteration is like a mini-project including:
  - Planning
  - Analysis and design
  - Implementation
  - Integration and test
  - An internal or external release

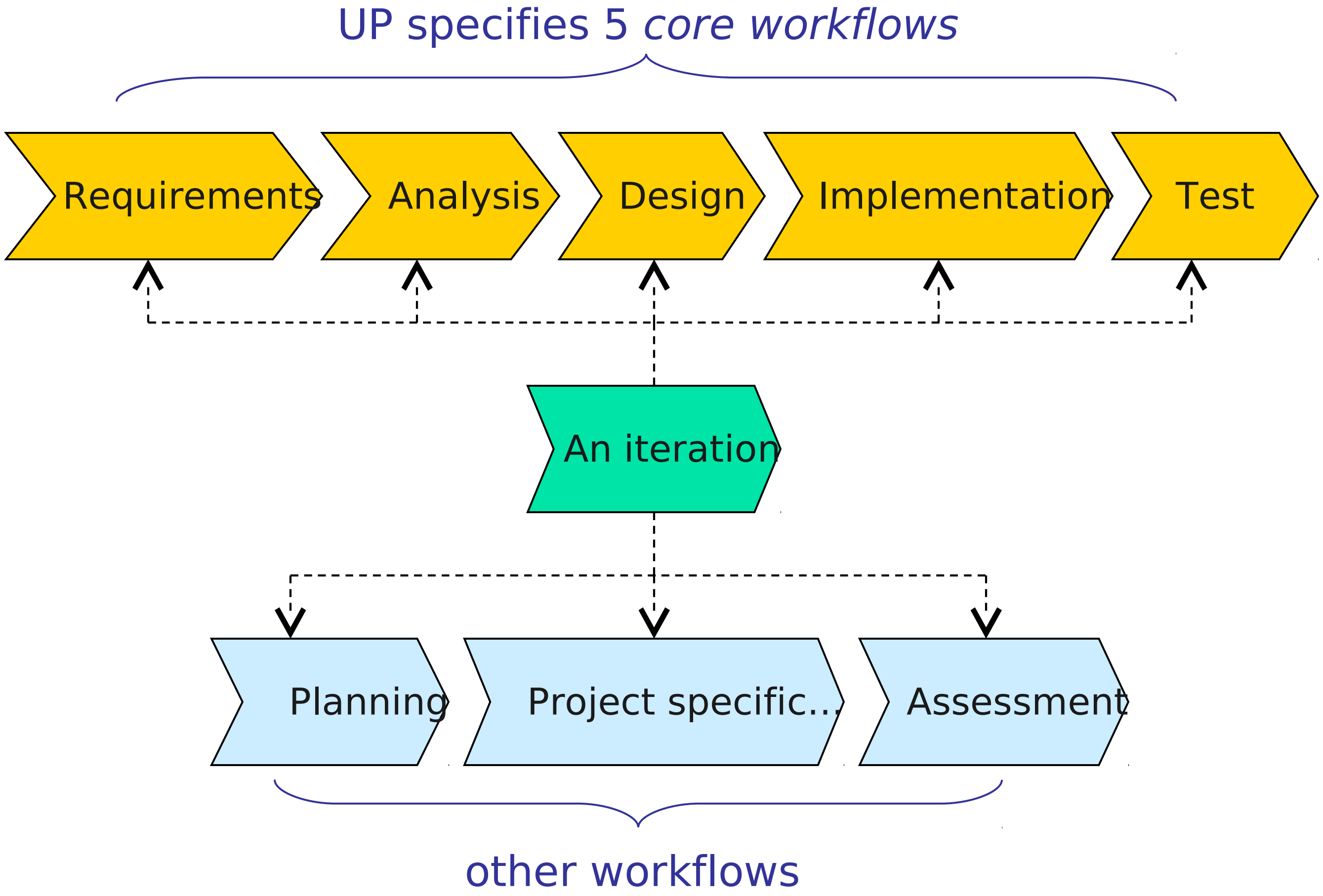


get early and continuous feedback

- 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 workflows

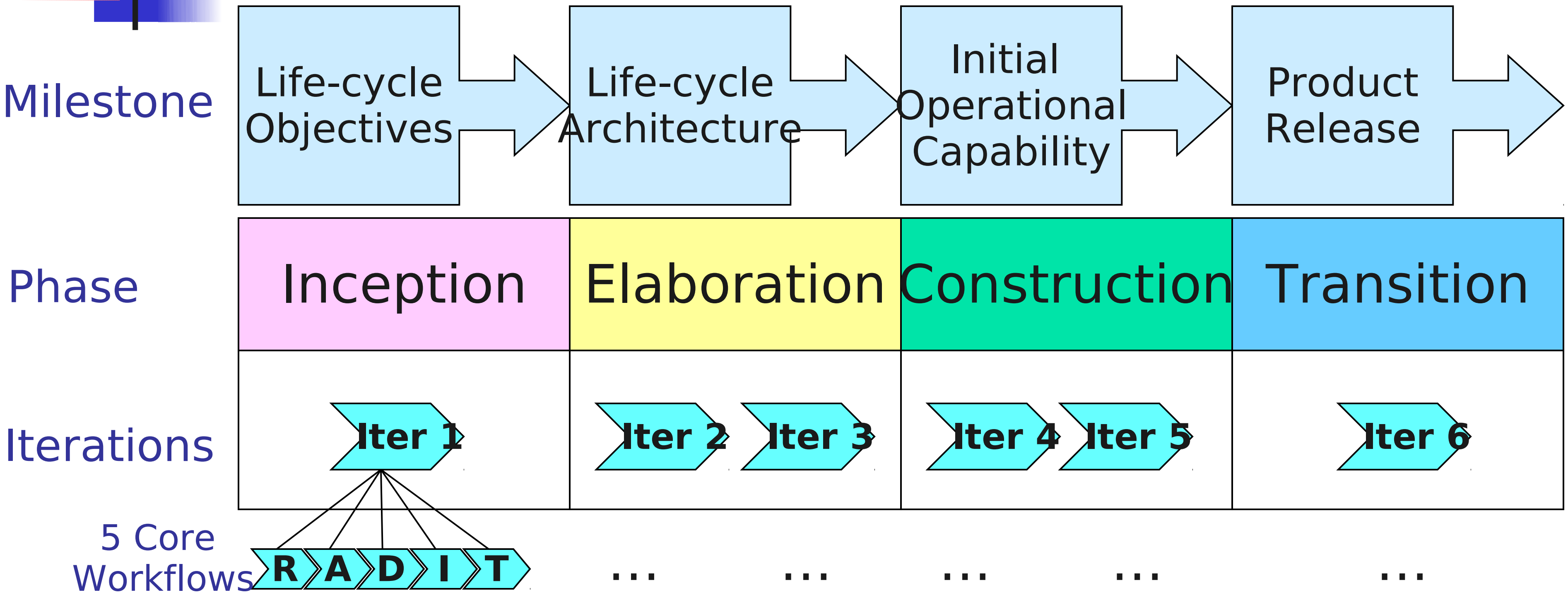
- Each iteration may contain all of the core workflows but with a different emphasis depending on where the iteration is in the lifecycle



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

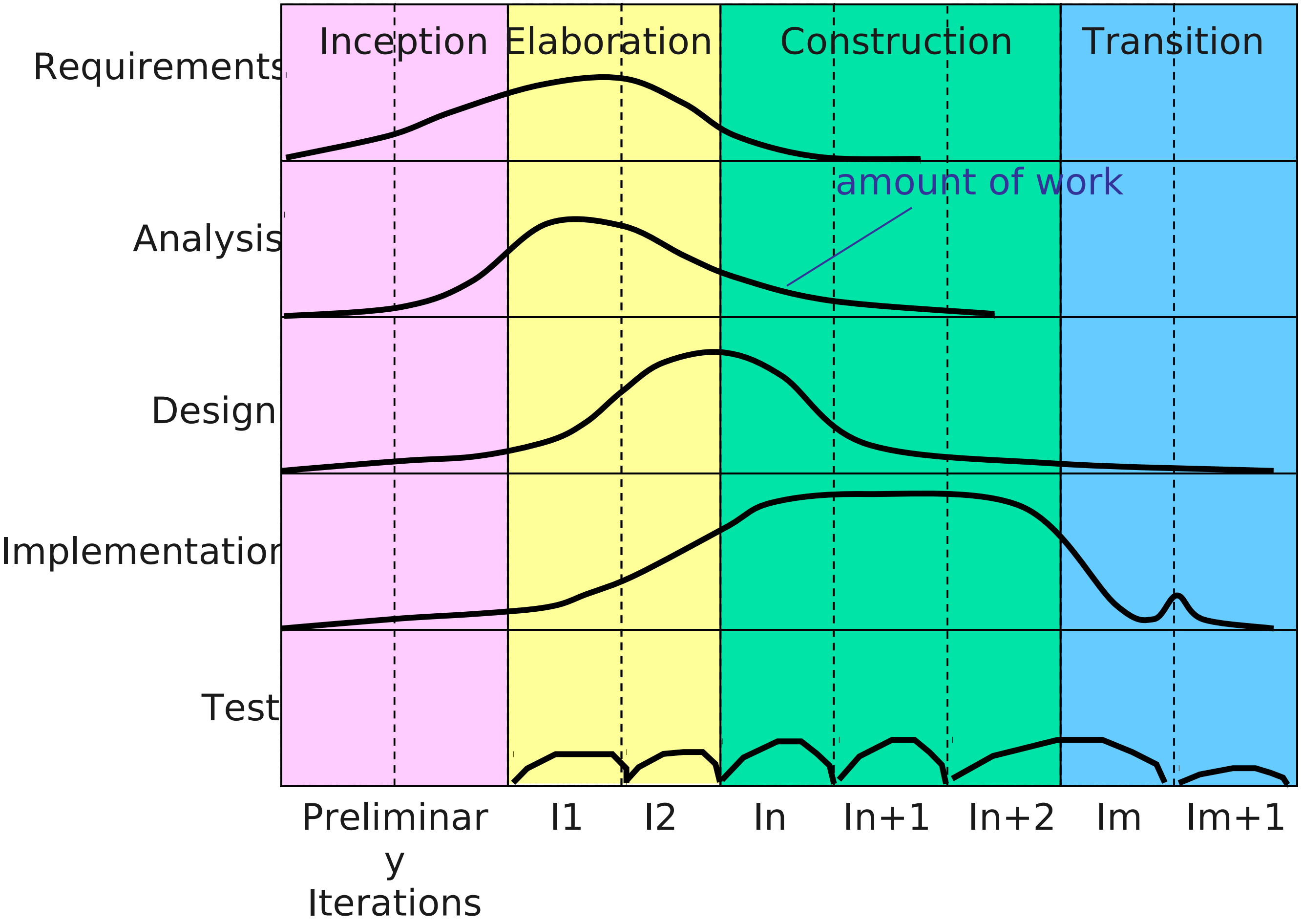


Milestone	Life-cycle Objectives	Life-cycle Architecture	Initial Operational Capability	Product Release
Phase	Inception	Elaboration	Construction	Transition
Iterations	Iter 1	Iter 2 Iter 3	Iter 4 Iter 5	Iter 6
5 Core Workflows	R A D I T	… …	… …	…

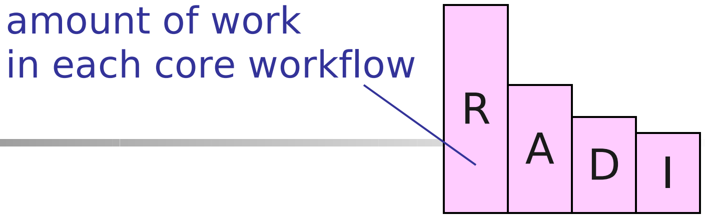
- Each phase can include several iterations
    - The exact number of iterations per phase depends on the size of the project! e.g. one iteration per phase for small projects
- Each phase concludes with a major milestone

Phases and Workflows

- This figure is the key to understanding UP!
- 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

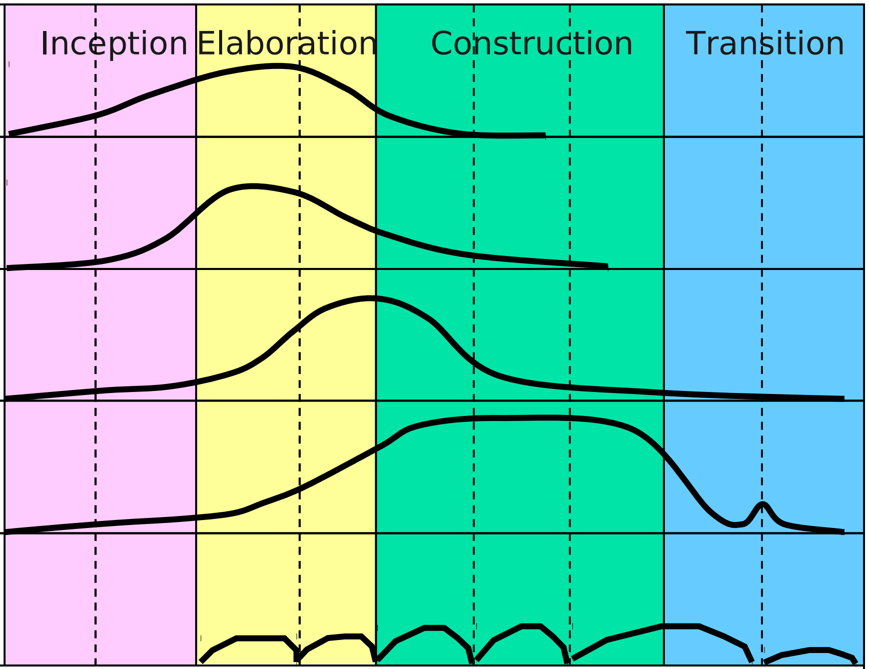


Inception



amount of work
in each core workflow

Focus	**Requirements** – establish business case and scope. Capture core requirements
	Analysis – establish feasibility
	Design – design proof of concept or technical prototypes
	Implementation – build proof of concept or technical prototype
	Test – not generally applicable
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

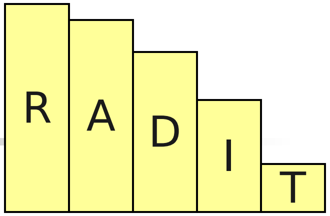


Inception - milestone

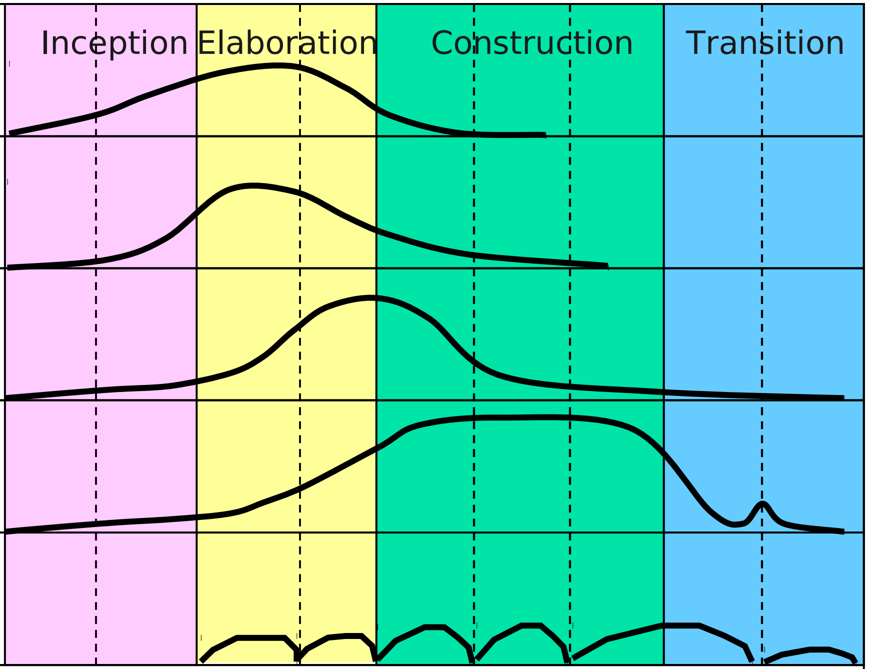
- Life Cycle Objectives - conditions of satisfaction:
  - System scope has been defined
  - Key requirements for the system have been captured. These have been defined and agreed with the stakeholders
  - An architectural vision exists. This is just a sketch at this stage
  - A Risk Assessment
  - A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

2.9.4

Elaboration



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

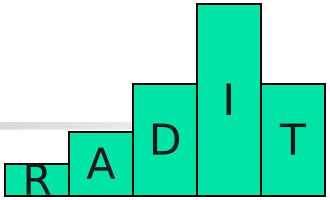


Elaboration - milestone

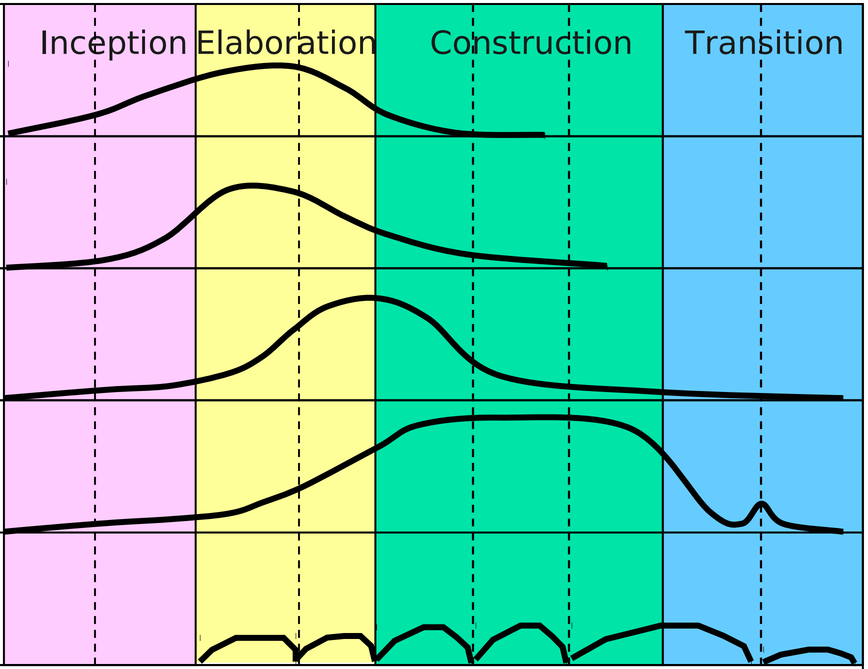
- Lifecycle Architecture - conditions of satisfaction:
  - A resilient, robust executable architectural baseline has been created
  - The Risk Assessment has been updated
  - A project plan has been created to enable a realistic bid to be formulated
  - The business case has been verified against the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

2.9.7

Construction



Focus	**Requirements** – uncover any requirements that had been missed
	Analysis – finish the analysis model
	Design – finish the design model
	Implementation – build the Initial Operational Capability
	Test – test the Initial Operational Capability
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

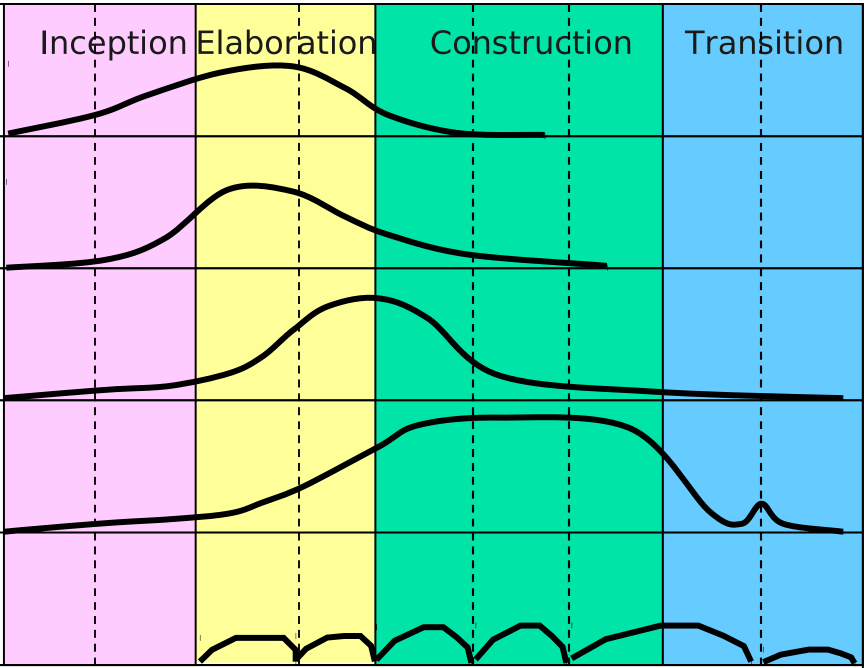


Construction - milestone

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

Transition

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

2.10

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