Unified Software Development Process (3C05/D22)

UCL ComputerScience

Unit 5: USDP

Objectives:
- Introduce the main concepts of iterative and incremental development
- Discuss the main USDP phases

UCL ComputerScience

USDP

- USDP is an industry standard software development process
  - Free!
  - The generic process for the UML
- USDP is:
  - Use-case and risk driven
  - Architecture centric
  - Iterative and incremental
- For reference: Ivar Jacobson, Grady Booch, James Rumbaugh: The Unified Software Development Process. Addison Wesley. 1999

UCL ComputerScience

USDP for your project…

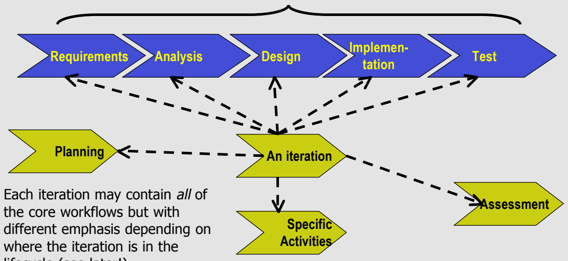
- USDP is a generic software engineering process. It has to be customised (instantiated) for your project:
  - In-house standards
  - Document templates
  - Tools
  - Databases
  - Lifecycle modifications
- Rational Unified Process is an instantiation of USDP. RUP is a product marketed and owned by IBM Software.
- RUP also has to be instantiated for your project!

Iterations

- Iterations are the key to the USDP
- Each iteration is like a mini-project including:
  - Planning
  - Analysis and design
  - Integration and test
  - An internal or external release
  - The result of an iteration is an increment
- We arrive at a final product release through a sequence of iterations
- Iterations contain workflows
- Iterations are organised into phases

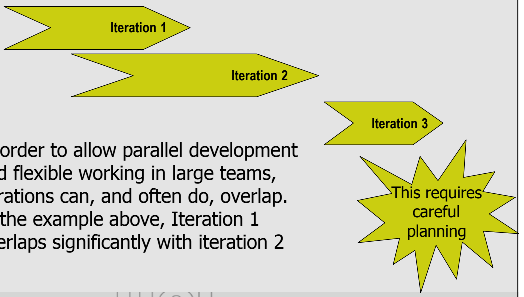
Iteration Workflows

USDP specifies 5 *core workflows*



Each iteration may contain *all* of the core workflows but with different emphasis depending on where the iteration is in the lifecycle (see later!)

Iterations may overlap



In order to allow parallel development and flexible working in large teams, iterations can, and often do, overlap. In the example above, Iteration 1 overlaps significantly with iteration 2

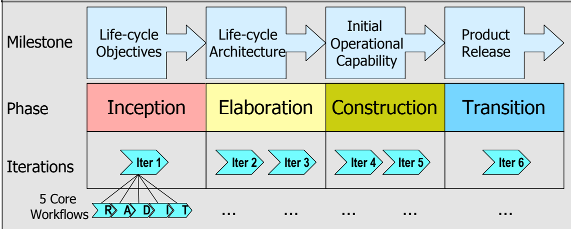
Increments

- Each iteration generates internal (or external) releases of various artefacts which together constitute a baseline
- A baseline is a set of reviewed and approved artefacts that:
  - Provides an agreed basis for further review and development
  - Can be changed only through a formal procedure such as configuration and change management
- An increment is the difference between the release of one iteration and the release of the next
  - The result of an iteration is an increment

USDP Lifecycle

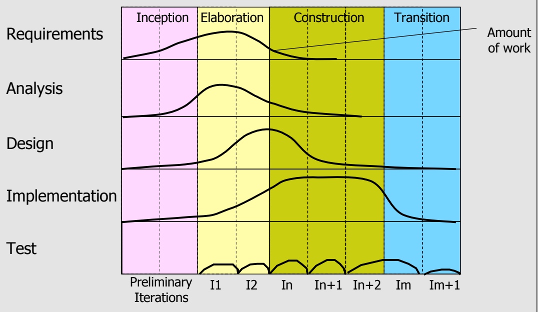
- The USDP lifecycle is divided into a sequence of phases
- Each phase may include many iterations
  - The exact number of iterations per phase depends on the size of the project!
  - One iteration per phase for small projects
- Each phase concludes with a major milestone

USDP Phases

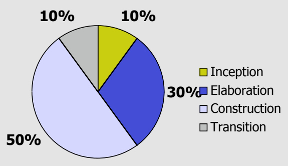


The exact number of iterations per Phase depends on the size of the project!
We have assumed a that this particular project lasts 18 months.

Phases and Workflows

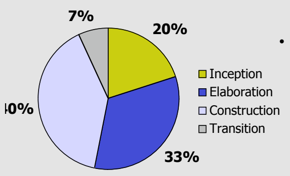


Time for a typical project



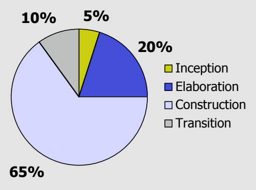
- If we consider a project of "typical" difficulty, then this is how the total time for the project is likely to be distributed over the phases

Time for a difficult project



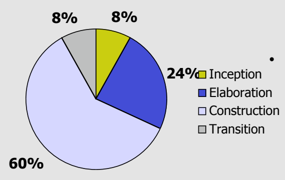
- If we consider a project of greater than normal difficulty, then this is how the total time for the project is likely to be distributed over the phases
- Note that for more difficult projects more time is spent in the early phases

Resource for a typical project

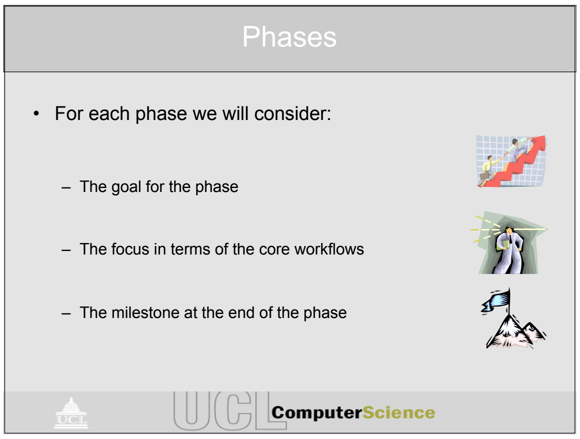


- If we consider a project of "typical" difficulty, then this is how the total resource for the project is likely to be utilised over the phases

Resource for a difficult project



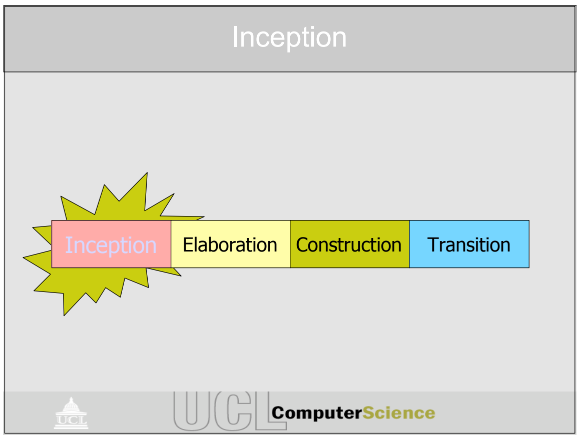
- If we consider a project of greater than normal difficulty, then this is how the total resource for the project is likely to be distributed over the phases
- Note that for more difficult projects more resource is used in the early phases



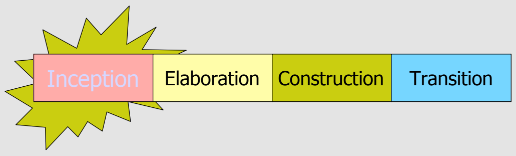
Phases

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

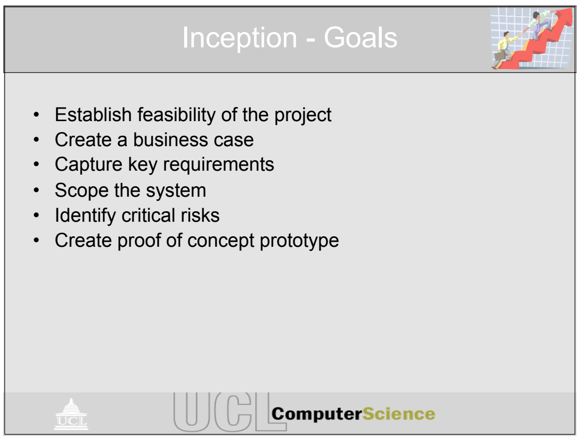
ComputerScience



Inception



ComputerScience

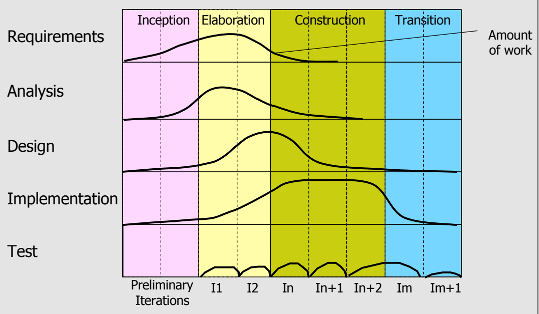


Inception - Goals

- Establish feasibility of the project
- Create a business case
- Capture key requirements
- Scope the system
- Identify critical risks
- Create proof of concept prototype

ComputerScience

Phases and Workflows



UCL ComputerScience

Inception - Focus

- Requirements – establish business case, scope and core requirements
- Analysis – establish feasibility
- Design – design proof of concept or technical prototypes
- Implementation – build the proof of concept prototype
- Test – not generally applicable

N.B. The blue bars indicate approximately the relative amount of resource needed

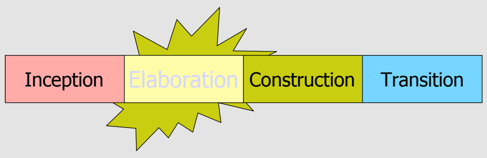
UCL ComputerScience

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

UCL ComputerScience

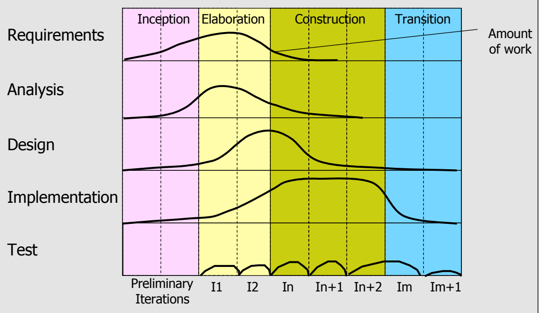
Elaboration

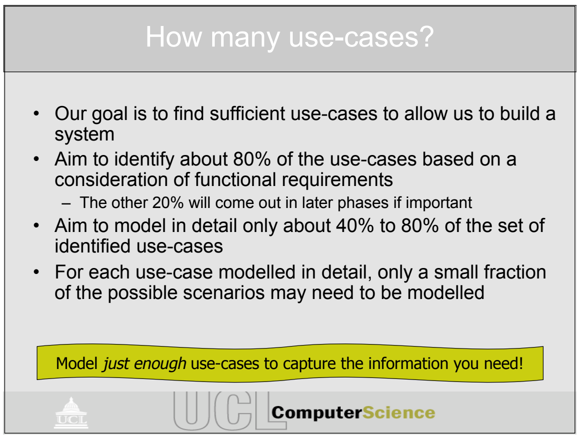


Elaboration - Goals

- Create an executable architectural baseline
- Refine Risk Assessment
- Define quality attributes (defect rates etc.)
- Capture use-cases to 80% of the functional requirements
- Create a detailed plan for the construction phase
- Formulate a bid which includes resources, time, equipment, staff and cost

Phases and Workflows

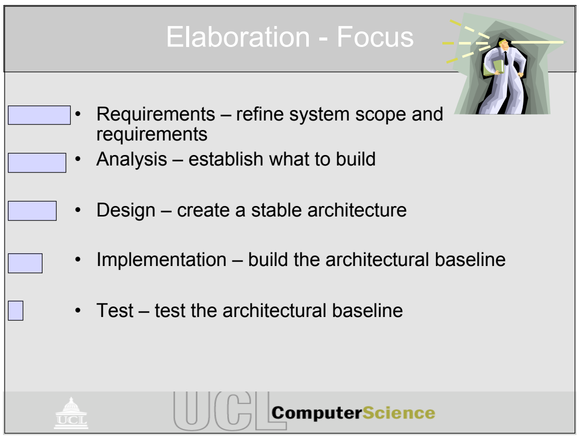




How many use-cases?

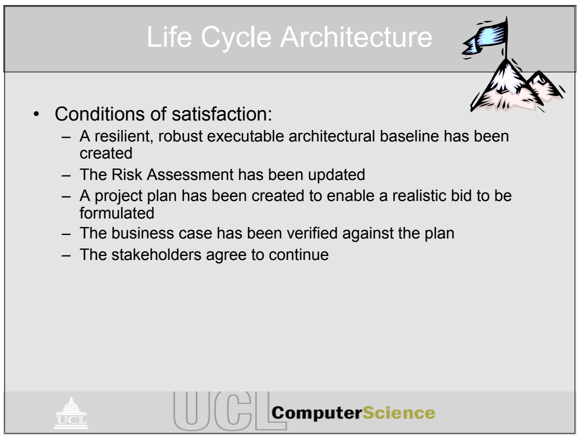
- Our goal is to find sufficient use-cases to allow us to build a system
- Aim to identify about 80% of the use-cases based on a consideration of functional requirements
  - The other 20% will come out in later phases if important
- Aim to model in detail only about 40% to 80% of the set of identified use-cases
- For each use-case modelled in detail, only a small fraction of the possible scenarios may need to be modelled

Model *just enough* use-cases to capture the information you need!



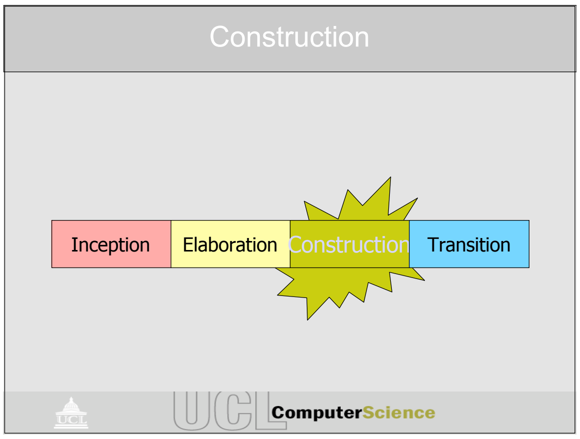
Elaboration - Focus

- Requirements – refine system scope and requirements
- Analysis – establish what to build
- Design – create a stable architecture
- Implementation – build the architectural baseline
- Test – test the architectural baseline



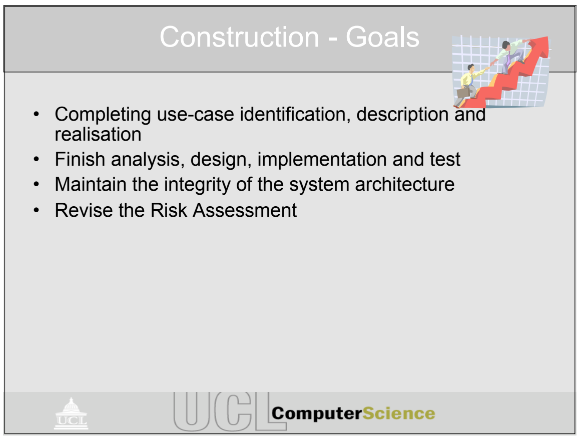
Life Cycle Architecture

- 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

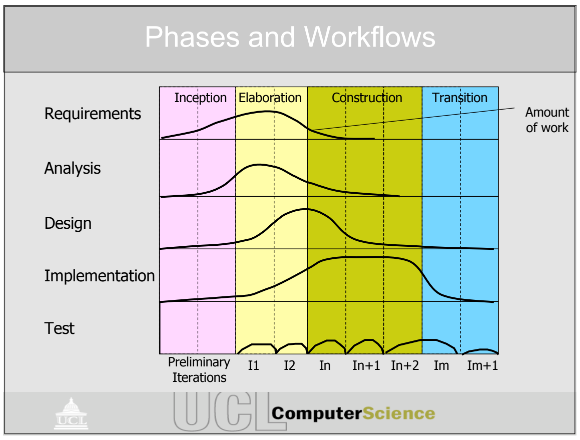
ComputerScience



Construction - Goals

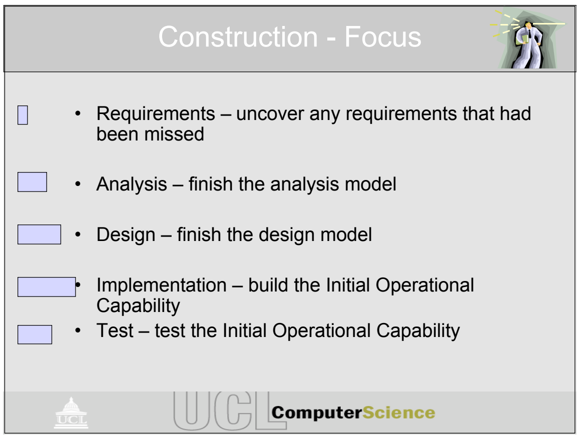
- Completing use-case identification, description and realisation
- Finish analysis, design, implementation and test
- Maintain the integrity of the system architecture
- Revise the Risk Assessment

ComputerScience



Phases and Workflows

ComputerScience

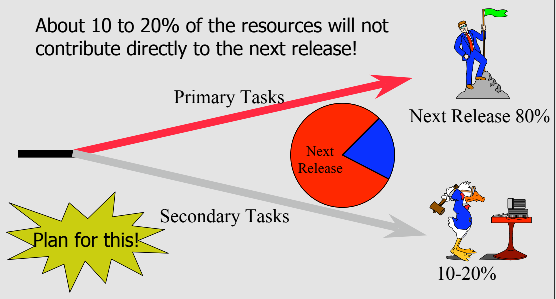


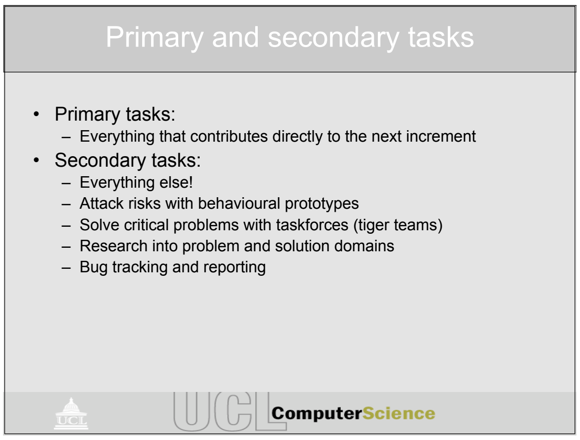
Construction - Focus

- 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

Plan for two lines of work…

About 10 to 20% of the resources will not contribute directly to the next release!





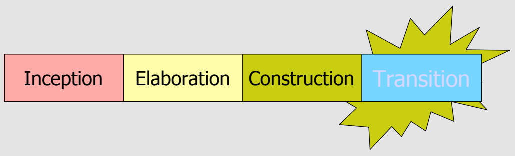
Primary and secondary tasks

- Primary tasks:
  - Everything that contributes directly to the next increment
- Secondary tasks:
  - Everything else!
  - Attack risks with behavioural prototypes
  - Solve critical problems with taskforces (tiger teams)
  - Research into problem and solution domains
  - Bug tracking and reporting

Initial Operational Capability

- Conditions of satisfaction:
  - The product is ready for beta testing in the user environment

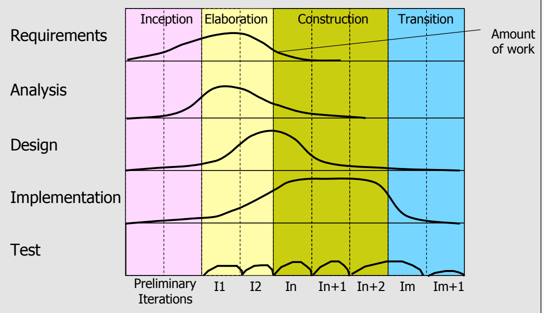
Transition



Transition - Goals

- Correct defects
- Prepare the users site for the new software
- Tailor the software to operate at the users site
- Modify software if unforeseen problems arise
- Create user manuals and other documentation
- Provide customer consultancy
- Conduct post project review

Phases and Workflows

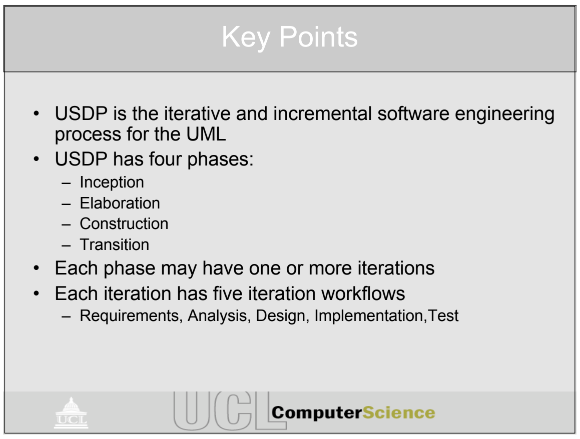


Transition - Focus

- Requirements – not applicable
- Analysis – not applicable
- Design – modify the design if problems emerge in beta testing
- Implementation – tailor the software for the users site and correct problems uncovered in beta testing
- Test – beta testing and acceptance testing at the users site

Product Release

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



Key Points

- USDP is the iterative and incremental software engineering process for the UML
- USDP has four phases:
  - Inception
  - Elaboration
  - Construction
  - Transition
- Each phase may have one or more iterations
- Each iteration has five iteration workflows
  - Requirements, Analysis, Design, Implementation,Test

UCL ComputerScience