

Methodologies

Objectives

- Give a definition for a methodology
- Describe three examples of methodology
- Categorise those that are AGILE methodologies
- Identify an appropriate methodology for a given project

Software Development Process

- Requirements gathering and analysis
- Design
- Implementation
- Testing

Implementation

- Realisation of a software element based upon the prior analysis and design
- The most popular element of the process
- Requires the least amount of time for the whole process

Decisions, Decisions

- What language should the implementation occur in ?
 - This is informed by the analysis stage, preference of the client and features of the language.
- What frameworks should be utilised ?
 - See above
- What standards should be utilised ?
 - See above

Processes, Models and guidelines

- As a result of the requirements analysis
 - We know what the system needs to do.
- As a result of the design stage....
 - We know how the system fits together and operates
- How do we know how to implement the software ?
 - Sometimes there is no plan
 - There are well-defined processes for approaching this task
 - Methodologies

Methodology

- What is a methodology ?
 - A plan for approaching the implementation task
- How does it benefit the developer ?
 - Provides a rigid process
 - Allows for a well-defined sequence
 - Avoids confusion
- Why do some developers avoid using them ?
 - Personal preference
 - Size and nature of a given project
 - Personal work

Categorisation

- There are literally hundreds of methodologies, how do we distinguish between applicable options ?
 - The nature of the project
 - The preference of the team
 - The preference of the client

Unified Process

software development process

who, what, when and how

Jacobson, Booch, Rumbaugh

Object-oriented Software Process (OOSP)

Unified Process

- use case driven
 - user requirements captured using Use Cases
 - sequences of actions performed by system
 - basis for design and implementation
- architecture centric
 - structure of system and sub-systems
 - component interaction and communication
 - hardware deployment

Types of Methodology

- iterative
 - mini-project
 - analysed, designed, implemented, tested
 - baseline generated
- incremental
 - difference between consecutive baselines
 - gradual development to delivered system
 - step-wise refinement

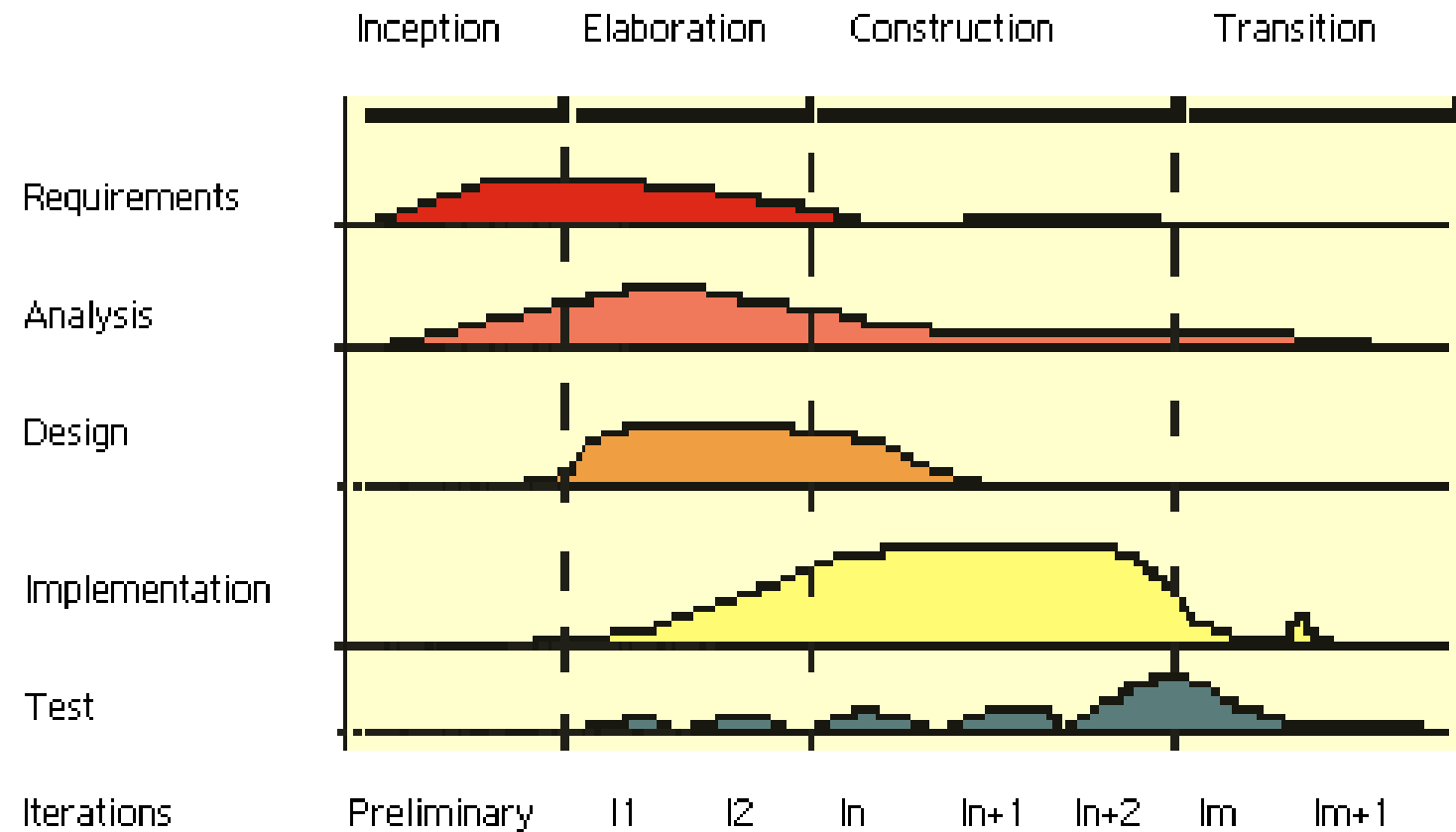
Unified Process

- iteration workflows
 - requirements, capture what system should do
 - analysis, refine requirements
 - design, realisation of requirements
 - implementation, producing software
 - test, verification of implementation

Unified Process

- UP lifecycle
 - inception
 - elaboration
 - construction
 - transition
- workflows each have associated lifecycle
- emphasis dependent on position of iteration within project

Unified Process



Unified Process usage

- Is this an agile methodology ?
 - No
- How do we Know ?
 - Process is incremental and iterative
 - Very rigidly defined

Can we be a bit more AGILE

- There are methodologies that more specifically fit the AGILE Manifesto, What are they ?
 - Dynamic Systems Development – DSDM
 - Rapid Application Development – RAD
 - Rational Unified Process – RUP
 - Extreme Programming - XP

Agile v UP

- Both iterative – Agile shorter iterations
- UP predictive, more structured
- Agile adaptive, more direct communication with customer
- Agile develops single features – but more refactoring as extras added
- Agile develops working, usable software more quickly
- Small versus large projects

Why so many methodologies ?

- Everyone thinks that their approach is better
- Everyone has a perspective
- There is no right or wrong way to approach the task
- Everyone wants a PhD

Rapid Application Development

- What is RAD ?
 - Rapid application development is an AGILE methodology
- What stages does it have ?
 - Requirements Analysis
 - User design
 - Construction / Implementation
 - Cutover

Requirements Analysis

- Mirrors the purpose of requirements gathering within UP
- What is the difference between UP requirements and RAD requirements ?
 - Less rigid in structure
 - Promotes discussion
 - Places the user in this discussion rather than getting final sign off

User Design

- Design stage, what it says on the tin
- The user will propose models, prototypes and interfaces
- This is a continuous process throughout for the realisation of specific features

Construction / Implementation

- Reflective of the SDLC counterpart
- Involves the implementation of software as identified within the analysis and design stages
- Users may still be involved throughout this process and propose changes as artefacts are being realised

Cutover

- The delivery Phase
- Software is supplied to the user, and employed through means of some changeover method
- User training is provided
- Delivered far sooner than the efforts of SDLC

Advantages

- Feature identification
- Feature realisation and testing
- Increased speed
- Reduced initial paperwork

Disadvantages

- Lack of control over the system
- Feature creep
- Dinosaurs
- Lack of appropriate or sufficient design

Tasks

- What is XP ?
- How does it differ from the unified process ?
- How does it differ from RAD ?
- What are the key components for the process to work effectively ?