DSDM Agile Project (DAP) Framework

Requirements modelling

Lecturer : Luis Aguirre

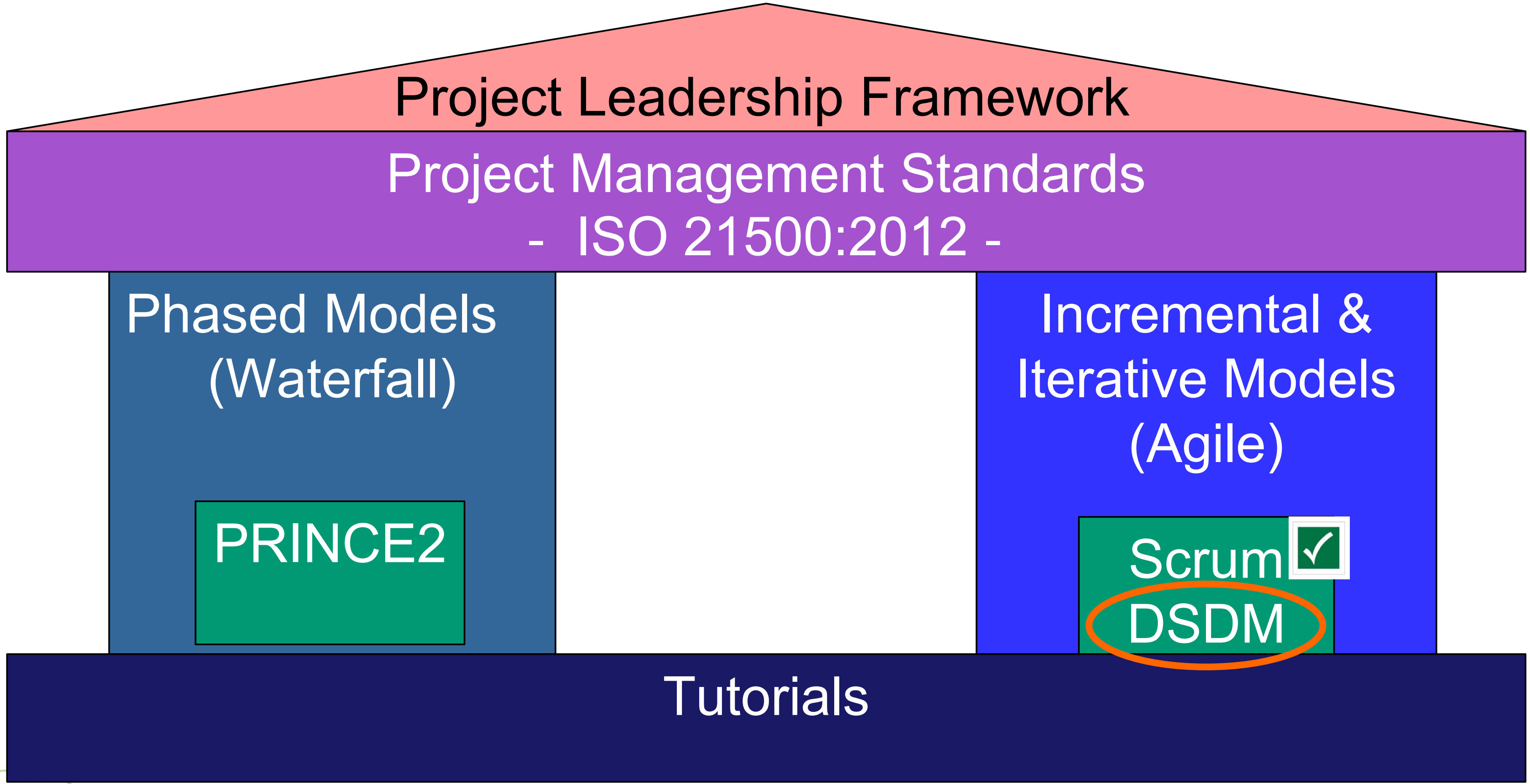
Topics for this week

1) **DSDM requirements modelling**

2) DSDM Structured Timeboxing

3) Control with Iterative Development

Unit Themes



Project Leadership Framework

Project Management Standards
- ISO 21500:2012 -

Phased Models (Waterfall)

PRINCE2

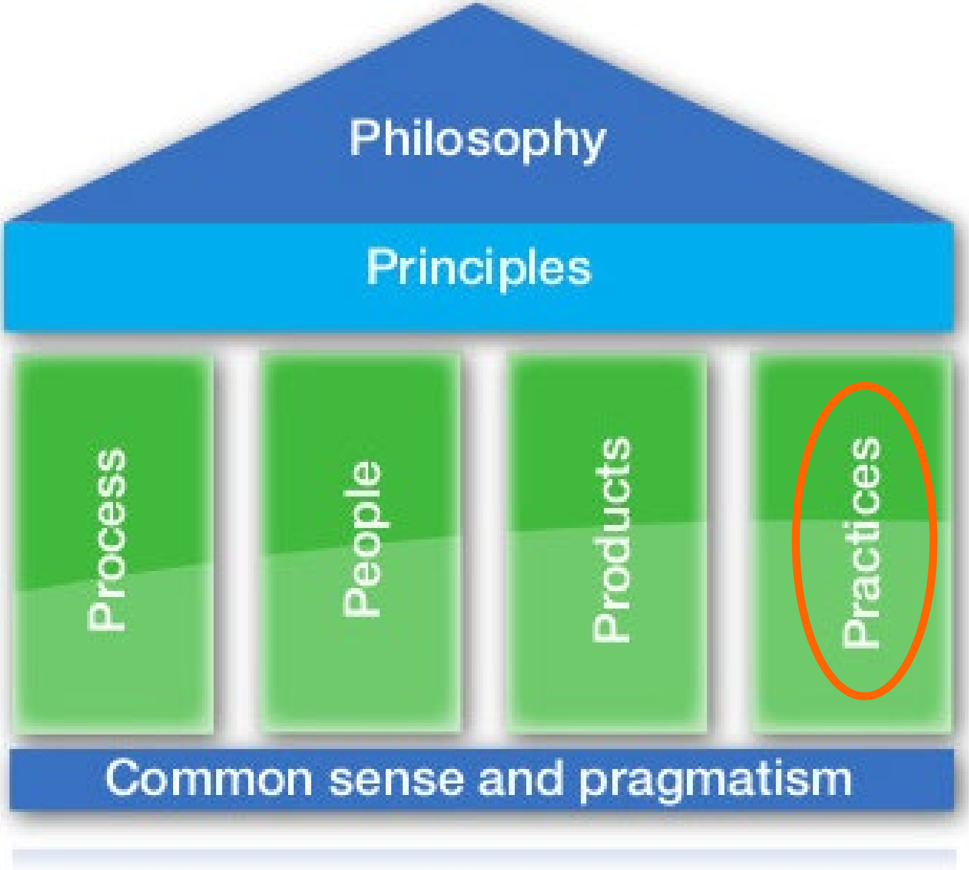
Incremental & Iterative Models (Agile)

Scrum ☑

DSDM

Tutorials

Key Practices - Control



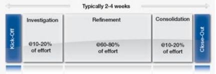
Key Practices

Iterative Development

Facilitated Workshops

WHY
WHERE
WHO
WHAT
HOW
WHEN

Modelling



Timeboxing

MoSCoW Prioritisation

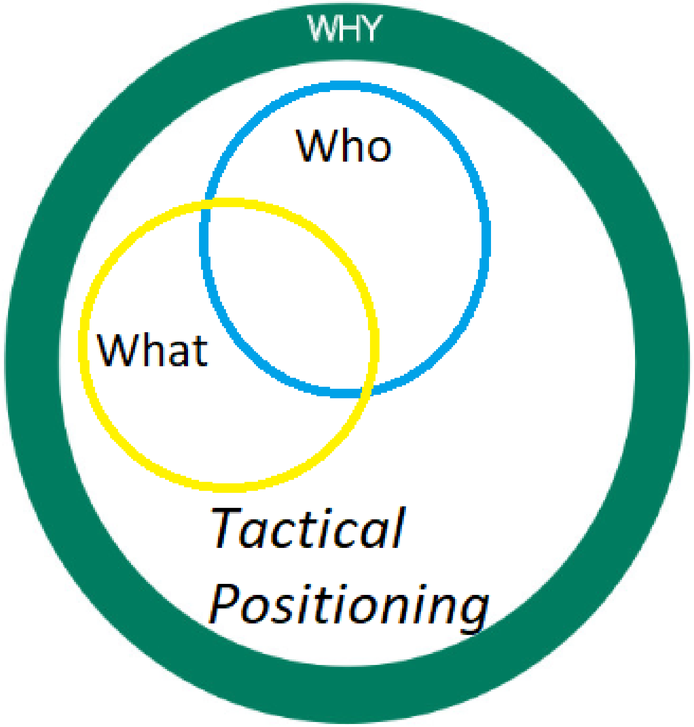
DSDM Modelling - Kipling’s Method 5W1H

WHY
Strategic Positioning

Kipling’s Poem

I have six honest serving men
They taught me all I knew
I call them **W**hat and **W**here and **W**hen
And **H**ow and **Why** and **W**ho

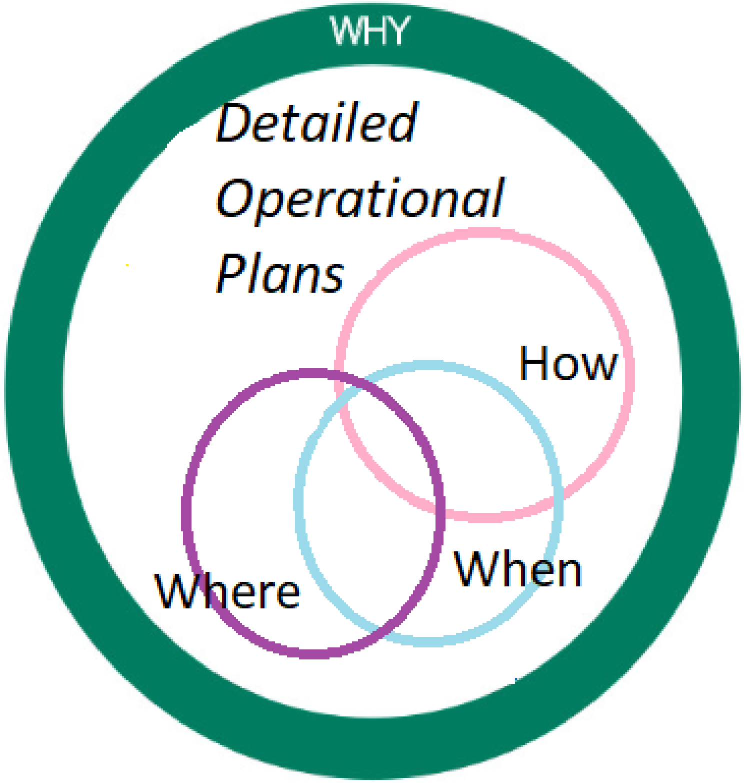
DSDM Modelling - Kipling's Method 5W1H



Kipling's Poem

I have six honest serving men
They taught me all I knew
I call them **What** and **W**here and **W**hen
And **H**ow and **W**hy and **Who**

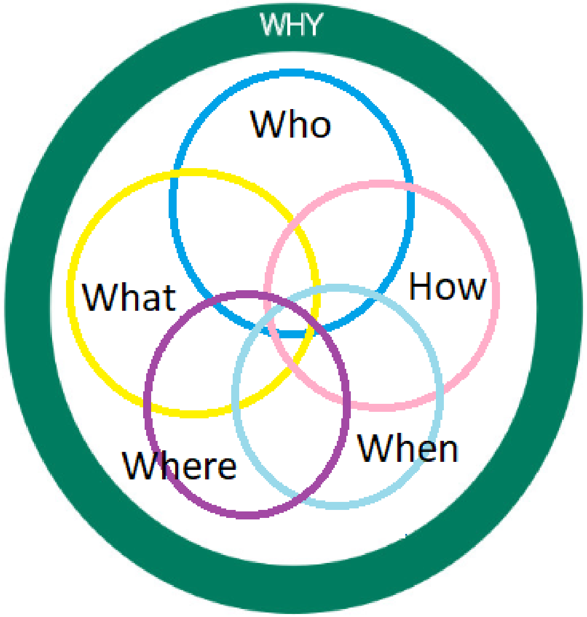
DSDM Modelling - Kipling's Method 5W1H



Kipling's Poem

I have six honest serving men
They taught me all I knew
I call them **W**hat and **W**here and **W**hen
And **H**ow and **W**hy and **W**ho

DSDM Modelling - Kipling’s Method 5W1H



Kipling’s Poem

I have six honest serving men
They taught me all I knew
I call them **W**hat and **W**here and **W**hen
And **H**ow and **W**hy and **W**ho

Achieved through:

- Visioning
- Modelling
- Prototyping

Modelling & Prototyping

Modelling techniques

- User stories
- Flow charts
- Swim lane (activity flow) diagrams
- Process Models
- Class Models
- Use cases & Use case diagrams
- Etc.

Prototyping techniques

- Story boards
- Wireframing
- PowerPoint with Live Links
- Web Tools (Wix.com etc.)
- Dedicated Tools e.g. Axure RP
- Etc.

Why Use Modelling?

Modelling techniques are designed to improve communications and prompt the right questions.

The purpose of modelling is to:

- Improve understanding through visual representations
- Support transparency by simplifying core elements of a requirement, usually in a picture
- Abstract the most relevant information for clarity
- Allow cross-checking for consistency

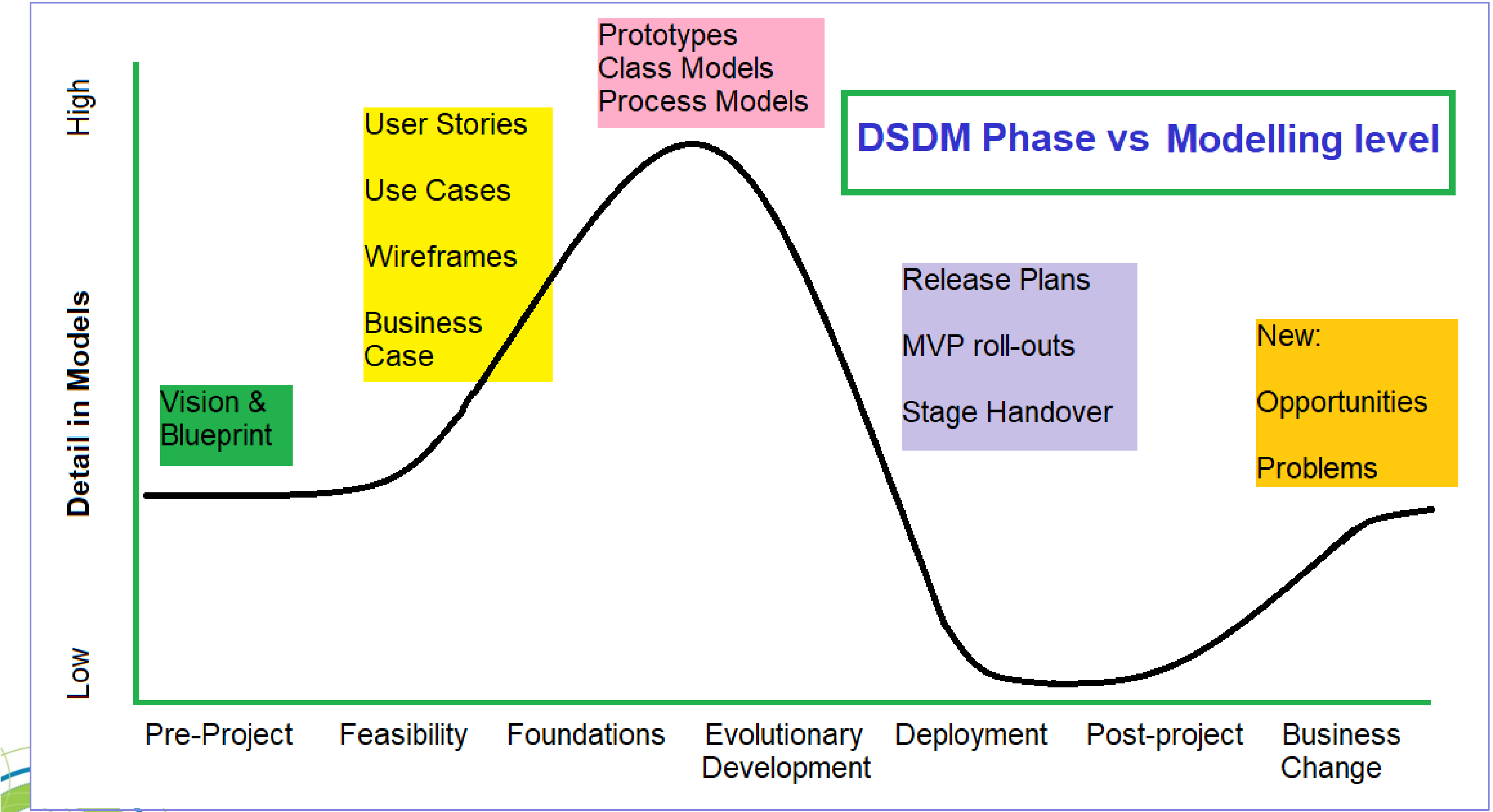
How to use Modelling?

Models can be defined as a:

- One of DSDM’s five key practices
- Description or analogy used to help visualise something that cannot be directly observed
- Small but exact copy of something
- Pattern or figure of something to be made

Modelling helps to make elements of the solution visible as early as possible.

When to use what Modelling?



When to use what Modelling?

The level of modelling at each phase must be appropriate to the level of complexity and characteristics of the project.

DSDM Phase	**Modelling level**
Pre-Project	Existing high-level models can be useful to illustrate how this project, or its solution, fits into a wider picture of change
Feasibility	Models support a simple ‘big picture’ view of what is being proposed and are used to explore possibilities and help communicate options
Foundations	More precise and elaborate models to help communicate plans and intentions to a variety of audiences
Evolutionary Development	High-level models continue to be evolved in terms of depth and detail, to help explore the detail of requirements, and ways that these can met as part of the Evolving Solution

When to use what Modelling?

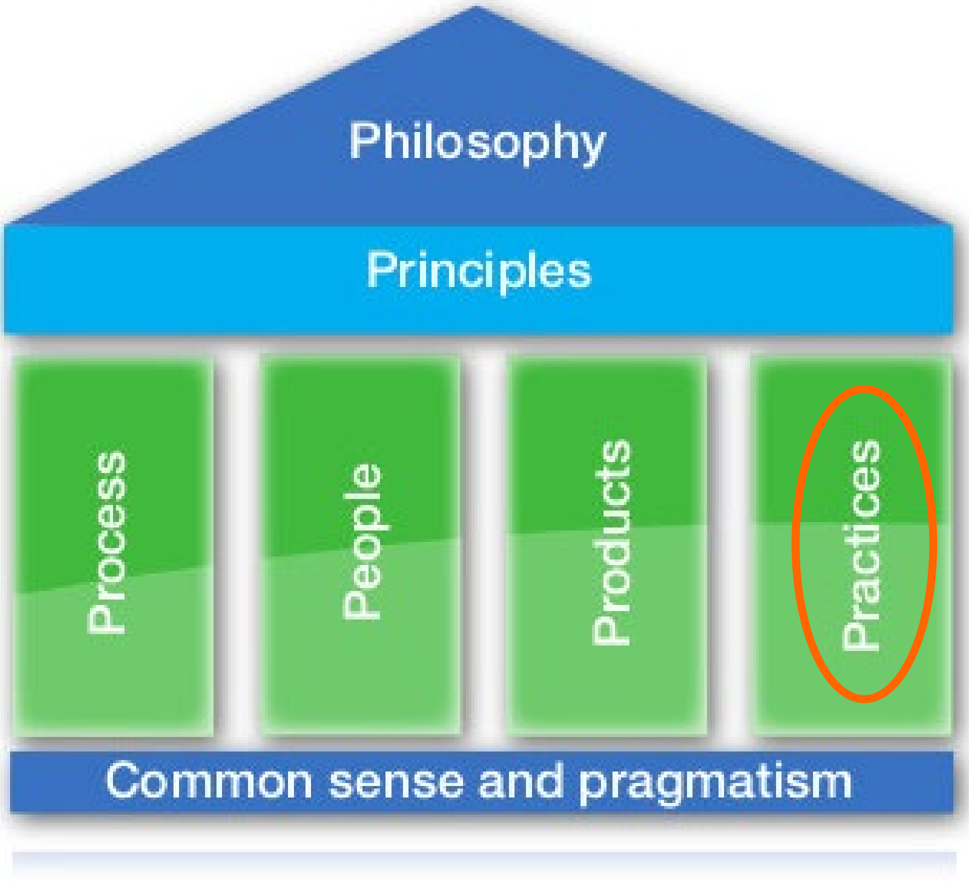
The level of modelling at each phase must be appropriate to the level of complexity and characteristics of the project.

DSDM Phase	Modelling level
Deployment	No new models created but models created to help operate and support the solution can be refined as it transitions into live use
Post-project	Refinement of models used to help to operate and support the solution, in line with any changes to the Deployed Solution over time
Progressive Business change	As the solution is deployed, the “as is” models of the current situation give way to the “to be” models that represent the new product or service

Topics for this week

1) DSDM requirements modelling
2) **DSDM Structured Timeboxing**
3) Control with Iterative Development

Key Practices - Control



Philosophy

Principles

Process

People

Products

Practices

Common sense and pragmatism

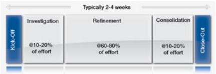
Key Practices

Iterative Development

Facilitated Workshops

WHY
WHERE
WHO
WHAT
HOW
WHEN

Modelling



Timeboxing

MoSCoW Prioritisation

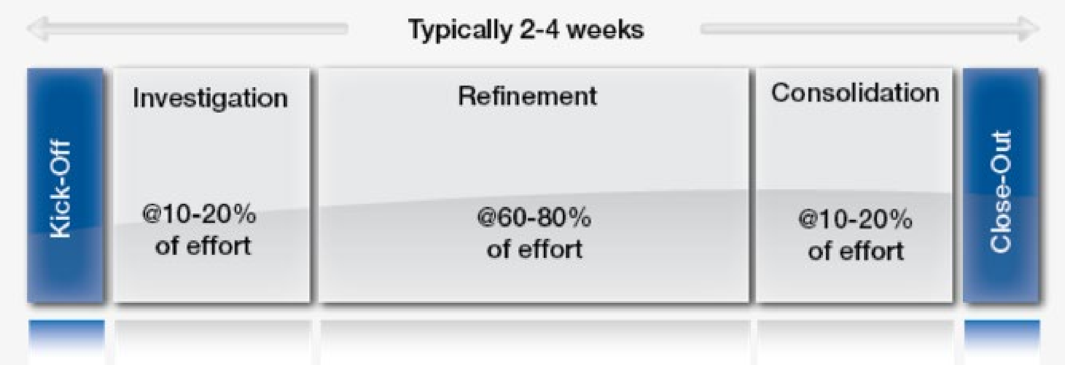
DSDM Structured Timeboxing

At any point during the DSDM structured Timebox, the whole Solution Development Team has visibility of progress and early warning if the overall Timebox objectives are at risk.

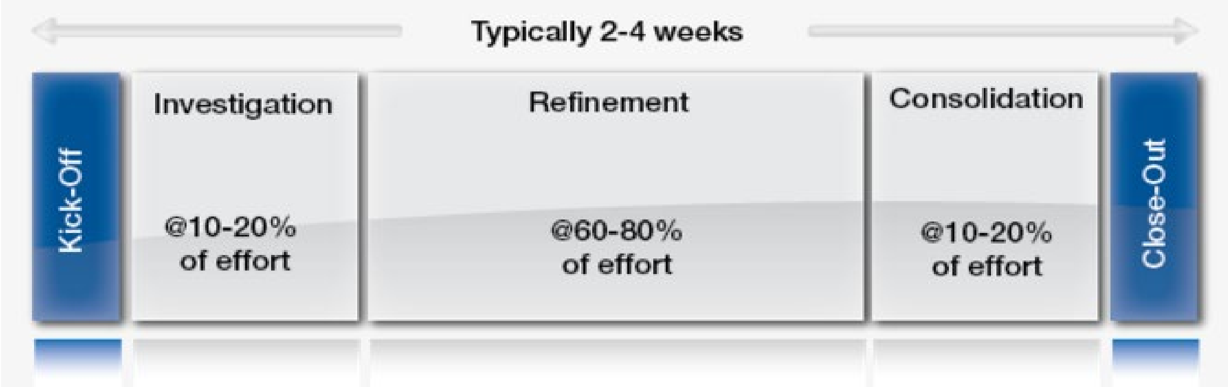
A DSDM structured Timebox comprises three main steps:

- Investigation
- Refinement
- Consolidation

Each of these steps ends with a review.



DSDM Structured Timeboxing – Iterations

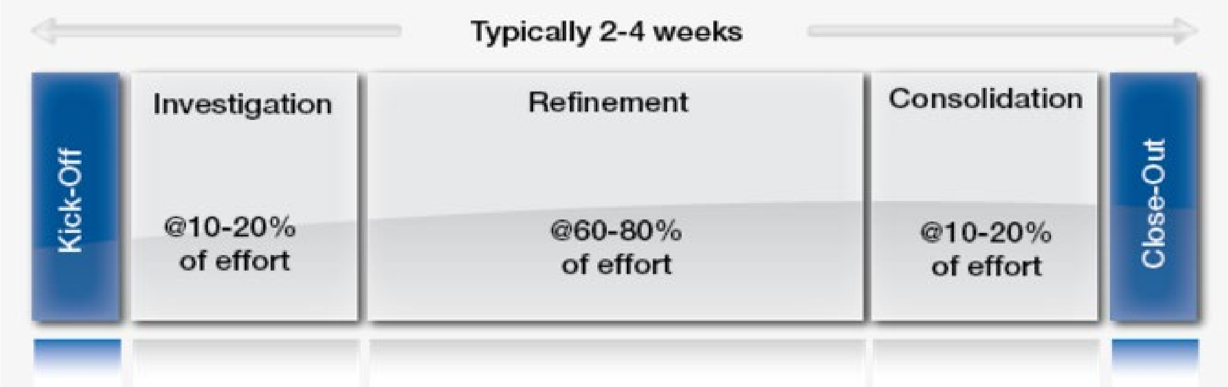


In each iteration of a Timebox:

- **Kick-Off:** understand the objectives and accept them
- **Investigate:** confirm the detail of all requirements and products to be delivered
- **Refinement:** develop in line with agreed priorities
- **Consolidation:** ensure all products meet their agreed acceptance criteria
- **Close-Out:** Business Visionary and Technical Coordinator formally accept deliverables

Each step ends with a review

DSDM Structured Timeboxing – Iteration Reviews



Investigation Review

- Team share results of their investigation with Ambassador, Visionary (possibly), and Technical Coordinator
- Team validate what they are intending to deliver by end of Timebox

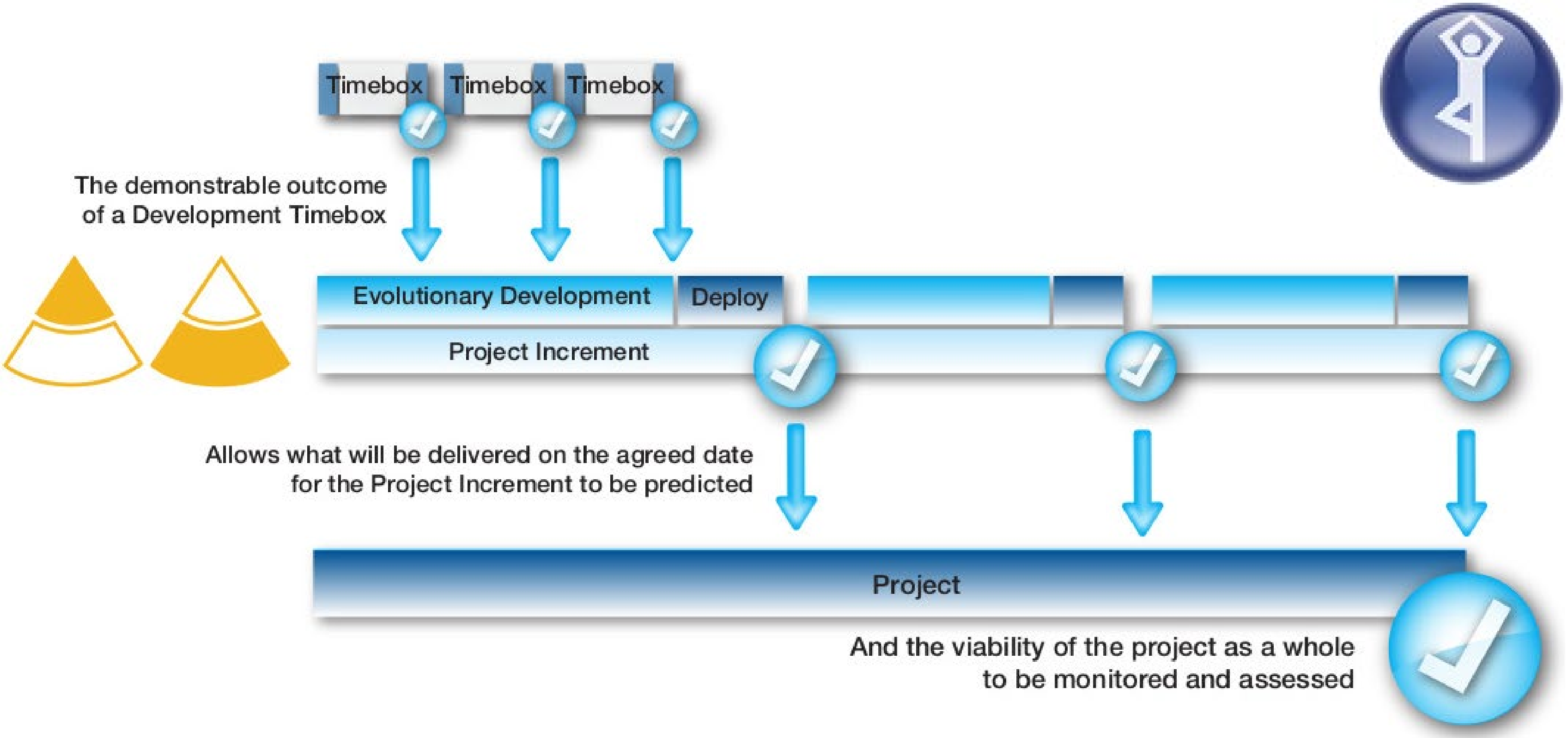
Refinement Review

- Team share results so far with Ambassador, Visionary (possibly), and Technical Coordinator
- Agree and prioritise work to be completed by end of Timebox

Consolidation Review

- Share final results of Timebox with Ambassador, Visionary (probably), and Technical Coordinator
- Confirm deliverables are fit for their intended purpose (meet agreed acceptance criteria)

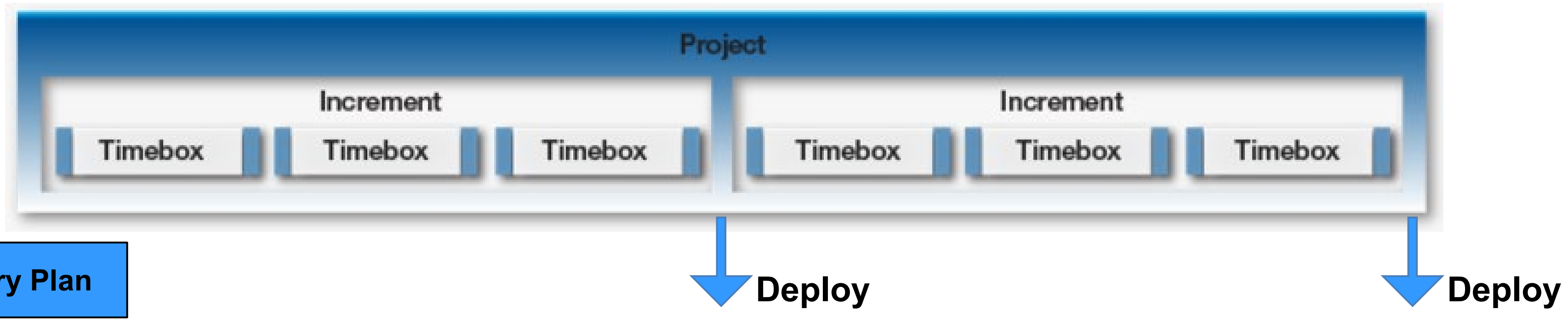
Timeboxing – Provides Control


Timebox
Timebox
Timebox
The demonstrable outcome of a Development Timebox
Evolutionary Development
Deploy
Project Increment
Allows what will be delivered on the agreed date for the Project Increment to be predicted
Project
And the viability of the project as a whole to be monitored and assessed

Topics for this week

1) DSDM requirements modelling
2) DSDM Structured Timeboxing
3) **<u>Control with Iterative Development</u>**

Timeboxing – Provides Control



Delivery Plan

Deploy

Deploy

Control is applied at the detail level

- Delivering on time every time
  - If this Timebox is on time, the Increment (and Project) are on time
- Delivering the right thing every time
  - MoSCoW Timebox contents, then use the priorities to drive the work
  - Define Acceptance Criteria and measure success against them
  - Team members are **all** responsible for doing a good job

Timeboxing – Provides Control

Control over the project variable 'time'

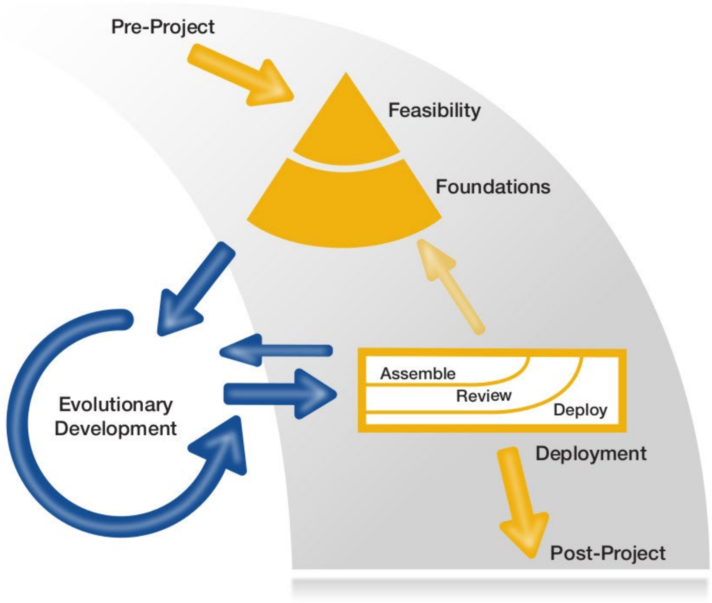
Objective - To understand the concept of Solution Development Timeboxes within an Increment and a Project.

The focus remains primarily on Solution Development Timeboxes.

- If the Team are in control at the Development Timebox level, then the Increment (Timebox) and the Project (Timebox) look healthy.
- If the current Development Timebox is failing, then the Increment and Project are seriously at risk, and action needs to be taken. The advantage of this is that it gives very early warning of problems, ensuring that the problem is not ignored

This overall picture is detailed in the Delivery Plan (created during Foundations)

Control with Iterative Development

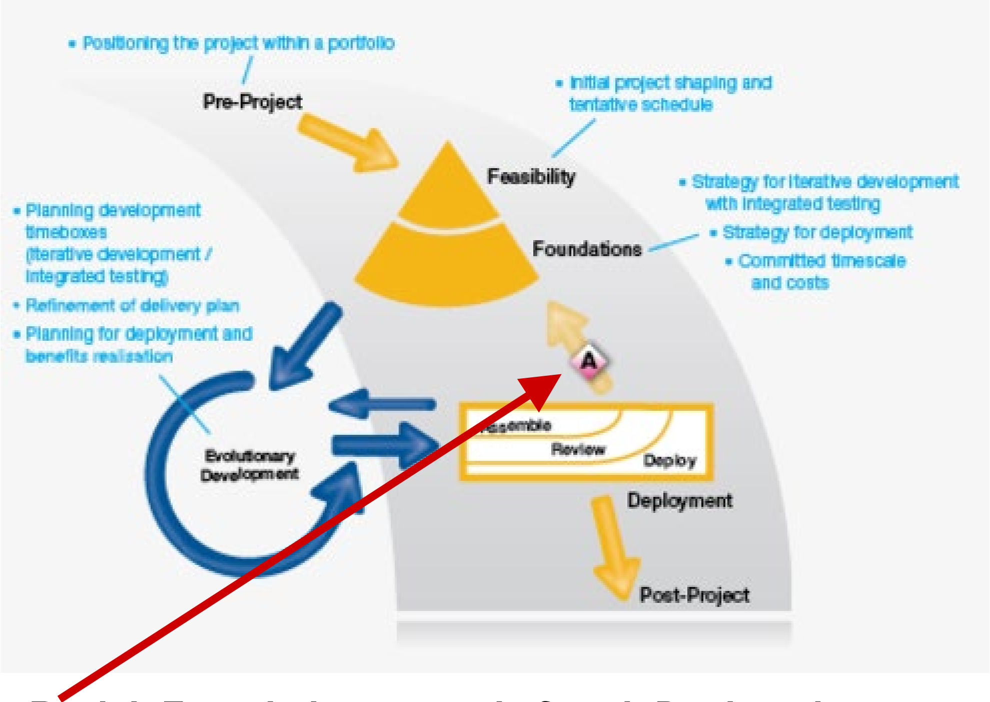


Prioritised Requirements List

Outline/Delivery > Timebox > Deployment Plan

Evolving > Deployed Solution

Requirements Through DSDM Lifecycle


• Positioning the project within a portfolio
Pre-Project
• Initial project shaping and tentative schedule
Feasibility
• Strategy for iterative development with integrated testing
• Strategy for deployment
• Committed timescale and costs
Foundations
• Planning development timeboxes (iterative development / integrated testing)
• Refinement of delivery plan
• Planning for deployment and benefits realisation
Evolutionary Development
A
Assemble
Review
Deploy
Deployment
Post-Project

Revisit Foundations at end of each Deployed Solution to check that project remains viable, and verify the validity and priority of requirements for the next Deployed Solution.

Feasibility:

- Outline Plan (approximate size and duration of project)
- Delivery Plan (high-level outline for first Increment)

Foundations:

- Schedule of Timeboxes for the first Deployed Solution with resources required
- How to develop and control the solution
- Strategy for deployment
- Baseline Delivery Plan

Evolutionary Development:

- Timebox planning (lowest level of planning)
- Timebox Plans based on task-level estimates and captured on a Team Board

- ➢ **Use PRL to manage Requirements**
- ➢ **Functional versus Non-Functional**
- ➢ **Requirements emerge: Depth versus Breadth**

Categories of Requirements

The success of any solution is based on:

- What it does (functionality, features)
- How well it performs against defined parameters (non-functional attributes, acceptance criteria, service levels)

Requirements are defined as:

- Functional Requirements (FRs)
- Non-Functional Requirements (NFRs)
  - Includes Usability Requirements

Functional Requirements (FRs)

FRs express function or feature and define what is required (for example):

- Visit customer site
- Obtain conference venue

FRs do **not** state how a solution will be physically achieved (for example):

- One possible solution: drive to customer site
  - Potential alternative solutions:
    - Fly to customer site
    - Travel by train
- One possible solution: build conference centre
  - Potential alternative solution: hire a hotel room

Stating requirements early in the project as **what** rather than **how** allows room for flexibility and innovation later.

Non-Functional Requirements (NFRs)

NFRs:

- Define how well, or to what level a solution needs to behave
- Describe solution attributes (usability, security, reliability, maintainability, availability, performance, response time, etc.)

NFRs can be:

- Solution-wide or impacting a group of functional requirements:
  - All customer facing functionality must carry the company logo
  - All customer-facing functionality must respond to requests within 2 seconds
- Related to a particular functional requirement, for example:
  - Hire conference venue might have NFRs of accessibility, security, and availability

Prioritised Requirements List (PRL)

- **Requirements driven by Business roles**
  - Visionary, Ambassador, Advisors

- **Feasibility phase**
  - High level requirements define main objectives of project
  - Small number of high level statements

- **Foundations phase**
  - High level requirements broken down into more detail
  - Good requirements define **what is needed**
    - Not **how** solution will be realised

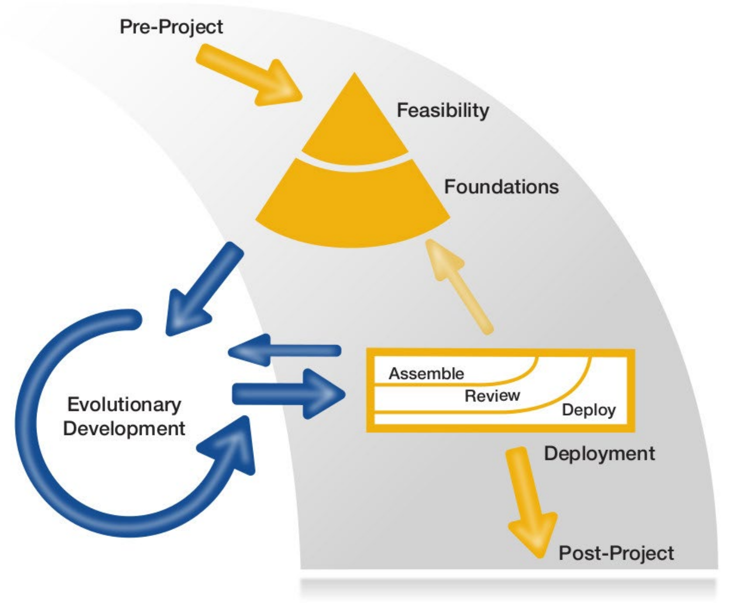
- **Solution Development Timeboxes (Evolutionary Development)**
  - Requirements investigated in full detail
  - Focus on a small subset of the PRL within the Timebox

Prioritised Requirements – Business Analysis

Business Analyst role is key

- Responsible for Prioritised Requirements List (*PRL*) aka Product Backlog
  - Ensuring clarity and completeness of requirements – by developing PRL
  - Ensuring business needs properly analysed and reflected in guidance for team to develop solution
- Facilitates communication between business and technical roles
  - But must **not** become an intermediary for the business roles
- Helps business think through full implications of their ideas
- Works closely with Business Visionary, Ambassador, and Tester
- Has skills and techniques to help identify:
  - Dependencies, overlaps, and conflicts between requirements
  - Effect of project level requirements on corporate objectives and direction

Control with Iterative Development



Prioritised Requirements List

Delivery > Timebox Plan

Evolving Solution

Delivery Plan – Foundations (1)

- **At end of Foundations requirements now better understood**
  - More detail, better sizing, and MoSCoW applied

- **Now possible to create a Delivery Plan, focused on 2 areas**
  - Scheduling Work
  - Defining the Approach

- **Scheduling work**
  - A plan for deployment of current Increment and agreed schedule of timeboxes leading up to that deployment
  - As each Increment is completed, new schedule of timeboxes is created for next Increment

Delivery Plan – Foundations (2)

- **Delivery Plan defines the approach**
  - Describes overall structure and approach to be adopted when working in Development Timeboxes
  - Confirms exact resources for the project
  - Outlines number and likely length of Development Timeboxes
  - Provides information on the probable focus for Timeboxes and strategy for developing solution
- **Delivery Plan also includes an outline of Project Increments**
  - Agreeing strategy for deployment and benefits realisation
- **Delivery Plan does *not* provide low-level plans for Timeboxes**
  - This is done by team at the start of each Timebox

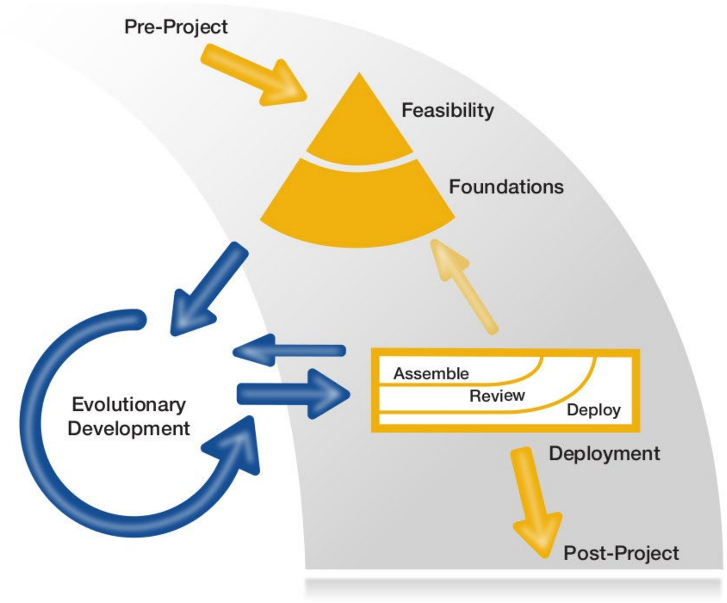
Timebox Plans – Evolutionary Development (1)

- **Lowest level of planning in DSDM projects**
- **For the next timebox, Solution Development Team**
  - Agrees and records:
    - What they will be working on in next few weeks
    - MoSCoW priorities
  - Predict:
    - What will be delivered (Timebox Must Haves)
    - What is highly likely to be achieved (Timebox Should Haves)
    - What may or may not get done in this Timebox (Timebox Could Haves)
- **Timebox Plan based on estimates at task level**
  - Less uncertainty, more accuracy

Timebox Plans – Evolutionary Development (2)

- **Not a detailed task plan**
- **Timebox Plan records**
  - Planned outputs
  - Acceptance criteria
  - MoSCoW priorities for timebox
  - Key milestones, such as dates for reviews
- **Timebox Plan provides information to PMs**
  - Enables PMs to communicate expectations to wider stakeholder group

Control with Iterative Development



Prioritised Requirements List

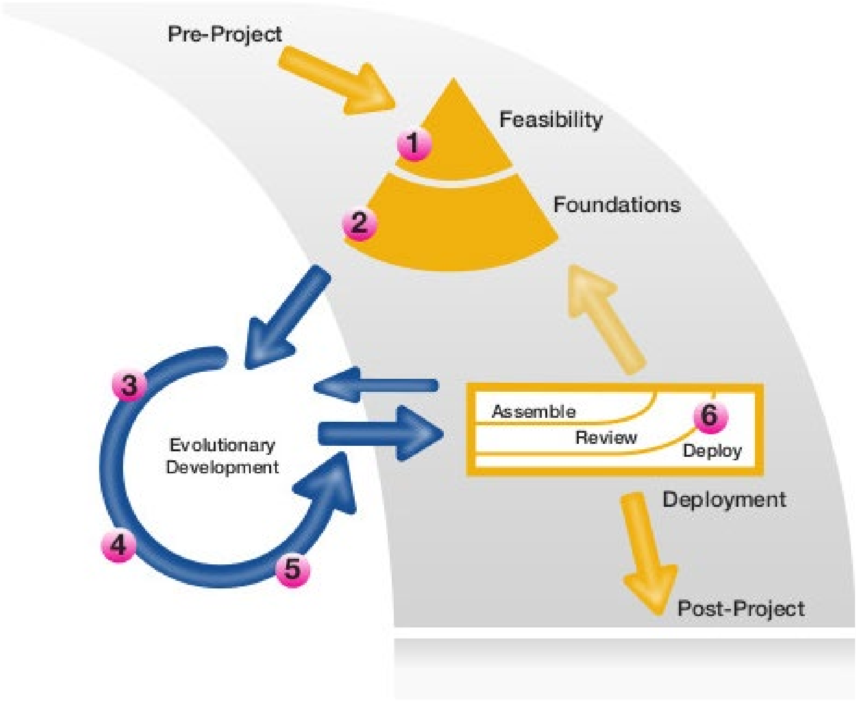
Outline/Delivery > Timebox > Deployment Plan

Evolving > Deployed Solution

DSDM Planning and Quality 1/2

Quality must be considered throughout the DSDM process

- Feasibility (1)
  - Consider risks at a high level
- Foundations (2)
  - Ask questions to assist in refining good high-level requirements
  - Establish three-way collaboration between Business roles, Solution Developers, and Solution Tester
  - Find the aspects of the solution that need the most attention
  - Define and agree required non-functional testing

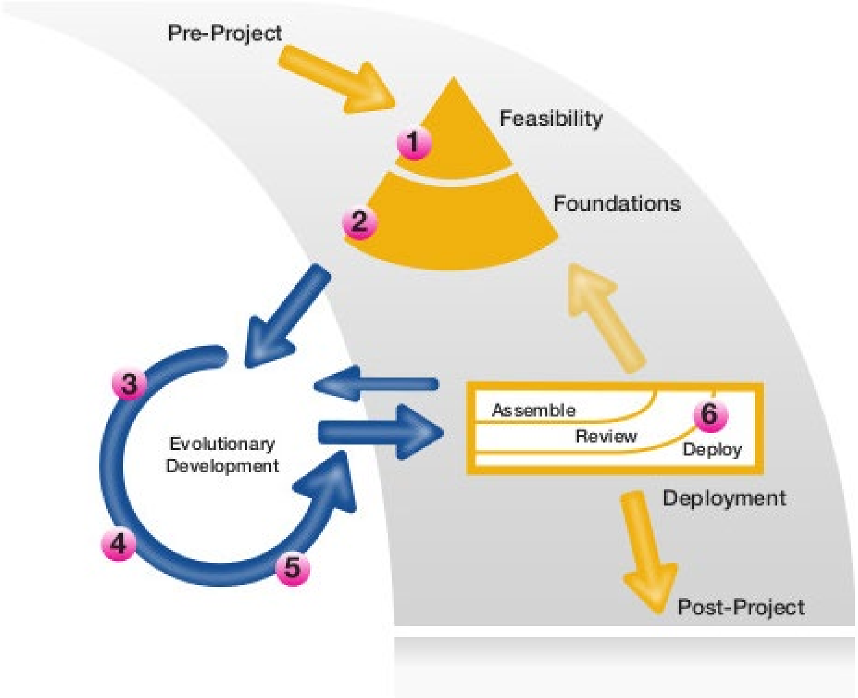


DSDM Planning and Quality 2/2

Quality must be considered throughout the DSDM process

- Evolutionary Development (3, 4, 5)
  - Prepare necessary quality assurance activities (3)
  - Define tests upfront (4)
  - Tests are part of the detailed requirement (4)
  - Identify highest value tests (4)
  - Assess quality and impact

- Deployment (6)
  - Test the full package to be delivered **and**
  - The process by which it will be delivered



Review (Take Aways) for this week

1) DSDM requirements modelling
   - ❖ Verify user stories
   - ❖ Clarify most relevant information

2) DSDM Structured Timeboxing
   - ❖ Investigate, Refine, Consolidate
   - ❖ Focus on Solution Development Timeboxes

3) Control with Iterative Development
   - ❖ Clear prioritised requirements
   - ❖ Communicate delivery plans and release expectations to complete stakeholder group
   - ❖ Assess quality and impact of highest value assets (testing)

Preparation for next week

Introduction to PRINCE2

Read: The 7 PRINCE2 processes.

(http://prince2.wiki/Processes)

Any questions?

Thank you.
See you next week.