



DSDM

Dynamic System Development Method



DSDM

- For selecting methods and tools in a given project environment
- Methodologies provide
 - guidance,
 - general principles and
 - strategies
- DSDM was created in 1994.



Methodologies provide guidance to These Key questions

- How much **involvement of the customer**
- How much **planning?**
- How much **reuse?**
- How much **modeling?**
- How much **process?**
- How much **control and monitoring?**



Introduction

- Projects were just
 - too big, and
 - too long,
 - communication was poor and
 - progress was measured in percentages, rather than deliverables.



Introduction

- When projects did deliver, they
 - often delivered late, and
 - often delivered the wrong thing,
 - due to
 - lack of on-going business involvement and
 - reliance on specifications which
 - tried and usually failed to capture and fix detailed requirements right at the start.



Introduction

- Some projects had tried a completely different approach: Rapid Application Development (RAD) with
 - users of the solution working closely with developers
 - to iteratively and incrementally build software applications,
 - not based on a formalised specification,
 - but on discussions, demonstrations and short feedback loops.



Introduction

- RAD addressed many of the problems of the traditional approach but,
- in doing so,
- it introduced a whole new set of problems,
 - particularly around the
 - supportability and
 - scalability of the solutions.



Introduction

- RAD provided quick fixes
- but often
 - its application adversely affected the quality of the solutions
 - because the disciplines of analysis and design were thrown out.



Introduction

- At that time, DSDM was launched to address the problems of the traditional approach
 - too slow,
 - too big,
 - not transparent enough,
 - not enough on-going business involvement
- as well as the problems introduced by RAD
 - focus only on
 - speed and
 - quick fixes,
 - no focus on quality,
 - no view of the big picture issues.



Introduction

- So DSDM brought together the best parts from a traditional approach
 - control and quality and
- from RAD
 - good communication, business involvement, transparency.



THE EIGHT PRINCIPLES

- Focus on the business needs
- Deliver on time
- Collaborate
- Never compromise quality
- Build incrementally from firm foundations
- Develop iteratively
- Communicate continuously and clearly
- Demonstrate control

- As a founder member of the Agile Alliance,
- DSDM has been at the heart of Agile since 2001.
- The philosophy and principles of DSDM helped shape the Manifesto for Agile Software Development,
 - although DSDM takes the concept of Agile far wider than just software.

- In DSDM,
- the iterative approach encourages detail to emerge over time;
- therefore,
- the current step needs to be completed in only enough detail to allow the project to move to the next step with any shortfall in detailed understanding being dealt with in a subsequent iteration of development.

- In DSDM,
 - the iterative approach encourages detail to emerge over time;
 - therefore,
- the current step needs to be completed in only enough detail to allow the project to move to the next step with any shortfall in detailed understanding being dealt with in a subsequent iteration of development.
- DSDM'de, yinelemeli(baştanalmalı, tekrarlı, iterative) yaklaşım, ayrıntıların zaman içinde ortaya çıkmasını teşvik eder;
- bu nedenle,
- mevcut adımlın ayrıntılı tanımlamasının,
 - detayları anlamada var olabilecek eksiklikler sonraki bir geliştirme yinelemesinde ele alınacağı için, -
- projenin bir sonraki adıma geçmesine yetecek düzeyde olması yeterlidir.

- There is also a very strong likelihood that the **business requirements will change** over time, and that such change is most likely to happen **at the detail level**.
 - This being the case, the effort spent on **detailed up-front work** is very **likely to be wasted**.
- In addition, solutions built using the DSDM approach address
 - the current and **imminent needs** of the business
 - **rather than**, for example,
 - the traditional approach of **attacking all the perceived possibilities**.

- This is preferable to
- trying to implement a more extensive solution that
 - has been complicated and
 - often compromised by failed attempts
 - to predict future business needs.



DSDM and Agile

- How Does DSDM Differ From Most Agile Approaches?
- "agreeing the foundations early in the project is essential".
- DSDM also addresses many of the general concerns about Agile development.
- Specifically,
- DSDM requires basic foundations for the project to be agreed at an early stage.
- This allows businesses to understand
 - the scope and
 - fundamental characteristics of the proposed solution,
 - and the way it will be created, before development starts.



DSDM and Agile

- Clarifying and agreeing the foundations for the project
 - from the **combined perspectives of business**,
- **reduces the likelihood of nasty surprises on DSDM projects.**
- In particular,
 - for larger corporate organisations
 - or organisations with a complex architecture
 - and/or governance standards,
- agreeing the foundations early in the project is **essential**.



DSDM and Agile

- DSDM also describes a broader set of roles than most Agile approaches
- giving it a better fit with most corporate environments without compromising Agility.



PROCESS

- In line with the DSDM philosophy that

“the best business value emerges when
projects are aligned to clear business goals,
deliver frequently and
involve the collaboration of motivated and
empowered people”.



PROCESS

- the DSDM approach to development and delivery is both
 - iterative and incremental,
 - with the most important business needs
 - typically being addressed early
 - while less important features are delivered later.



PROCESS

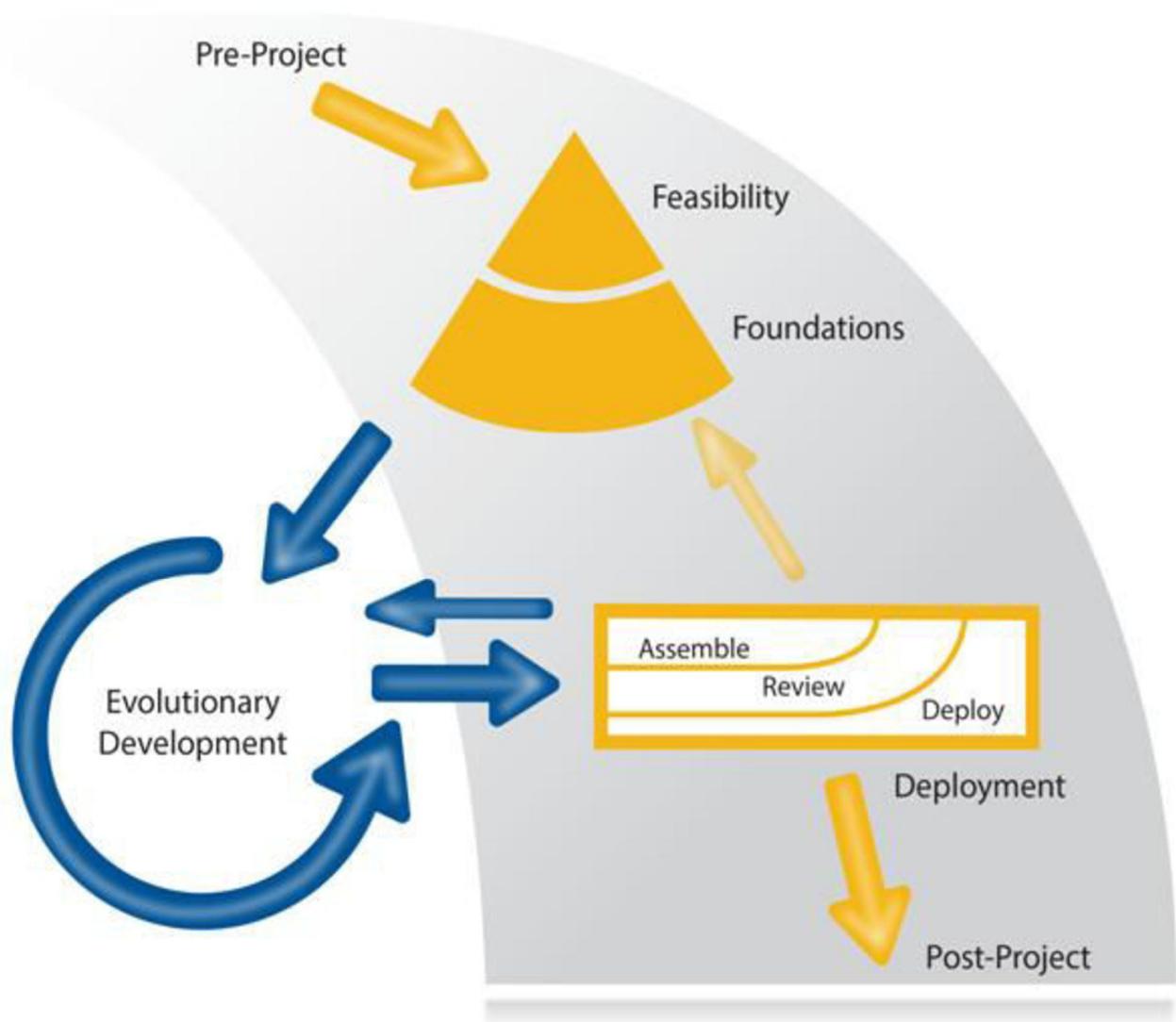
- The iterative nature of DSDM
- enables business representatives
 - to see the solution as it evolves,
 - to provide feedback on it and
 - to request changes
 - throughout the development of the solution.



PROCESS

- Unlike most Agile approaches, DSDM
 - integrates project management
 - and
 - product development
 - into a single process.

PROCESS





PROCESS

- The project process, as shown in the figure above, has four main phases:
 - Pre-Project,
 - Feasibility,
 - Foundations,
 - Evolutionary Development,
 - Deployment,
 - Post-Project.



Pre-Project Phase

- In line with the DSDM Philosophy that **“best business value emerges when projects are aligned to clear business goals”**
- the Pre-Project phase ensures that
 - **only the right projects are started, and**
 - **that they are set up correctly,**
- **based on a clearly defined objective.**



Feasibility Phase

- The Feasibility phase is intended primarily to establish
- whether the proposed project
 - is likely to be feasible from a **technical perspective** and
 - whether it appears cost-effective from a **business perspective**.
- The effort associated with Feasibility should be just enough
 - to decide **whether further investigation is justified**, or
 - whether the **project should be stopped now**, as it is unlikely to be viable.



Foundations Phase

- The Foundations phase takes the preliminary investigation from Feasibility to the next level.
- It is intended
 - to establish a **fundamental** (but not detailed) understanding of the business rationale for the project,
 - the potential solution that will be created by the project, and
 - how development and delivery of the solution will be managed.



Foundations Phase

- By intentionally avoiding low levels of detail,
- the Foundations phase should last no longer than a few weeks
 - even for large and complex projects.
- The detail
 - associated with requirements, and
 - how they should be met as part of the solution,
- is intentionally left until
 - the Evolutionary Development phase of the project.



Foundations Phase

- It may sometimes be necessary **to revisit Foundations after a Deployment phase.**
- The decision to revisit Foundations may be planned in from the start of the project;
 - for example, on a project where the business environment is sufficiently dynamic that the Foundations are expected to encounter significant change during the life of the project.
- Alternatively, the decision to revisit Foundations may be taken after a Deployment has produced an **unexpected outcome.**



Foundations Phase

- Returning to the Foundations phase
 - to re-affirm and,
 - where necessary,
 - to refine the foundations of the project
- normally takes significantly less time than
 - establishing them in the first place and
- may be as short as a single Workshop.



Foundations Phase

- For smaller, simpler projects,
- the Feasibility and Foundations phases can often be merged into a single phase.



Foundations Phase

- The aim of Foundations is
 - to understand the scope of work,
 - how it will be carried out,
 - by whom,
 - when and where.
- The Foundations phase also determines the project lifecycle
 - by agreeing how the DSDM process will be applied to the specific needs of this project.



Evolutionary Development Phase

- The Evolutionary Development phase requires the Solution Development Team(s)
 - to apply practices such as
 - Iterative Development,
 - timeboxing, and
 - MoSCoW prioritisation,
 - together with Modelling and Facilitated Workshops,
- to converge over time
- on an accurate solution that
 - meets the business need and
 - is also built in the right way from a technical viewpoint.



Evolutionary Development Phase

- **MoSCoW** is a prioritisation technique for helping to understand and manage priorities. The letters stand for:
 - **M**ust Have
 - **S**hould Have
 - **C**ould Have
 - **W**on't Have this time
- The use of MoSCoW works particularly well on projects.
- It also overcomes the problems associated with simpler prioritisation approaches which are based on relative priorities:



Evolutionary Development Phase

- Working within Timeboxes,
- the Solution Development Team create
 - Solution Increments,
 - iteratively exploring the low-level detail of the requirements and testing
 - continuously as they move forward.



Deployment Phase

- The objective of the Deployment phase is
- to bring a baseline of the Evolving Solution
- into operational use.
- The release that is deployed may be the final solution, or a subset of the final solution.
-

- The Deployment phase comprises three main activities:
 - Assemble,
 - Review and
 - Deploy.
- In addition, after the last release, the project is formally
 - Closed.



Assemble

- Assemble encompasses the work
- to “bring together” what is to be released.
- On a small simple project, the work involved during Assemble may be **minimal**.
- On larger more **complex projects** or programmes where multiple projects are feeding into a single release, the amount of work to assemble a number of Solution Increments into a single release could be **significant**.



Assemble

- Once all the elements of a release have been assembled, in most circumstances
- there will be some form of “**approval to deploy**”.
- This will be based on a final review of the solution before it goes into operational use
 - to ensure the proposed release **meets the appropriate standards** and
 - is **complete enough to be viable**.



Review

- At this point,
- the team also carries out a **retrospective for the Project Increment**,
- focusing on **ways of working** and **potential areas for improvement**.
- Information from both
 - the retrospective and
 - the formal review of the product
- **help shape plans for future increments**
- and can be used to **facilitate learning across projects within a portfolio**.



Deploy

- Once approval has been given,
- Deploy is the **physical act of putting** what has been assembled (the release) into operational use.
- It includes any **technical work**, such as transfer of the solution into the live (production) environment, but also the **enactment of any plans** for business change.



Close

- After the final Deployment, the project is formally closed.
- At this point, the whole team hold a **retrospective to review the overall project performance,**
 - both from the technical and/or process perspective and
 - from the business perspective.



Deployment and complexity

- This means that the Deployment phase may be
 - a simple or a complex activity.
- How deployment is done varies
 - from organisation to organisation, and
 - from project to project.
- For many organisations,
 - decisions about how deployment is handled are imposed by the organisation itself, and
 - are not negotiable by an individual project.



Post-Project Phase

- After the final Deployment for a project, the Post-Project phase checks
 - how well the expected business benefits have been met.
 - Although it may be possible to highlight immediate benefits,
 - most benefits will accrue over a pre-defined period of live use of the solution.



Post-Project Phase

- The Post-Project phase produces one or more **Benefits Assessments** for these realised benefits in relation to the business case.
- Benefits may be assessed
 - for individual releases (in which case the assessment of benefit should start before the Post-Project phase is reached),
 - for the whole project or may be omitted completely,
- depending on the needs of the organisation.