COURSE NAME

SOFTWARE ENGINEERING

CSC 3114

(UNDERGRADUATE)

CHAPTER 6

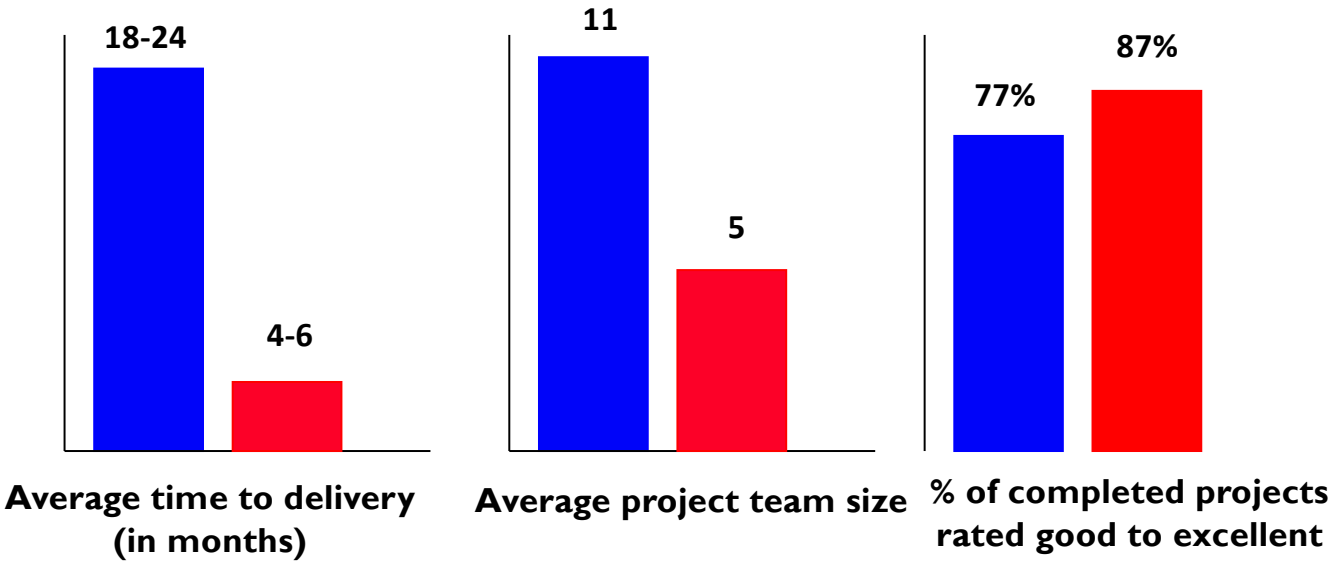
THE DYNAMIC SYSTEMS DEVELOPMENT METHOD (DSDM)

DSDM

- **The Dynamic Systems Development Method (DSDM)** is a public domain Rapid Application Development method which has been developed through capturing the experience of a large group of vendor and user organisations. It is now considered to be the UK's de-facto standard for RAD.

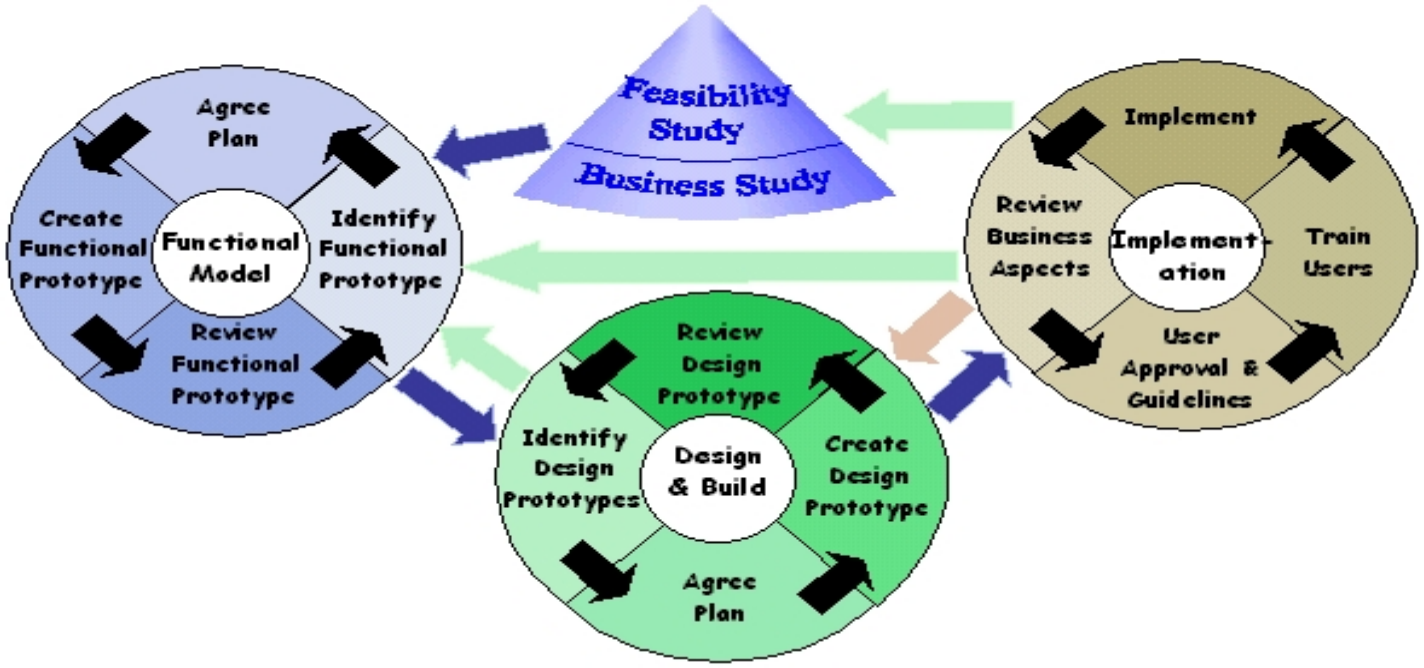
- The key to DSDM is to deliver **what** business needs **when** it needs
  - Achieved by using the various techniques in the framework and flexing requirements
  - The aim is always to address the current and imminent needs of the business rather than to attack all the perceived possibilities

TRADITIONAL METHOD VS. DSDM



Source: British Airways IM Department, Newcastle

DSDM PROCESS VIEW



DSDM PROCESS

Activity	Sub activity	Description
Study	**Feasibility Study**	Stage where the suitability of DSDM is assessed. Judging by the type of project, organizational and people issues, the decision is made, whether to use DSDM or not. Therefore it will generate a FEASIBILITY REPORT, a FEASIBILITY PROTOTYPE, and a GLOBAL OUTLINE PLAN which includes a DEVELOPMENT PLAN and a RISK LOG.
	Business Study	Stage where the essential characteristics of business and technology are analyzed. Approach to organize workshops, where a sufficient number of the customer's experts are gathered to be able to consider all relevant facts of the system, and to be able to agree on development priorities. In this stage, a PRIORITIZED REQUIREMENTS LIST, a BUSINESS AREA DEFINITION, a SYSTEM ARCHITECTURE DEFINITION, and an OUTLINE PROTOTYPING PLAN are developed.

DSDM PROCESS

Activity	Sub activity	Description
Functional Model Iteration	Identify functional prototype	Determine the functionalities to be implemented in the prototype that results from this iteration. In this sub-stage, a FUNCTIONAL MODEL is developed according to the deliverables result of business study stage.
	Agree schedule	Agree on how and when to develop these functionalities.
	Create functional prototype	Develop the FUNCTIONAL PROTOTYPE, according to the agreed schedule and FUNCTIONAL MODEL.
	Review functional prototype	Check correctness of the developed prototype. This can be done via testing by end-user and/or reviewing documentation. The deliverable is a FUNCTIONAL PROTOTYPING REVIEW DOCUMENT.

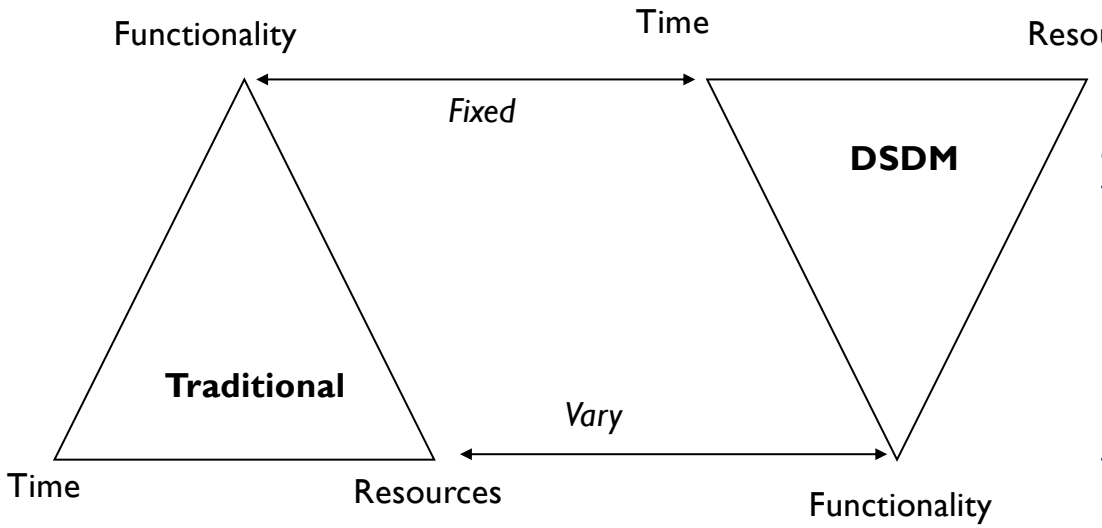
DSDM PROCESS

Activity	Sub activity	Description
Design and Build Iteration	Identify design prototype	Identify functional and **non-functiona**l requirements that need to be in the tested system. And based on these identifications, an IMPLEMENTATION STRATEGY is involved. If there is a TEST RECORD from the previous iteration, then it will be also used to determine the IMPLEMENTATION STRATEGY.
	Agree schedule	Agree on how and when to realize these requirements.
	Create design prototype	Create a system (DESIGN PROTOTYPE) that can safely be handed to end-users for daily use, also for testing purposes.
	Review design prototype	Check the correctness of the designed system. Again testing and reviewing are the main techniques used. An USER DOCUMENTATION and a TEST RECORD will be developed.

DSDM PROCESS

Activity	Sub activity	Description
Implementation	User approval and guidelines	End users approve the tested system (APPROVAL) for implementation and guidelines with respect to the implementation and use of the system are created.
	Train users	Train future end user in the use of the system. TRAINED USER POPULATION is the deliverable of this sub-stage.
	Implement	Implement the tested system at the location of the end users, called as DELIVERED SYSTEM.
	Review business	Review the impact of the implemented system on the business, a central issue will be whether the system meets the goals set at the beginning of the project. Depending on this the project goes to the next stage, the post-project or loops back to one of the preceding stages for further development. This review is will be documented in a PROJECT REVIEW DOCUMENT.

DIFFERENCE BETWEEN TRADITIONAL DEVELOPMENT VS. DSDM



Traditional Method

- Functional/requirements are fixed
- Time & resources can varies

DSDM/Agile Methods

- Functional/requirement varies
- Time & resources are fixed

TECHNIQUES TO CONSIDER IN DSDM

- Flexibility
- Timeboxing
- MoSCoW Rules
- Prototyping
- Facilitated Workshops

DSDM TECHNIQUES: FLEXIBILITY

- A fundamental assumption of DSDM is that nothing is built perfectly first time
- 80:20 Rule: assumes that a usable and useful 80% of the proposed system can be produced in 20% of the time it would take to produce the total system.
- In "traditional" development practice, a lot of time is spent in getting from the 80% solution to the total solution, with the assumption that no step ever needs to be revisited. The result is either projects that are delivered late and over budget or projects that fail to meet the business needs since time is not spent reworking the requirements.
- DSDM assumes that all previous steps can be revisited as part of its iterative approach. Therefore, the current step need be completed only enough to move to the next step, since it can be finished in a later iteration.

DSDM TECHNIQUES: TIMEBOXING

- Without effective timeboxing, prototyping teams can lose their focus and run out of control.
- Timeboxing works by concentrating on when a business objective will be met as opposed to the tasks which contribute to its delivery.
- **Timeboxing Basics**
  - Time between start and end of an activity
  - DSDM uses nested timeboxes, giving a series of fixed deadlines
  - Ideally 2 - 4 weeks in length
  - Objective is to have easiest 80% produced in each timebox
  - Remaining 20% potentially carried forward subsequent timeboxes
  - Focus on the essentials
  - Helps in estimating and providing resources

DSDM TECHNIQUES: MOSCOW RULES

- **MoSCoW** rules formalised in DSDM version 3

Must have – fundamental to project success

Should have – important but project does not rely on

Could have – left out without impacting on project

Want to have but Won't have this time for those valuable requirements that can wait till later development takes place; in other words, the Waiting List.

DSDM TECHNIQUES: PROTOTYPING

Prototypes are necessary in DSDM because

- Facilitated workshops define the high-level requirements and strategy
- Prototypes provide the mechanism through which users can ensure that the detail of the requirements is correct
- Demonstration of a prototype broadens the users‘ awareness of the possibilities and assists them in giving feedback to the developers
- Speeds up the development process and increases confidence that the right solution will be delivered

DSDM TECHNIQUES: FACILITATED WORKSHOPS

- Purpose is to produce clear outcomes that have been reached by consensus

- **Participants**
  - Workshop sponsor
  - Participants (development team)
  - Scribes (record)
  - Observers
  - Prototypers
  - Facilitator (help a group of people understand their common objectives and assists them to plan how to achieve these objectives)

- **Advantages of Workshops**
  - Speed
  - Involvement /ownership
  - Productivity
  - Consensus
  - Quality of decisions
  - Overall perspective / synergy (cooperation)

REFERENCES

- R.S. Pressman & Associates, Inc. (2010). *Software Engineering: A Practitioner's Approach.*
- Kelly, J. C., Sherif, J. S., & Hops, J. (1992). An analysis of defect densities found during software inspections. *Journal of Systems and Software*, *17*(2), 111-117.
- Bhandari, I., Halliday, M. J., Chaar, J., Chillarege, R., Jones, K., Atkinson, J. S., & Yonezawa, M. (1994). In-process improvement through defect data interpretation. *IBM Systems Journal*, *33*(1), 182-214.