



SSW-555: Agile Methods for Software Development

Dynamic Systems Development Method (DSDM)

Dr. Richard Ens
Software Engineering
School of Systems and Enterprises

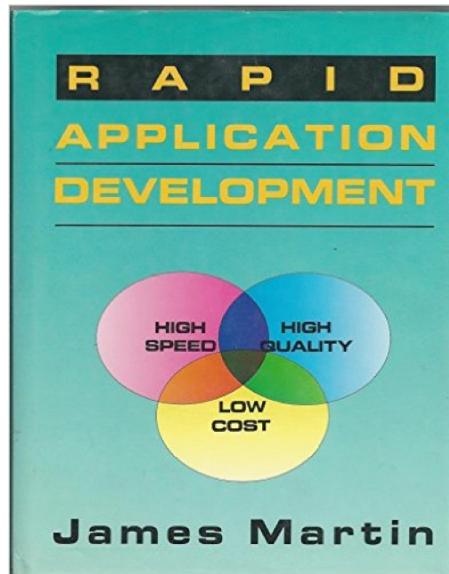


Acknowledgements

This lecture includes material from:

<http://www.codeproject.com/Articles/5097/What-Is-DSDM>

Rapid Application Development, James Martin, ISBN-13: 978-0023767753

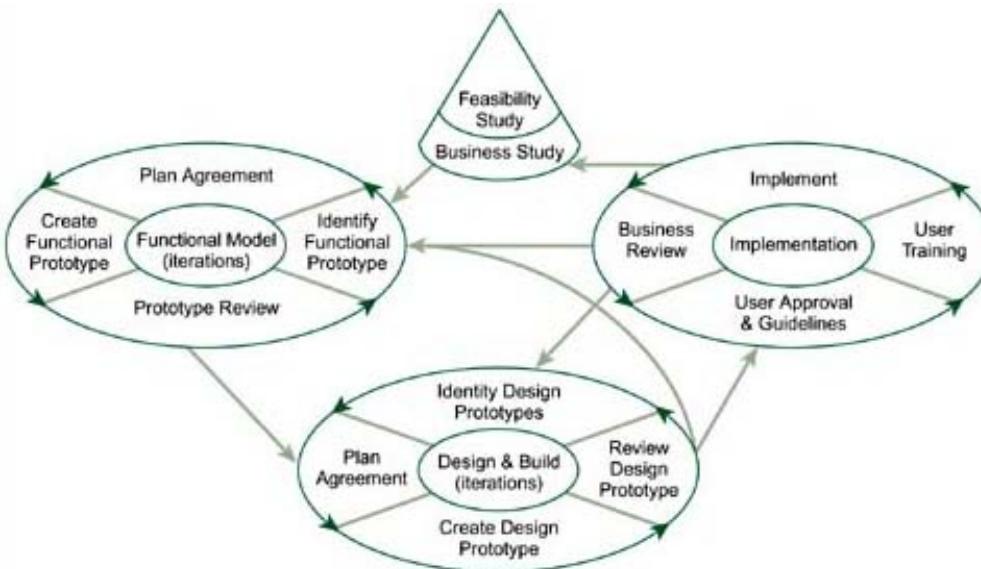


Today's topics

Rapid Application Development (RAD)

Dynamic Systems Development Method (DSDM)

Core techniques of RAD and DSDM



Rapid: the precursor to Agile (1970s)

The traditional software development model was based on a Customer-Supplier relationship

Used a traditional engineering model

Create design, construct solution, and deliver to customer

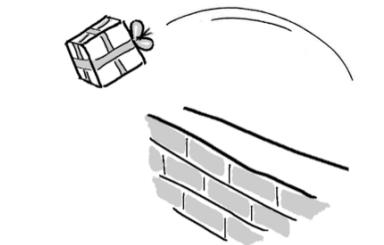
Fixed contract between customer and supplier

E.g. big government agency and software supplier

Fixed schedule, e.g. 1-2 years

Requirements negotiated up front

Customers and developers were independent and frequently didn't trust each other



Rapid: the precursor to Agile (1980s)

Companies created in-house IT departments



e.g. create new reports from an existing database or
automate manual processes

Customers and suppliers work for the same company

In-house development could use a different model

- Schedule could trade features for speed
- Requirements could be changed during development
- No need to renegotiate contract terms and conditions
- Customers and developers could work together



Rapid Application Development (RAD)

Originally developed in the 1970s by Dan Gielan at New York Telephone Co's Systems Development Center

Later codified by James Martin at IBM in his book *Rapid Application Development*, 1991

Reduces development time by:

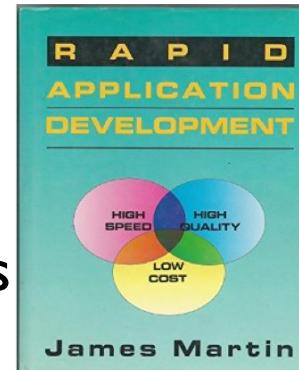
- Substituting prototypes for requirements specifications

- Using existing databases, data models, and tools

- Trading features for speed

- Implementing most important features first

- Adding more features incrementally





RAD Process

- 1. Requirements Planning phase** – Users, managers, and IT staff members discuss requirements
- 2. User design phase** – Users interact with systems analysts and develop reusable (evolutionary) prototypes
- 3. Construction phase** – Users continue to participate and can still suggest changes or improvements as actual screens or reports are developed
- 4. Cutover phase** – Data conversion, full-scale testing, system changeover, user training
- 5. Repeat until done**

RAD Techniques

Facilitated joint workshops (JAD)

Customers and developers

Evolutionary prototyping evolving into finished products

Time boxed incremental delivery

Strict schedules that can't be changed

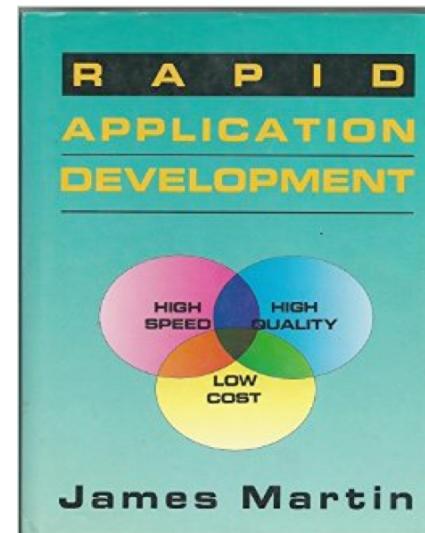
Drop features rather than miss the deadline

Small teams including users (6-12)

CASE tools:

Rapid GUI development, e.g. Visual Basic

Assumes simple data models



Dynamic Systems Development Method (DSDM)

Developed in England in the 1990s

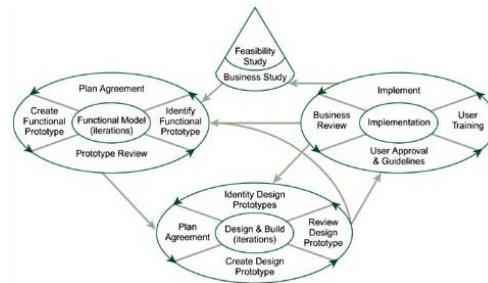
Formalization of Rapid Application Development (RAD)

DSDM Consortium:

Created in 1994 by 16 European companies

Purpose: "jointly developing and promoting an independent RAD framework" by combining their best practice experiences

Publishes standards and supporting material



Dynamic Systems Development Method (DSDM)

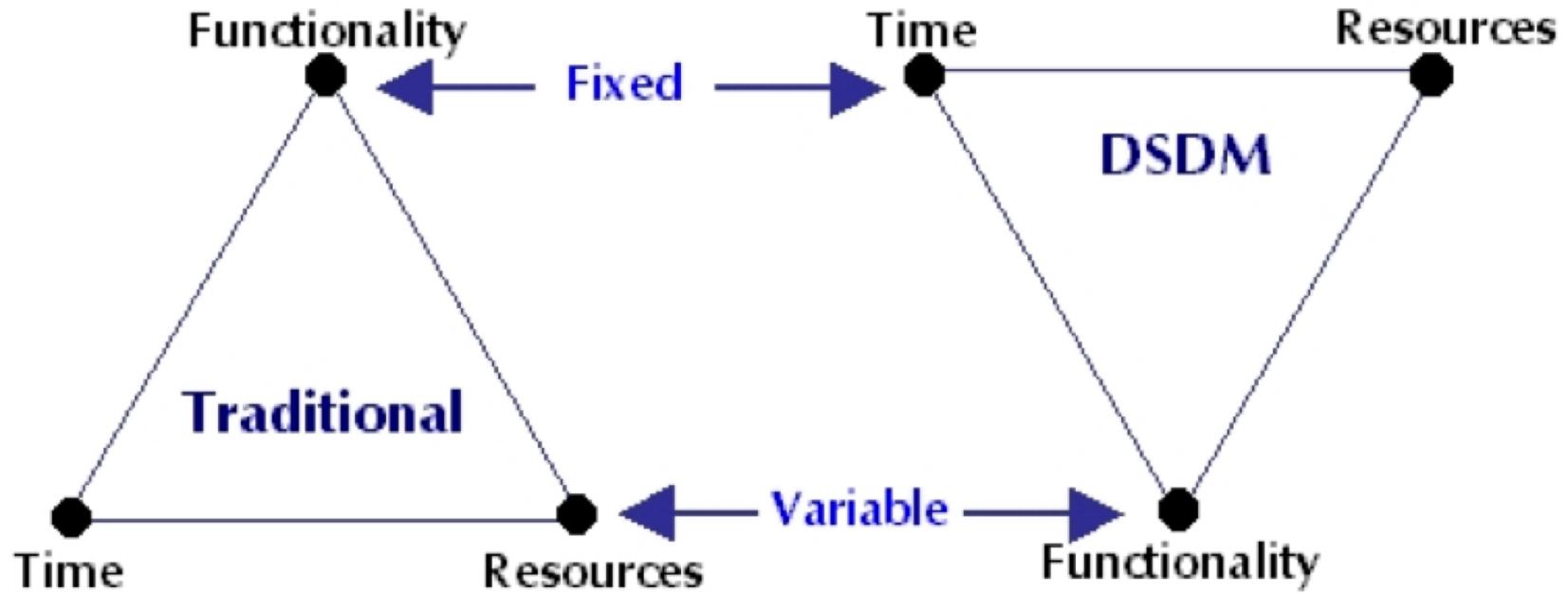


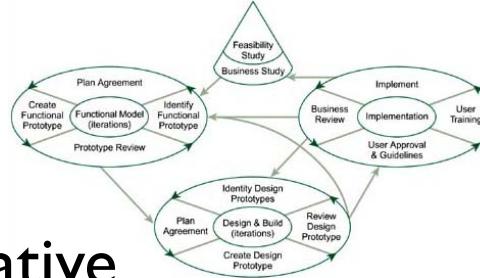
Figure 1

Whereas in traditional development methodologies, functionality is fixed, and time and resources are variable, in DSDM, time is fixed, and functionality are variable.

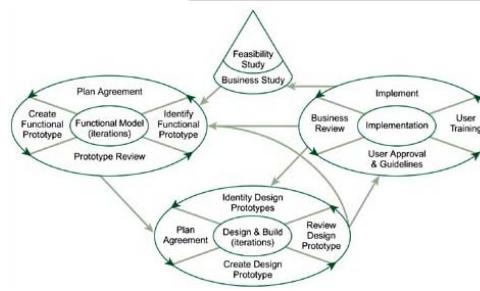
Source: <http://www.codeproject.com/Articles/5097/What-Is-DSDM>

DSDM Principles

- I. Active user involvement** is imperative
- 2. DSDM teams must be empowered** to make decisions
- 3. The focus is on frequent, incremental delivery** of products
- 4. Fitness for business purpose** is the essential criterion for acceptance of deliverables, i.e. meet the business need
- 5. Iterative and incremental development** is necessary to converge on an accurate business solution
- 6. All changes during development are reversible**



DSDM Principles



6. Requirements are baselined only at a **high level**
Use prototypes to define the details
7. **Testing** is integrated throughout the lifecycle
8. A collaborative and co-operative approach between **all stakeholders** is essential, i.e. customer and supplier are on the same team
9. **80/20 Rule** – 80% of the solution can be build in 20% of the time. Prioritize the other 20%

DSDM Process



Source: Highsmith chapter 18

Pre-project phase

Develop a high level concept

Decide whether to start project

Does it make sense?

Is it technically feasible?

Identify stakeholders, especially:

Visionary – driving force behind the project

Project Manager – manages the project and the people

Plan feasibility study



Feasibility Study

Starts with a Facilitated Joint Workshop
Includes users and developers

Produces a Feasibility Report:

Short business case

Suitability of DSDM

Early investigation of requirements

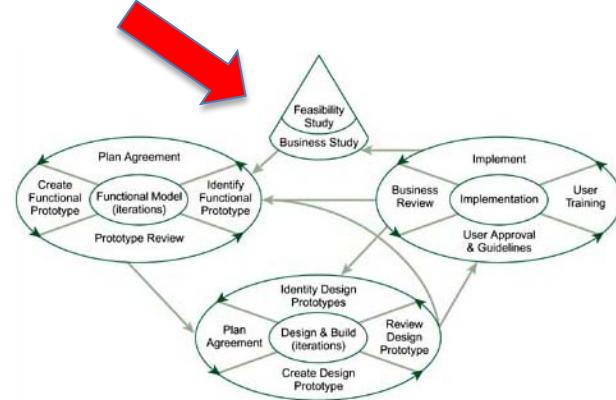
Can it be accomplished with available time and budget?

Also produces:

Outline plan with schedules and budget

Risk log

May produce a first prototype



Business Study

Another Facilitated Joint Workshop

Go down to the next level of detail

Does this make good business sense?

Produces:

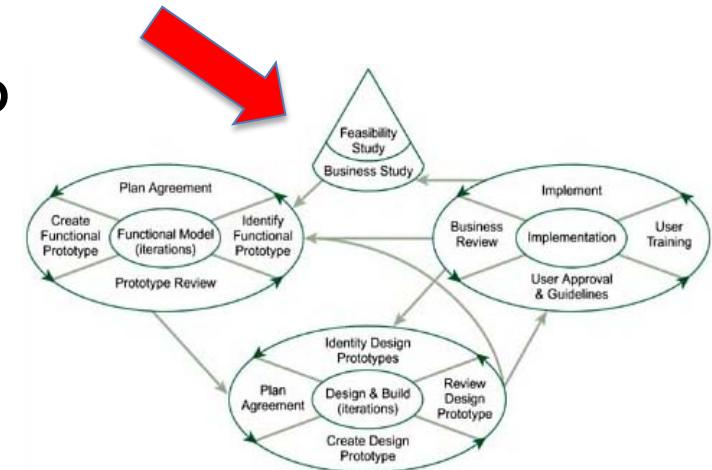
Business Area definition

Prioritized requirements List

Development plan

System architecture definition

Updated risk log



Functional model iteration

Produce functional model of system:

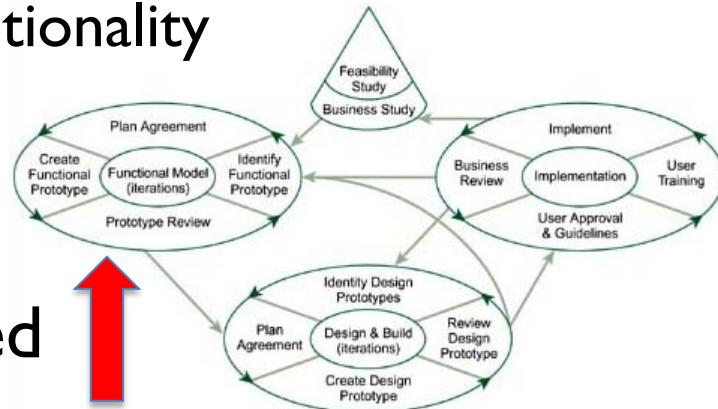
Prototypes

Early versions of the system's functionality

Documentation

4 activities:

1. Identify what is to be produced
2. Agree how and when to do it
3. Create the product
4. Check that it has been produced correctly



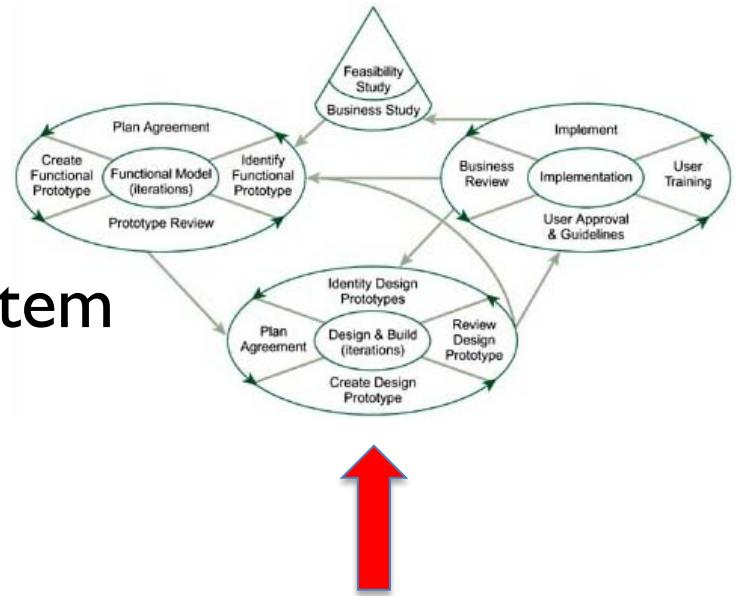
Design and build iteration

Continue to refine the current prototype

System development

Test often

Result is the current release of system



Implementation

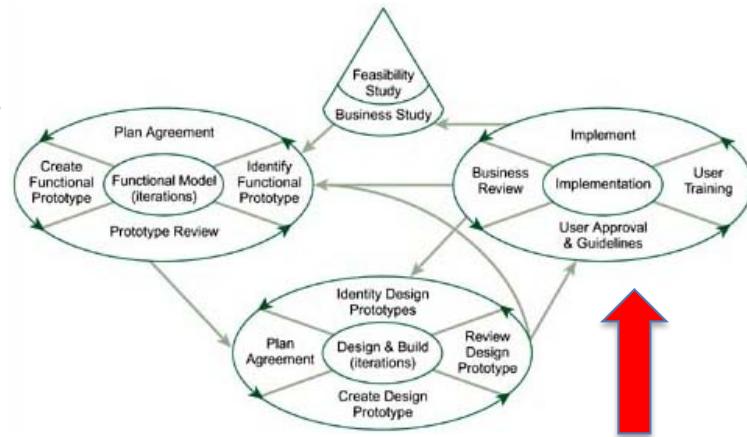
Installation and Deployment

Represents cutover to operations

Installation of the completed code

Produce documentation

Train users



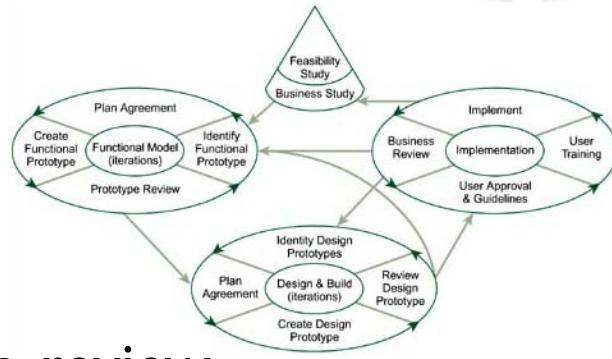
Post-project phase

Support and maintenance of product

Post-project review

DSDM waits until the end of the project to review

eXtreme Programming reviews throughout the project



Core techniques of DSDM and RAD

MoSCoW

Time boxing

Evolutionary prototyping

Facilitated Joint Workshops (JAD)



MoSCoW – 80/20 prioritization

Recall that DSDM trades features for time/schedule

Prioritization of requirements



Must have: system is useless without these features

Should have: important requirements, but there is a work-around so that these can be left out for now

Could have: consider for future iterations

Want to have (**W**aiting list/**W**on't have): least important

At each iteration the team decides which requirements to include

Focus on must-haves (60%)

Must include some others (20% should-haves, 20% could-haves)

Time boxing

Set deadlines and adjust requirements and resources

Each increment has a **fixed** deadline

Each increment should deliver something useful

Development team may **drop some deliverables** in order to **meet deadline**

- Drop requirements

- Acquire third party solution for requirements

- Add resources



Prototyping

Rather than spend all the effort to develop a complete system, build a quick prototype to test its feasibility

Prototypes are widely used in most forms of engineering

Low-fidelity (throwaway) prototyping

Quick demonstration of a feature

Result is not useful for development into a product or service



Evolutionary (rapid) prototyping

Rapid construction of a simple feature, often done with special tools

Result may be modified to evolve into the final product or service

Joint Application Design (JAD)



Developed by Chuck Morris and Tony Crawford at IBM in late 1970s

IBM owned 90% of the computing market in the early 1970s

Bring users and developers together to solve a problem

Facilitated joint workshops:

Small team of users and developers work offsite for 1-5 days

Focus on most important features

May create low-fidelity prototypes

May produce detailed specifications of features and functions

Questions?

