# Prototyping and Rapid Application Development (RAD)

**2 steps forward, 1 step back?**

Jeremy Reece - School of Computing
University of Wolverhampton

email: JPR@wlv.ac.uk



## Contents





## RAD - Background

- Topical in 1990's after
  - Book Rapid Application Development by Martin, J (1991)
- Became latest buzz phrase
- Further books
- Support tools to facilitate it
- DSDM
  - consortium of leading industrial names
  - have produced a RAD method
- Many techniques used have been around for a long time
  - recently brought forward as solutions to IS problems.

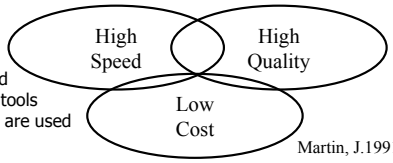


## What is RAD

- RAD is
  - a complete methodology covering systems development from business requirements to ongoing development (not maintenance)
  - a 'tool kit' methodology
  - Open, can be adapted to projects, individual and organisational philosophy
  - can utilise a wide range of techniques and tools

## RAD - Goals

- Radically changes way systems are developed with goals of.
  - High quality systems
  - fast development and delivery
  - low costs
  - These should go hand in hand when the right tools and techniques are used




Martin, J.1991



## RAD - Quality

The definition of quality in a RAD environment is put well by James Martin

*"meeting the true business (or user) requirements as effectively as possible at the time the system comes into operation"*

(Martin 1991)

as opposed to:

*"conforming to the written specification as effectively as possible"*

(Martin 1991)



## RAD - Properties

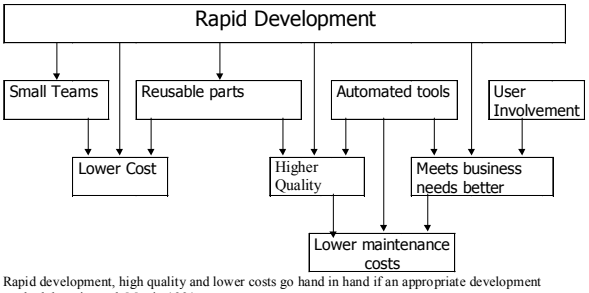
- Does away with the concept of a frozen specification
- Place emphasis on user involvement and responsibility throughout whole development
- Properties
  - Must be delivered in 2 - 6 months
  - split into increments if too large to enable this,
  - each increment is implemented separately with frequent delivery of working parts of system.



## RAD - Cost

- Emphasis placed on low cost
  - All developments should be cost effective though
  - identified as potential expensive option in terms of up front investment (Graham 1995a)
- Organisation may pay more for a quality system in a shorter time-scale
- Evidence all goals are achievable including lowering of costs (Goodwin 1993)

## Rapid Development




Rapid development, high quality and lower costs go hand in hand if an appropriate development methodology is used, Martin 1991



## RAD - Traditional Methodologies

- Goals of RAD diametrically opposed to more traditional methodologies which adopt a waterfall model
- Although quality and speed of delivery are paramount, does not mean what is good in traditional system development is thrown away. There must be
  - Effective project management
  - appropriate up to date documentation
  - requirements specification
  - appropriate maintainability
  - testing
  - quality assurance
  - designs
  - reuse etc



## RAD - Traditional Methodologies (Cont'd)

- DSDM manual (1995) emphasises systems must be build on sound s/w engineering principles
- Key difference
  - these issues are not allowed to dominate business requirements and speed of delivery
  - must be appropriate to project needs
  - e.g. a system with a lifetime of 3 months shouldn't take 18 months to develop



## RAD - Applicability to Organisations

- RAD is more applicable to organisations because
  - World market place is competitive, need right system at the right time to get competitive edge
  - Organisations are dynamic and evolving, requirements change as a system is built, a frozen specifications become outdated
  - IT now viewed as a cost centre not a resource, once system delivered it starts earning money
  - Systems are used by users, if jointly developed by users then more likely to be accepted

## The DSDM View

- A product based view of development is more flexible than an activity based view
- Interactive development is important to developing systems rapidly
- Rapid application development must involve user involvement

An attempt to provide a framework/method (DSDM 1995)



## DSDM Principles

- Development teams - consist of developers and users who are empowered to make decisions
- Business requirements paramount
- Developers and users communicate closely through iterative development involving prototypes
- Change is encouraged and is reversible
- All involved must be highly skilled and motivated towards business objectives
- Testing is done throughout development and reviewed by whole team



## DSDM Principles (Cont'd)

- Larger projects - frequent deliverables should be scheduled
- High level scope and purpose of the system should be agreed and fixed early in development
- Relationship between vendor and purchaser must be one of co-operation.



## RAD - Usage

- May be based upon tools and techniques found in other methodologies
- Hence can be superimposed on existing skills
  - although easier for some than others
- Goals and Principles enforce changes in
  - project structure
  - project management
  - team organisation and motivation
  - philosophy and
  - working practice
- All left is modeling tools and some techniques

## RAD - Essentials

- Tools
  - Code generators, CASE tools, prototyping tools and 4GLs
- Methodology
  - to use tools as effectively as possible
- People
  - right skills and talents. Well selected and motivated. End users
- Management
  - not place obstacles, facilitate fast development
- Infrastructure
  - In which fast development can take place

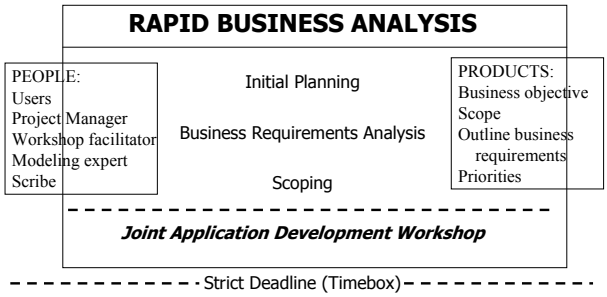


## RAD - Project Structure

- Main changes are
- Rapid Business Analysis
  - Reduced timescale to identify business requirements making the use of JAD workshops
  - Business requirements and systems analysis are undertaken in a fundamentally different way
  - After feasibility study and appropriate research we have a JAD workshop
- Incremental Delivery
  - Iterative development in conjunction with users involving prototyping and frequent delivery of working products



## RAD - Rapid Business Analysis

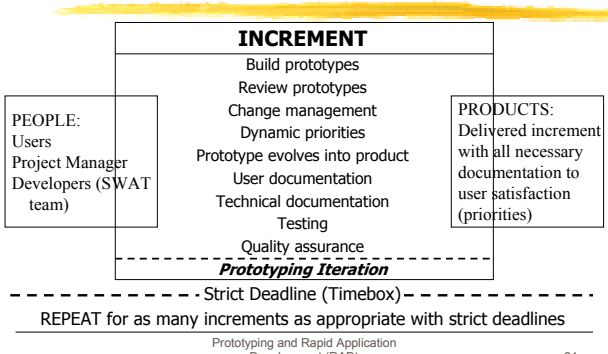





## JAD Workshop

- Takes place away from business environment
- Produces business requirements, fully documented after 3 to 5 days
- Work under the direction of facilitator who must be highly skilled to ensure
  - all issues discovered, refined and reviewed and
  - all political difficulties resolved
  - users and developers have equal influence
- Aim is as much to get common purpose as obtaining system requirements and business objectives

## RAD - Iterative development






## RAD - Iterative Development

- The other fundamental change iterative (or evolutionary) development involves prototyping
- Prototyping has been around for a long time (early 80s)
- Research on has identified trends in prototyping towards
  - better meeting of requirements
  - cheaper cost of production
  - 'Gold plating' - developer introduced features, which are not asked for
    - with undergraduate developments,
    - not found in iterative prototyping were developers in communication and tune with users and there priorities



## Prototyping

"As prototyping has become more widely written about the number of definitions of what it actually represents has substantially increased". (Crinnion 1992).

**This is a bit of a problem!**



## Prototyping

- However Crinnion states that there are now two generally accepted forms:
  - Throw away
  - Incremental (or evolutionary)

  These definitions seem to have become the *de facto* standards, although alternatives are still often quoted

- We use the definition that
  - prototyping takes place during any activity where clients and developers review and refine systems by use of working software

## Prototyping is Essential to RAD

- The characteristics if prototyping can be summarised as
  - Involves animated versions of part of a software system
  - Evaluation in collaboration with clients to optimise quality
  - A joint learning process between users and developers
  - Facilitation of the discovery of errors (testing)
  - May be used for a variety of purposes
  - An approach to systems development which can be combined with other approaches to advantage

(Bates 1989)



## Prototyping

- Prototyping is not a panacea,
  - not magic solution to system development ills
- To gain benefits within a project
  - it must be well structured and carefully managed
- As part of iterative or evolutionary development it is essential to RAD
- Prototyping is viewed to be used for one of 2 scenarios
  - where interfaces are built, evaluated and thrown away or
  - to be part of an iterative or evolutionary building of software with user collaboration



## Prototyping - usage

- Throw-away
  - evaluated with users
  - then developed in a different environment or programming language
- Evolutionary
  - prototype is evaluated
  - refined
  - converges into a project deliverable



## Prototyping Project Structures

- Many have been presented
  - a life cycle for throw-away prototyping, (Dearnley and Mayhew 1983)
  - incremental development and delivery, (Graham 1989)
  - evolutionary development, (Gilb 1988)
  - spiral life cycle, (Boehm 1988)
- Papers on prototyping, prototyping methods, evolutionary and incremental development by
  - Smith 1991, Arthur 1992, Graham 1991, Hial and Soltan 1992, Connel and Shafer 1989 and Vonk 1990.
  - Holloway 1992 draws a single model showing the relationship between incremental, evolutionary and throw-away prototypes

## Prototyping Methods

- All these,
  - are iterative
  - are designed to reduce risk by involving users and developers in a joint development process with good communication assumed
  - assume not possible to know everything about a system up front and then develop it separately
  - assume only possible to get a system partially correct
  - incorporate building of working software, which can be evaluated in conjunction with clients or users
  - encourage change because assume developers and clients not certain of what is required until they are immersed in project and see what is better



## Incremental Development

- Involves splitting projects into smaller mini-projects
- Requires initial investigation and analysis of project as whole
- Increments are self-contained systems, which are
  - developed
  - documented and
  - delivered
- Each increment usually involves iterative development and prototyping



## Prototyping Gives

- Iterative or evolutionary development coupled with prototyping
  - empowers users, feeds in business requirements to development
  - reduces risk, as developers / users can see and communicate on progress
  - encourages communication, trust and communication between developers and users/clients
  - change to improve system is aim and encouraged
  - testing done throughout development by developers and clients
  - provides early project deliverables



## Prototyping - Essential to RAD

- If compare to goals and principles of RAD can see prototyping is essential to RAD
- DSDM has defined four categories of prototype based on purpose
  - Business - to clarify and agree business function
  - Usability - to demonstrate and review user interface
  - Performance and capacity- to check that system will perform adequately
  - Capability / technique - trialing new techniques or environments
- A set of prototypes can encompass more than one purpose

## RAD - Anecdotal Advantages

- It should enable good communication between the user and developer
- Requirement changes can be made at an early stage
- Gradual introduction with increased user acceptance should be possible
- Development time should be reduced, system testing, debugging and modification should be less as the user is kept close to the project at all time



## RAD - Anecdotal Problems

- Lack of standard procedures
- Use of prototype as a production system
- Lack of suitable tools
- Inadequate systems analysis. Prototyping can lead to the wrong problem being focused on.

(Necco et al 1989)

These are typical examples which can be found in the literature



## RAD - Anecdotal Problems

- Parasitic on re-use
- Neglects integration and architecture issues
- Changes of work practices can alienate intended users
- De-motivates people not involved in development
- Problems in estimation and resource planning
- Undue user expectation
- Contractual difficulties



## Survey Evidence

- Necco *et al* (1989)
- Palvia and Nosek (1990)
- Martin and Carey (1991)
- Hardgrave *et al* (1993)
- Hardgrave and Wilson (1994)
- Kinmond (1994)
- Cerpa and Verner (1996)

## Survey - Advantages

- Prototyping will strongly aid learning and communication, both for users and developers
- There is also strong evidence that user satisfaction levels will be higher



## Survey - Disadvantages

- User time required is undoubtedly increased if prototyping is used
- The responsibility of users within a project is increased
- If there are difficulties in ensuring sufficient commitment to a project from users, then there are likely to be severe project difficulties
- Developers can be put under pressure to implement solutions which were never designed to be permanent - these pressures must be avoided
- The entire relationship between prototyping and documentation is unclear



## Survey - Other Issues

- Estimation, management and control procedures appear to be confused areas within the prototyping area
- The problems of iteration and the mapping of conventional management techniques onto evolutionary methods seems to cause unease in commercial software development
- There seems to be little evidence that supports the view of Boehm (1984) that software produced using prototyping leads to improved maintainability



## Our Experiences

- Information Systems
  (manufacturing, small business, local authority, education)
- Decision Support Systems
  (manufacturing)
- Tutoring/Patient Support Systems
  (health care)
- Undergraduate Experiments

## Various Levels of "Success"

- Some successes
- Some notable failures!
- Conclusions that can be drawn



## Observations

On a positive note

- Motivated users can lead to excellent systems
- Communication can be markedly improved
- Incremental delivery motivates users
- Documentation need not be a problem



## Observations

On a cautious note

- Uncritical users will lead to systems failing
- Boundaries will shift
- Control and estimation is difficult
- Function points are not the way forward
- Short delivery times do pressurise developers
- Contractual problems can occur



## Undergraduate Experiments

It should be noted that group ability and group communication through prototyping does influence:

- the quality of the code produced
- the closeness of the produced system to the required specification

This finding seems to echo the evidence of DSDM consortium members (DSDM 1996) who focus on the need for highly skilled technical development teams who are also capable of effective communication with clients and user groups.

## Undergraduate Experiments

- Little evidence from this experiment to back up the claims made by other work that suggests that software built using high levels of prototyping and communication is more reliable and maintainable

This seems to contradict statements made by other authors in the field



## Summary

- New practical initiatives in the area (DSDM)
- Lots of hyperbole
- Not the silver bullet
- Commercial developers must appreciate where problems will occur
- More work is needed in some areas
  (eg. project control, documentation, dynamic contracting)



## References

**Cerpa N, Verner J** (1996) 'Prototyping: some new results'. *Information and Software Technology.* Vol 38, December 1996

**Crinnion J** (1992) 'Exploitation of the 4GL'. *Software Development '92-Management Track.* Blenheim Online, London 1992

**DSDM** (1995) '*DSDM Manual*'. Second Edition Tesseract Publishing: Surrey UK. December 1995

**Hardgrave BC, Doke ER, Swanson NE** (1993) 'Prototyping Effects on the System Development Life Cycle: An Empirical Study'. *Journal of Systems Management.* Spring 1993

**Hardgrave BC, Wilson RL** (1994) 'An Investigation of Guidelines for Selecting a Prototyping Strategy'. *Journal of Systems Management.* April 1994



## References

**Kinmond RM** (1994) '*Prototyping: three perspectives'.* MSc Thesis University of Wolverhampton. 1994

**Martin J,** (1991) '*Rapid Application Development'.* Macmillan Publishing, New York. 1991

**Martin MP, Carey JM** (1991) 'Converting prototypes to operational systems: evidence from a preliminary industrial study. *Information and Software Technology* Vol33 No 5. June1991

**Necco CR, Tsai N and Gordon CL** (1989) 'Prototyping: Use in the Development of Computer Based Information Systems' *The Journal of Computer Systems.* Fall 1989

**Palvia P, Nosek JT** (1990) 'An Empirical Evaluation of System Development Methodologies' *Information Resources Management Journal* Vol 3. Summer 1990