*European Journal of Information Systems* (1999) **8**, 211–223

http://www.stockton-press.co.uk/ejis

# Rapid application development (RAD): an empirical review

P Beynon-Davies[1], C Carne[1], H Mackay[2] and D Tudhope[1]

[1]*School of Computing, University of Glamorgan, Wales;* [2]*Sociology Discipline, The Open University*

Rapid application development (RAD) is an approach to information systems (IS) development which is much discussed in the practitioner literature. However, there is comparatively little research data on this topic. This paper forms a report of the results of a multi-disciplinary research project which has been studying this development approach for the last three years. The paper discusses seven case studies of RAD projects and compares each to issues relating to a number of RAD principles as represented in methodologies such as the recent open standard known as dynamic systems development method. We conclude with a discussion of a number of important questions relating to further research on RAD.

## Introduction

Rapid applications development (RAD) appears to have first become topical with the publication of a text by James Martin with the same title (Martin, 1992). Martin defines the key objectives of RAD as: high quality systems, fast development and delivery and low costs. These objectives can be summed up in one sentence: the commercial need to deliver working business applications in shorter timescales and for less investment.

RAD has been much discussed in practitioner circles, but there appears to be very little academic material assessing RAD. This is not suprising in the context of a systematic survey of the existing literature on information system development methodologies (ISDMs) conducted by Wynekoop and Russo (Wynekoop & Russo, 1997). They found that over half of the 123 research papers examined consisted of normative research in which concept development was not based on any empirical grounding or theoretical analysis, but merely on the authors' speculations and opinions. Of those which constituted empirical research, almost half were undertaken to evaluate ISDMs or parts of ISDMs. Few studies were undertaken to identify how ISDMs are selected or adapted or how they are used. There also appears to be little interpretive research and few practice descriptions or case studies of this phenomenon.

The main aims of this paper are to address some of these limitations in terms of one particular ISDM. The paper provides a review of the practitioner material on RAD and assembles from this material a number of key features of the RAD approach. It then discusses a number of case studies of RAD projects and compares each to issues relating to the key principles of RAD as represented in the ISDM Dynamic Systems Development Method. We conclude with a discussion of a number of important questions relating to further research work on RAD.

## RAD as method

A number of people see RAD as a complete approach to information systems development in that it covers the entire life cycle, from initiation through to delivery. Not surprisingly there are a number of methods available for RAD—such as Martin and more recently in the UK, the dynamic systems development method (DSDM). The DSDM consortium has produced a number of versions of a public domain RAD method (Consortium, 1995). This method seems particularly directed at melding standard development issues such as project management, quality assurance and software testing with the exigencies of rapid development. The expressed aim of the consortium is to remove the 'hacker' connotations associated with what many people refer to as 'first generation RAD'.

DSDM can be characterised as an ISDM in that it provides elements in each of the five areas used to define an ISDM (Avison & Fitzgerald, 1995):

(1) Model of the development process: DSDM utilises an iterative or incremental model of the development process. This model defines four key phases with iteration both within and between phases.
(2) Set of techniques: DSDM emphasises some new development techniques such as joint requirements planning workshops, joint application design workshops and time boxing but generally adopts traditional techniques such as entity-relationship diagrams etc. in a contingent way.
(3) Documentation method: The method expresses a loose set of suggested documentation approaches. The method generally expects that documentation is kept to a minimum within IS projects.
(4) Fit between documentation method and techniques: Some indication is provided in the DSDM manual of how various techniques and documentation stan-

dards can be contingently used in relation to a project.

(5) Philosophy: DSDM utilises a standard philosophy founded in rational business oriented performance. Unusually for an ISDM, there is also some acknowledgement of cultural issues and organizational learning within its description of the method.

Stapleton, in her recent book on DSDM (Stapleton, 1997), includes a number of descriptions of projects taken from the DSDM Consortium's Early Adopters programme. For instance, Scottish National Heritage used DSDM to overhaul its administrative systems. In a similar manner, Irish Permanent used RAD techniques such as joint application design workshops, timeboxing and wash-up sessions to build a system to enable branches to process loan applications. Sema group built a new administrative system for the British Midland frequent flyer programme using a RAD approach. Finally, the UK mobile phone operator, Orange, utilised DSDM in a pilot to upgrade the functionality of the company's system for handling credit card payments.

### *Components*

The following appear to be the common components of RAD approaches discussed in the literature:

*Joint application design (JAD)*

RAD seems to be characterised by small development teams of typically four to eight persons. Such teams are made up of both developers and users who are empowered to make design decisions. This means that all RAD team members must be skilled both socially and in terms of the business. Users must possess detailed knowledge of the application area; developers must be skilled in the use of advanced tools. Hence, 'team-building' activities such as team dinners are seen as an important part of a RAD project. Most approaches to RAD seem to use joint application development (JAD) workshops at various points in the development process, particularly to elicit requirements. In such workshops, key users, the client, some developers and a 'scribe' produce system scope and business requirements under the direction of a 'facilitator'. Development teams are usually expected to come up with fully documented business requirements in three to five days. Such requirements may specify a series of phased deliverables over a given time-span. Further development workshops may be scheduled during the life of a project to develop jointly each deliverable.

*Rapidity of development*

RAD projects seem to be typically of relatively small-scale and of short duration. Also, two to six months is frequently discussed as being a normal project length. The main rationale being that any project taking more than six months to complete is likely to be overtaken by business developments. In total, it has been suggested that no more than six man-years of development effort should be devoted to any particular RAD project. For example, British Rail (Anonymous, 1996b) conducted a RAD project on a mixed Oracle/Cobol system for recording time and attendance of staff. It is claimed to have completed the project in four months rather than the expected twelve months.

*Clean rooms*

JAD workshops are usually expected to take place away from the business and developer environments in 'clean' rooms—that is, places free from everyday work interruptions and full of requisite support facilities such as flip charts, post-its, coffee, computers etc. The emphasis is on highly focused problem solving.

*Time boxing*

Project control in RAD is seen to involve scoping the project by prioritising development and defining delivery deadlines or 'timeboxes'. If projects start to slip, the emphasis in RAD projects is on reducing the requirements to fit the timebox, not in increasing the deadline. Figure 1 illustrates the relationship between the use of timeboxes and the review of development products by teams of users. For instance, the UK Football Association (Anonymous, 1996a) developed three inter-linked information systems for support of the Euro '96 football championship in three very short timeboxes: an information system which stored historical and current information pertaining to the championship for casual users; an operational management system to provide accreditation and media ticketing, VIP management, volunteer management and materials management; the results service which provided information for broadcasters of the events. A hybrid system incorporating PCs, Windows 95, NT, SQL Server and Visual Basic was developed in a matter of a few months.

*Incremental prototyping*

RAD is frequently discussed in terms of incremental prototyping and phased deliverables. Prototyping is essentially the process of building a system in an iterative way. The developers, after some initial investigation, construct a working model that they demonstrate to a representative user group. The developers and the users then discuss the prototype, agreeing on enhancements and amendments. This cycle of inspection-discussion-amendment is usually repeated at least three times in RAD projects, until the user is satisfied with the system. In RAD, prototyping may be used at any stage of development: requirements gathering; application design; application build; testing; delivery.

*Rapid development tools*

It is not surprising to find that modern approaches to RAD demand good support from tools for rapid develop-

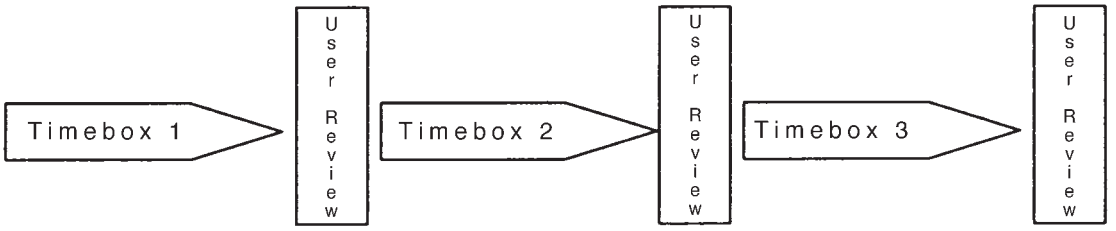


**Figure 1** Timeboxes and user reviews.

mental change. This normally means some combination of fourth generation languages (4 gls), graphical user interface (GUI) builders, database management systems (DBMS) and computer-aided software engineering (CASE) tools. Using such tools some changes to prototypes can be made *in situ* at user-developer meetings. Kerr and Hunter (1994), for instance, describe an early RAD project which utilised Martin's RAD methodology in the development of a financial system for a US bank. The book describes the heavy utilisation of CASE technology on this project, as well as a number of interesting issues such as developer burnout.

*Highly interactive, low complexity projects*

Most RAD projects seem to be conducted on applications that are highly interactive, have a clearly defined user group and are not computationally complex. For example, the UK financial company, Norwich Union (Anonymous, 1996c), produced an electronic trading system originally for the motor insurance sector of the business using an in-house RAD approach. It apparently took the development team only three months to convert this system for the household sector.

The tendency is to rule out the applicability of RAD for large-scale, infrastructure projects, particularly the construction of large, distributed information systems such as corporate-wide databases. Evidence suggests that such infrastructure is best put in place before undertaking RAD projects. Such an infrastructure can then act as a feeder to systems developed using RAD. This is perhaps not surprising when one considers that most RAD tools work off a database in some way. Therefore, the database needs to be created before application development begins.

*Types of RAD project*

There generally appear to be two types of RAD project: the intensive and the phased RAD project. In the highly intensive type of project, a team of developers and users are closeted away in a clean room for some weeks, and are expected to produce a working deliverable at the end of that time. A phased project is one spread over a number of months. Such projects are normally initiated by a JAD or joint requirements planning (JRP) workshop. The subsequent phases of the project are then normally organised in terms of the delivery and demonstration of three incremental prototypes. The aim is to continually refine the prototype into something that is deliverable at the end of timebox.

## Dynamic systems development method (DSDM)

Dynamic systems development method (DSDM) is a non-proprietary RAD method produced by the DSDM consortium, a non-profit-making organization of vendors, users and individual associates of RAD. In December 1995, the consortium had almost 100 member companies (Stapleton, 1997). Its intention is to become the UK and international standard for RAD work. Many vendors of application development tools are committed to it and many companies have now adopted it as their preferred ISDM for RAD projects.

***DSDM principles***

The DSDM Consortium maintain it is based on nine fundamental principles:

(1) Active user involvement is imperative: DSDM sees itself as a user-centred approach. Active involvement by the user community throughout the development project is therefore seen as crucial.

(2) DSDM teams must be empowered to make decisions: DSDM project teams consist of both developers and users. Both groups must be given the power to make key decisions. The developers need to be able to rapidly decide on technical solutions. The business users need to be able to decide upon key requirements for the application.

(3) The focus is on frequent delivery of products: The work of a DSDM project is focused on application products that can be delivered within agreed periods of time. This enables the project team to define quickly the optimal approach to achieving the products required in the time available.

(4) Fitness for business purpose is the essential criterion for acceptance of deliverables: The focus of a DSDM project is in delivering business functionality in the required time. This means that a system may be rigorously engineered later if this is felt fit. Traditionally, the focus has been on rigorously engineering systems to satisfy a requirements docu-

ment, whilst ignoring the fact that documented requirements may be inaccurate or incomplete.

(5) Iterative and incremental development is necessary to converge on an accurate business solution: The key emphasis in DSDM is on evolving a system by incremental steps. Partial solutions may be delivered to fulfil immediate business need. Later versions of a system are built on the basis of lessons learned in the feedback process from users. Only by explicitly reworking a system in this way can an accurate business solution be produced.

(6) All changes during development are reversible: The ability to backtrack to a previous version of a system is seen to be an inherent and important feature of a DSDM development project.

(7) Requirements are baselined at a high level: Requirements are 'frozen' at a high-level by agreeing the purpose and scope of the system without presupposing that a more detailed investigation of requirements is needed.

(8) Testing is integrated throughout the life cycle: Testing is not treated as a separate phase or activity within DSDM. As the system is developed incrementally, so is it tested incrementally. Testing is conducted both to ensure that it is fulfilling business need as well as being technically sound.

(9) A collaborative and co-operative approach between all stakeholders is essential: The nature of DSDM projects in which low-level requirements are not fixed at the outset demands continuous cooperation and collaboration between sponsors, developers and users throughout the life of a project.

***The DSDM life cycle***

There are five phases of development within a DSDM project (see Figure 2):

(1) Feasibility study: This phase considers the feasibility of the project in business and technical terms as well as the suitability of the project for a RAD approach.

(2) Business study: This phase defines the high level functionality and the major business entities affected.

(3) Functional model iteration: This phase is used to construct and demonstrate the required functionality using a working prototype.

(4) System design and build iteration: This phase is used to refine the functional prototype, particularly to meet non-functional requirements.

(5) Implementation: The implementation phase includes the handover to users followed by a review of the project's success.

## Limitations of practitioner material on RAD

Although a large amount of practitioner material has been published on RAD in recent times, such material is insufficient as evidence to allow us to assess the efficacy of RAD for a number of reasons:

(1) Positive reporting: It seems something of a truism that many of the innovations in the information systems area are subject to a degree of 'evangelism'. Therefore, one might expect that the cases of a new approach detailed in the practitioner literature are likely to be substantially made up of those which positively reflect on the RAD approach. This is similar to the experience of Lacity and Hirschheim (1993) in their study of the information systems outsourcing phenomenon. In their study they criticise practitioner case studies of outsourcing for portraying a highly positive view of outsourcing. Wilcocks and Smith (1995) have come to similar conclusions in their study of business process re-engineering projects in the UK.

(2) Folk theory: Harel (1980) has portrayed much of the knowledge in computing as being made up of folk theory—a collection of accepted wisdom which has three major characteristics: popularity, anonymous authorship, and apparent age. Much of the RAD literature takes on the features of folk theory in that appeals are made to the inherent utility of approaches such as the importance of user involvement with little empirical evidence to substantiate the claims made.

(3) Lack of breadth and depth: Much of the material on RAD projects provided in the practitioner literature lacks sufficient breadth and depth to allow any detailed comparison across projects. The cases lack any consistent framework, particularly in terms of presentation and analysis.

(4) Quality and quantity: The practitioner material varies in terms of its quality and quantity. There is also some ambiguity in terms of questions such as the level of user involvement in many examples of such material. This is perhaps not surprising in light of the fact that a substantial amount of this material is produced in support of sales of a particular IS development method, technique or tool.

## Case descriptions

In this section we present seven case descriptions from our research on RAD projects. The descriptions are organised in terms of the following headings: organizational context, development context, system description, development process, use of RAD components, and distinctive issues. We also include a brief description of data collection methods in each case (names of projects and individuals have been changed to protect confidentiality). It must be said that there are some differences in both the quality and quantity of data available between the projects described, but in each case there was sufficient data for comparability across the features

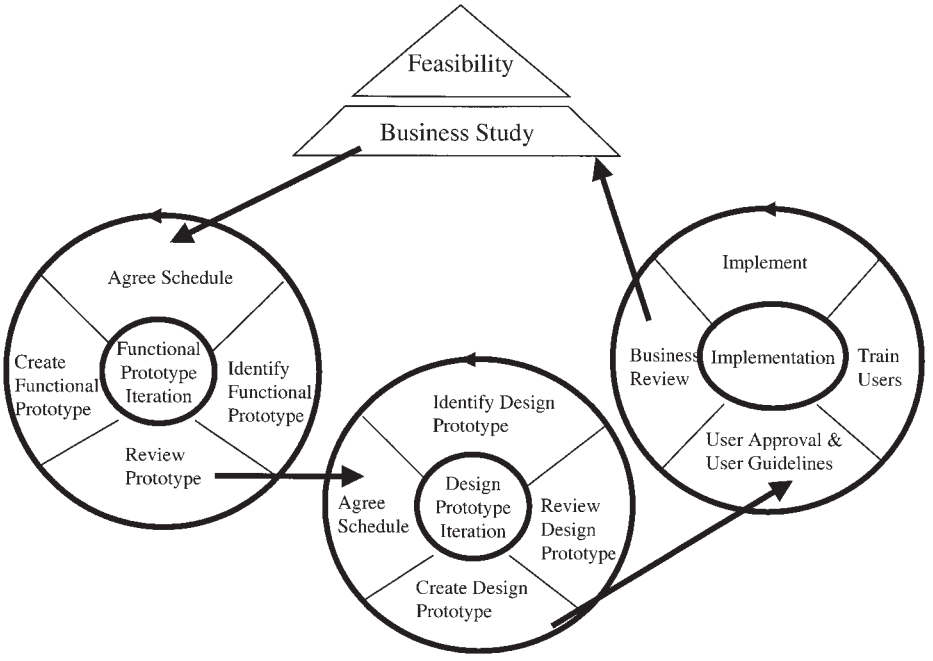


**Figure 2** The DSDM life cycle.

outlined above. Four organizations' experience of RAD was studied between the period 1995 and 1997—two in the public utility sector, one in the financial sector and one in higher education.

### *Case study 1: SWALEC*

*Data collection*

Data was collected through observation at project meetings, interviews with key project participants and documentary analysis.

*Organizational context*

The organization concerned is a public utility company which has its own information systems development department composed of approximately 200 people. At the time of our study, most of the current systems were Cobol-based applications interfacing to a DB2 relational DBMS, running on an IBM mainframe. A large-scale development project was underway to replace these mainframe-based systems with a client-server architecture. The project we studied was not a part of these large-scale developments, although the trajectory of the project was constrained in certain respects by the larger IS strategy of the organization.

*Development context*

Development projects at SWALEC were normally undertaken using an in-house, waterfall-based methodology similar to SSADM. Prototyping was undertaken by a small applications development team (two to three members) working with an object-oriented development tool under the OS/2 operating system. Most of the development conducted by this group involved building small, PC-based, applications that interfaced to the large centralised database system. At the time the study was undertaken the organization did not utilise any formal RAD approach.

*System description*

The system developed was meant to replace an earlier system written by a technologically sophisticated user (we shall refer to him as Royce). This user was the manager of a department which had responsibility for maintaining and servicing a network of utility lines over a large geographical area. The original system developed by Royce essentially stored data relating to the current state of repair of the utility network. This data was collected by repair-workers regularly while working on parts of the utility network and input into hand-held data recorders (Psion Organisers). On return to their base (either headquarters or remote site) the lines-people would download their data into the system using Psion cradles attached to remote terminals. This system enabled Royce to produce reports detailing the tasks needed to be undertaken on the utility network, such as inspection, maintenance or repair, on each working day. These reports were then used by administrative staff to produce daily work-sheets for the lines-people. The main problem with this system was that it utilised aged technology. Royce had developed this system using a mainframe-based application generator a decade or so earlier and both the hardware and software support for this environment was being phased out by the organization. Therefore, the intention of the project described here was to re-write and extend the system using modern appli-

cation-development tools that were supported by the IS department.

*Development process*
The project is perhaps slightly unusual (at least for SWALEC) in that it was initiated and indeed driven by this particularly technologically-literate business user. The development team referred to Royce as their 'super-user'.

To initiate the project, a meeting was held with representatives from all the remote sites under Royce's control. The people attending this meeting were divided into groups and given a morning to produce a wish-list of requirements for the system. In the afternoon, the groups were then brought together and a common wish-list produced.

Following this meeting the prototyping group constructed a first prototype of the system. This prototype was built also on the basis of early discussions between Royce and Jones, the leader of the prototyping group. The developers had initiated the process by constructing a few data entry screens from an examination of Royce's original system.

A short way into the project another business area decided to participate in the development process and the design of the system was changed slightly to accommodate their interests. Two more versions of the prototype were then produced, but formal user reviews of the system were rare. However, Royce was geographically located close to the development team's office and hence would visit them periodically to check on progress. The system was finally completed some four months after initiation with final delivery and acceptance testing at a number of sites included within the process.

*Use of RAD components*
At the start of this project a hybrid JRP/JAD workshop was used. The project was phased over a period of some four months and was staffed by a team of four people (three developers and one user). The team was skilled in the use of the development environment (Object-oriented tool plus DBMS) and were knowledgeable concerning the business issues. Throughout the duration of the project no use was made of a clean room and both developers and user interleaved their work on this project with other activities. The system produced (a line management system) was of medium background complexity and displayed a high level of interactivity. No explicit or implicit use was made of the concept of time-boxing but incremental prototyping was used throughout.

*Distinctive issues*
It is interesting that no explicit reference was made to RAD by the project team or in relation to the project, although there is a sufficient utilisation of RAD elements within this case for it to be seen as a RAD project. Although we have described the project as one involving the migration of a system onto new hardware and software platform, the developers did undertake to include a number of new requirements in the new system. One of the most distinctive elements of this project was the way in which it was driven by a very informed and knowledgeable user. Perhaps because of this, the project seemed to lack any clear and formal structure in terms of the management of the iterative development process.

***Case study 2: BT/Face***
*Data collection*
Data was collected through continuous observation throughout the duration of the project, interviews with key project participants and documentary analysis.

*Organizational context*
The project was conducted in the context of a large private utility company in the UK, and was being conducted for the public relations department (PRD) within this company by members of the IS division. PRD were the department within BT which dealt with all press and public relations throughout the organization. This department was distributed across a number of regions of the UK.

*Development context*
BT has a large, internal IS division which again is distributed throughout the company. Up until comparatively recently all development work was conducted under the auspices of an in-house methodology similar in form to SSADM. Over the last couple of years, a number of 'champions' within the IS division had been promoting RAD approaches. Prior to the project described in this paper, one RAD project had been conducted for the PRD, some twelve months earlier. Both this project, and the one described here, had been identified by such champions as 'flagship' projects for RAD within BT. More recently within BT all development projects now include a RAD suitability assessment in their feasibility study phase. In other words, all development projects now must consider using a RAD approach. The declared aim according to a number of internal RAD consultants is to have as much as 30% of projects developed through RAD approaches by IS developers within this company over the next few years.

*System description*
The system being developed was the second iteration of a system for intra-organizational communication. The system consisted of a diary, project management module, and a project management 'manual'. The term project here refers to a PRD project, not to an IS project. The system was aimed at the control of public relations

campaigns (called variously projects and programmes) against corporate objectives. We shall call this IS project, 'project Face', since there were consistent formulations about the proposed application being used to supply a consistent face for the organization to the external world. The basic objective was to provide an on-line resource for all members of PRD in relation to BT's public relations. The avowed aim was to ensure that 'nobody could do their jobs without having their computer switched on', meaning that all outside communications from PRD should be done with reference to Face.

In very broad technological terms this IS project can be characterised as an intranet development. In terms of RAD, this project was therefore perhaps unusual in using HTML editors and the PERL UNIX scripting language as primary development tools as compared to the usual fourth generation language (4 gl)/computer aided software engineering (CASE)/database management system (DBMS) type of toolkit. It was also distinctive in that both developers and users performed development work of a sort. The two developers did all their work using workstations and PERL (the UNIX scripting language), while customers used PCs for running HTML editors and office applications.

*Development process*

The development team consisted of two developers and five users. Interestingly, the project manager was a business user. Both users and developers were closeted away for three weeks in a clean room and expected to deliver a working system at the end of that time. All the work on the project was conducted in an office geographically remote from both the developers' and customers' normal places of work. All the team members stayed in the same hotel during the working week and some travelled back home at weekends.

Since users and developers shared the same workspace, much use was made of informal design devices such as low-technology prototypes for the design of system functionality. Developers would also make some changes to prototypes dynamically at their workstations while in conference with users.

The day ended with a wash-up session (sometimes referred to as a mop-up meeting). Broadly a wash-up session involved the following activities: (1) review of days progress with regards to objectives set; (2) review of what had not been completed and hence remains to be done; (3) generating and documenting what is to be done the next day, and who is to do it—a so-called 'to-do list'; (4) documenting the ways in which requirements were being met in a log maintained by the project manager.

At the start of each working day team members would first inspect the to-do list and then go off to their individual workspaces to conduct work. Periodically and on an *ad hoc* basis, groups would be formed to address design issues.

*Use of RAD components*

At the start of this project a hybrid JRP workshop was used. The project was undertaken over a highly intensive period of some three weeks and was staffed by a team of six people (two developers and four users). The team was skilled in the use of the development environment and were knowledgable concerning the business issues. All project activity took place in a location remote from both the developers' and users' normal place of work. The developers were solely focused on the project during the three-week period but the users did engage in a certain amount of non-project activity during this time. The system produced (an intranet) was of low background complexity and displayed a high level of interactivity. Implicit use was made of the concept of timeboxing and incremental prototyping with *in situ* modification of prototypes being used throughout.

*Distinctive issues*

IS staff in BT referred to this project as the purest they had conducted. We found later that intensive projects of this kind appear to be quite rare amongst organizations. Other distinctive features are the very short time-span in which work on this project was completed and explicit utilisation of the clean room idea. Another notable facet of this project was the way in which both developers and users clearly negotiated work on this project. This meant that in some respects the roles between developers and users became blurred.

### *Case study 3: Barclays*

*Data collection*

Data was collected through observation at project meetings, interviews with key project participants and documentary analysis.

*Organizational context*

Barclays is a UK financial institution which runs a large development centre in the UK with 3000 staff. At this centre, we were told that whereas the development centre had overall control over large development projects, it lacked control over small to medium-scale projects. It was felt that many projects of this scale were being conducted outside of the auspices of the development centre because of a dissatisfaction with the speed of response of the centre. RAD was therefore being proposed by members of the new technology group within the development centre at Barclays as a possible way of redressing this problem.

*Development context*

Like BT, the development centre at Barclays has a set of documented and detailed procedures for IS development

work. One set of procedures closely prescribes a development approach fitted to a standard waterfall model of the systems development life cycle. Another set of procedures prescribes at a high level the integration between a range of distinct development approaches and techniques. Within the development centre, a small group of people were interested in championing the RAD approach. They managed to get permission to run a 'flagship' project to gain experience of RAD and test the utility of this approach. The eventual aim was to incorporate a RAD 'route' into the organization's methodology handbook.

*System description*
The system chosen for flagship status was a replacement for an older mainframe-based system. Essentially the development team used an on-line analytical processing (OLAP) tool to develop an executive information system (EIS). This EIS enabled managers of financial units to pull down data off a large centralised database which accumulated data on costs associated with such units. The managers were then able to 'drill down' into this data in a number of different ways to monitor unit performance.

*Development process*
It is interesting that the project had started prior to its being chosen as a flagship RAD project. During its life it was adapted to fit, in some ways, the RAD approach as described in the DSDM manual. Two developers worked solely on the project for a period of nine months and called on the services of other development staff within the development centre such as an external and internal RAD methodology expert throughout the life cycle. Only one user was present throughout the project life cycle. Versions of the system were periodically mailed to the user for comment. A small number of user review sessions were held at the development centre, particularly towards the end of the project. An attempt was made to reduce the amount of documentation required by the organization's stipulated methodology, but the developers acknowledged that too much documentation was produced on the project. The system went through a formal delivery and acceptance testing phase, at which point it was handed over to the support group within the development centre. The development team took great pride in delivering what was seen as a critical replacement system on time and within budget.

*Use of RAD components*
There is little evidence of the use of JRP or JAD workshops on this project. The project was phased over a period of some nine months and was staffed by a team of two to seven people consisting of two dedicated developers, one 'power' user and four supporting development staff that contributed to elements of the project work. The team was skilled in the use of the development environment (OLAP tool plus DBMS) and the user was knowledgable concerning the business issues. All project activity took place in a corral within an open-plan office. The developers were solely focused on the project during the nine months but the user had only periodic involvement with the project. The system produced (a branch costs decision-support tool) was of medium background complexity and displayed a high level of interactivity. No use was made of the concept of timeboxing and there appeared to be little use of formal user reviews. Limited use of prototyping was evident in various stages of the project.

*Distinctive issues*
What is remarkable about this project is the way in which despite it having very few explicit RAD features, the organization accounted for the development process in terms of RAD. For instance, it lacked a clear iterative framework and had quite low levels of direct user involvement. However, it must be acknowledged that this was an important project for the promotion of the adoption of RAD within the organization. Following completion of this project a formal RAD route has now been incorporated within the organization's standard development framework.

***Case study 4: BT/research 1***
*Data collection*
This description is taken from an in-house report published by researchers at BT.

*Organizational context*
This project was conducted in the research and development agency of BT. The project team were taken from a group of people investigating new development technologies. The aim of the work described here was to identify some of the problems in utilising RAD in general and DSDM in particular on a practical organizational project. To enable this, the project manager went on a three-day DSDM practitioner course three weeks before the start of the project.

*Development context*
Although members of the development team had experience of techniques such as prototyping and had used rapid development tools such as object-oriented 4gls before, this was the team's first real exposure to RAD as a complete development approach. The team had recourse to an internal RAD facillitator on a periodic basis and all members of the team underwent a one-day RAD awareness training session.

*System description*
The system was described by the development team as an asset broker for development materials. The idea was

to provide a facility to link together a library of software-related assets such as designs, requirements, code and test scripts. The broker, built on a World Wide Web (WWW) platform was intended to enable the search for and retrieval of such assets, thereby encouraging re-use of system components in development work.

*Development process*
The project team was composed of a core of four developers with a part-time person fulfilling the role of project manager and scribe. The team deliberately attempted to maintain a close adherence to the DSDM life cycle. One week was spent in initial study, followed by three one-week functional model iterations, two one-week design and build iterations and one weeks implementation. The project was originally estimated to take some twelve weeks but was eventually undertaken in seven weeks. The team generally worked normal office hours and held a short 'wash-up session' at the end of each working day. At the start of each working week, the team held a planning meeting, and at the end of each week they held a formal review session to which users were invited. Towards the end of the project the development team found themselves working late into the evenings on a number of days to complete the system on time, mainly because of hardware failure.

*Use of RAD components*
A RAD awareness session rather than a true JAD session was run at the start of this project. The project was undertaken over a seven-week period and was staffed by a team of four people (two developers and two users). The team was skilled in the use of the development environment (Object-oriented development environment) and were knowledgable concerning the business issues. All project activity took place in a corral in an open-plan office situated on the developers' site. The developers were solely focused on the project during the project period but the users interleaved their project involvement with other work. The system produced (an asset broker) was of medium background complexity and displayed a high level of interactivity. Explicit use was made of the concept of timeboxing and incremental prototyping was used throughout the project.

*Distinctive issues*
The system was completed on time and within budget. However, the project team seemed to have been extremely concerned about the way in which RAD de-emphasises key software engineering strategies such as formal documentation. They therefore explicitly implemented a system in which paper copies of requirements, designs and screen changes were produced for the explicit signing-off by their users. There was also some concern expressed over the amount of time spent in informal communication between developers themselves and between users and developers on the project. This emphasis may merely be a reflection of the way in which technical rather than business imperatives seem to have driven this project.

***Case study 5: BT/research 2***
*Data collection*
This description is taken from an in-house report published by researchers at BT.

*Organizational context*
Again, this project was conducted in the research and development agency of BT. Also, as well as conducting a systems exercise, the project aimed to test the feasibility and utility of RAD for such work within the organization.

*Development context*
This project is unusual in being a business process re-engineering (BPR) exercise rather than a strict information technology project. The organization had engaged in a number of BPR projects in the past. However, RAD had not been used for BPR within the organization prior to this project.

*Product description*
The aim of the project was to re-engineer the organization's business process concerned with managing research with UK and European universities. This process was estimated as involving £4.5 million of business for the organization at the time the project was conducted. The main output from the project was to be a documented, re-engineered process for this domain. Implementation of this process was planned to be introduced in the next financial year following completion of the project.

*Development process*
The project team consisted of two full-time 'developers', one part-time 'developer' who acted in the role of project manager, and five customers who had periodic involvement with the project. The team also had access to the services of an internal RAD facillitator. The team attempted to use the DSDM life cycle but found great difficulty in distinguishing between functional and design and build iterations in terms of this BPR project. They eventually decided on two major iterations through the process re-design. Timeboxing was conducted on a weekly basis with a planning session on each Monday morning and a review session with customer involvement on each Friday. At the end of each working day a wash-up session was conducted to keep a detailed track of progress on the project. The project was undertaken in eight weeks, six weeks comprising the core timebox, two weeks for RAD training and one week for what the team called 'tidying up'.

*Use of RAD components*
A JRP session was run at the start of this project. The project was undertaken over an eight-week period and was staffed by a team of four to eight people (two full-time IS/IT persons, one part-time IS/IT person and five users). No IT system was produced as output from the project. Instead, a process map for a key business process was produced. Most of the project activity took place in a corral in an open-plan office situated on the developers' site. Explicit use was made of the concept of timeboxing and incremental prototyping of a developing process model was used throughout the project.

*Distinctive issues*
The project was unusual in being a BPR exercise. Because of this, unlike software development projects, the project team found themselves spending 50% of their time on documentation. Also, the team commented that they felt they needed more time with customers than on a software development project.

***Case study 6: UGCS***
*Data collection*
Data was collected on this project during participation in the development process. Audio tapes were made of all user reviews, project diaries were kept by participants and all project documentation was collected.

*Organizational context*
This project was conducted in the context of a small, commercial organization employing approximately forty people. The main business of the company is to sell and deliver training courses to commerce and industry. Prior to this project the organization had no in-house IT expertise. Also, no bespoke systems were in place in the organization although there was much use of shrink-wrapped office products.

*Development context*
The intention of this project was to develop an information system to support the work of the organization's sales and marketing team. A system for sales as well as other functionality was eventually delivered at the end of a six-month period. The emphasis was on producing a working system to a given time-scale, within a finite budget, and with a clear objective of productivity improvement.

The core development team consisted of two developers. One developer acted in the role of project manager/analyst and the other as the programmer. There was also casual representation of the user community in review sessions. At the time the project took place, both members of the development team had no prior experience of RAD. The programmer was also new to rapid development tools.

*System description*
The system was planned to enable sales and marketing staff to maintain a database of company and contact information. In relation to this database, staff would be able to enter details of all contact interactions with the organization. As output from the system, a key element was the need to produce contact lists for direct mailing purposes.

*Development process*
An initial JRP workshop was held with user representatives. This served to scope the project in terms of a six months deadline for the work. The development activity was organised in terms of three timeboxes with each such timebox being terminated by a user review. No use was made of a clean room although the programmer was focused solely on the project. Continuous access to the user community was also readily available throughout the project.

*Use of RAD components*
A JRP session was run at the start of this project. The project was undertaken over a three-month period and was staffed by a team of six people (two developers and four users). The team had some skills in the use of the development environment (RDBMS) and were knowledgable concerning the business issues. No clean room was used for the project. One of the developers was solely focused on the project during the project period but the users interleaved their project involvement with other work. The system produced (a sales and marketing system) was of medium background complexity and displayed a high level of interactivity. Explicit use was made of the concept of timeboxing and user reviews and incremental prototyping was used throughout the project.

*Distinctive issues*
What is interesting about this project is firstly the way in which the project failed on some key aspects of suitability for RAD. For instance, the development team had no previous experience of RAD methods or tools, the requirements were unclear, and the user group was initially ill-defined. Secondly, it is interesting the way in which user representatives became committed to the process of user review and comment quite soon into the process. Having said this, user review sessions tended to be dominated by the viewpoints of one vocal middle-manager.

***Case study 7: University of Glamorgan***
*Data collection*
Data was collected on this project during participation in the development process. Project diaries and all documentation pertaining to the project was collected.

*Organizational context*
This project was conducted in the context of a UK university. The university utilises piecemeal information systems to support the core organizational processes of teaching, research and consultancy. The specific project described here emerged from problems experienced with data collection and manipulation of data for the University's last research assessment exercise submission. The project was initiated with the intention of developing a research database system which would obviate this problem in the future.

*Development context*
The university has an internal information systems department which has been down-sized over the last few years. Consequently, no major IS development takes place in-house. Although the development effort on this project was reasonably small-scale, the IS department seemed reluctant to take on this project because, amongst other reasons, they felt that the requirements were not agreed by the diverse stakeholders.

*System description*
The system was intended to act as a centralised resource for research information such as publications, grants, research students etc. It was intended that the system be used to continually monitor research performance of academic units and to be a key input into the future research strategy for the university.

*Development process*
The development team consisted of two developers and two user representatives. One developer acted in the role of project manager/analyst, the other as a programmer. The project manager had prior experience of RAD, the developer had none, although she had experience of using prototyping and rapid development tools. An initial JRP workshop was conducted with a limited number of user representatives. This enabled scoping of the project within the six months available. Three timeboxes were planned terminating in a user review which was open to all stakeholders in the university. After the third timebox, a period of some three weeks was spent in consolidation work (documentation), testing and training.

*Use of RAD components*
An initial JRP session was run followed by three formal JAD workshops. The project was undertaken over a six-month period and was staffed by a team of four to 24 people (four core members consisting of two developers and two users; 10–20 other stakeholders were periodically involved). The team was skilled in the use of the development environment (RDBMS) and were knowledgable concerning the business issues. All project activity took place in an open-plan office situated at the users' site. One of the developers was solely focused on the project during the project period but the users interleaved their project involvement with other work. The system produced (a research administration system) was of medium background complexity and displayed a high level of interactivity. Explicit use was made of the concept of timeboxing and incremental prototyping was used throughout the project.

*Distinctive issues*
The most interesting feature of this project was the large number of stakeholders affected by the system. Virtually every academic unit and the majority of administrative units had some impact on the system in terms of either supplying data to the system or needing to pull data off of the system. Therefore, user review sessions were as much a forum for informing and consulting with diverse stakeholders as they were opportunities for design.

## Conclusion

***Feature analysis***
The case study material above provides much-needed evidence of the way in which RAD as a development approach has been adopted and adapted in practice. It is notable that in some projects which overtly claimed to be following RAD some major elements of 'best practice' were missing. It is also notable that at least three of the projects described above were early experiments in the use of this development approach. Table 1 provides a comparative analysis of the features across the projects discussed against some of the key principles of RAD. The ticks are a qualitative assessment of the degree to which each of these projects satisfied the principles of DSDM. The crosses are particularly significant in the area of reversible changes. Although the ability to move back to previous versions of a system is cited as a key benefit of incremental prototyping there is little evidence in the projects we have studied of this having occurred.

***Further issues***
RAD particularly raises a large number of questions concerning the appropriate place of this ISDM within IS development practice. In the process of conducting our research IS practitioners have continually reiterated to us their concerns over issues such as:

(1) Cost: Does RAD cost more than conventional development? In our interviews with practitioners many have questioned, for instance, the cost implications of maintaining clean rooms and the greater degree to which business users are involved in RAD projects.
(2) Scalability: All of the projects described in this paper were small-scale. This begs the question, is RAD scalable from small-scale through medium-

**Table 1**

| ***Project*** | *Swalec* | *BT/Face* | *Barclays* | *BT/R1* | *BT/R2* | *UGCS* | *Glamorgan* |
|---|---|---|---|---|---|---|---|
| User involvement | ✓ | ✓✓ | ✓ | ✓✓ | ✓ | ✓ | ✓ |
| Empowerment | ✓ | ✓✓ | ✓ | ✓✓ | ✓ | ✓ | ✓ |
| Product-based development | ✓ | ✓✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Incremental development | ✓ | ✓✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Fitness for business purpose | ✓ | ✓✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Reversible changes | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| High-level requirements | ✓ | ✓✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Iterative testing | ✓ | ✓✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Collaboration and cooperation | ✓ | ✓✓ | ✗ | ✓ | ✓ | ✓ | ✓ |

scale to large-scale projects? Clearly organizations are very interested in whether the benefits of RAD approaches may be achieved in relation to infrastructure as well as interface projects.

(3) Justification: How do you account in formal terms for the business benefits of RAD approaches? Many of the proposed benefits of RAD are intangible benefits, such as greater satisfaction with systems, greater commitment on the part of users to systems. Such benefits are clearly very difficult to quantify and particularly difficult to assess in summative IS evaluation exercises.

(4) Culture: What changes are required in terms of both the organization of development and the organization of business to enable the effective utilisation of RAD approaches? In discussions with a number of organizations this appears to be frequently formulated in terms of a needed culture change both on the part of developers and that of users. Developers need to be more prepared to accommodate user involvement and need better training in the management of such involvement. Users need to be more prepared to be involved in and more committed to the process of development.

These questions represent the rationale for more much-needed research in this important area of IS development.

*Acknowledgements* – The work on this project was funded by the UK Economic and Social Research Council (Grant No: R000 23 5505). We would like to thank all those at the various organizations who allowed us to collect data relating to their RAD work.

## References

Anonymous (1996a) DSDM Scores with Euro 96. *The Computing Guide to DSDM* 16–17.

Anonymous (1996b) On track with British Rail. *The Computing Guide to DSDM* 13–15.

Anonymous (1996c) Radical change. *The Computing Guide to DSDM* 10–12.

Avison D and Fitzgerald G (1995) *Information Systems Development: methodologies, tools and techniques*. McGraw-Hill, Maidenhead, UK.

Consortium DSDM (1995) Dynamic Systems Development Method (version 2).

Harel D (1980) On folk theorems. *Communications of the ACM* **23(7)**, 379–389.

Kerr J and Hunter R (1994) *Inside RAD: how to build fully functional computer systems in 90 days or less*. Mcgraw-Hill, New York.

Lacity M and Hirschheim R (1993) *Information Systems Outsourcing: myths, metaphors and realities*. John Wiley, Chichester.

Martin J (1992) *Rapid Application Development*. Prentice-Hall, Englewood Cliffs.

Stapleton J (1997) *DSDM – Dynamic Systems Development Method: the method in practice*. Addison-Wesley, Harlow, UK.

Wilcocks L and Smith G (1995) *IT-Enabled Business Process Re-engineering: from theory to practice*. In *Proceedings of the European Conference on Information Systems*, Athens, Greece.

Wynekoop JL and Russo NL (1997) Studying system development methodologies: an examination of research methods. *Information Systems Journal* **7(1)**, 47–65.

## About the authors

**Paul Beynon-Davies** is reader in information systems at the University of Glamorgan. He has research interests in information systems development and has written numerous academic papers, professional articles and books on this topic.
**Chris Carne** obtained a doctorate from Brunel University in the sociology of science and was Research Fellow on the research project described here.
**Hugh Mackay** is senior lecturer at the Open University where he researches and writes courses on the impact of new technology.
**Douglas Tudhope** is a senior lecturer at the University of Glamorgan. He has research interests and has published in the areas of hypermedia systems and participating design.