

# A Strategy for Developing End-user Developers: 'I Thought I Hated Every Minute of It, But I Learnt Such a Lot'

*Ann Monday*

*University of South Australia, Adelaide, Australia*

[Ann.Monday@unisa.edu.au](mailto:Ann.Monday@unisa.edu.au)

## Abstract

A continuing challenge for university lecturers is to develop successful programs that challenge students whilst at the same time responding to a 'customer' driven market place. In 2000, two new courses were introduced into a business degree program as a result of feedback from past and present students and the business community. These courses aimed to explore the wider issues of end-users developing small-scale applications to support their role at work whilst possessing a limited skill set in applications development. In each year (2000 and 2001) a different approach was adopted. This paper reflects upon the teaching of these two courses over the two-year period, the responses of students to the different approaches and the extent to which the learning outcomes were achieved.

**Key Words:** End-user developers (EUDs), user-developed applications (UDAs), business students, case study, role- play, student feedback

## Introduction

In 1999, when the Bachelor of Business (Administrative Management) degree program was revalidated, two new courses were introduced, Data Management for Administrators (DMA) and Decision Support for Administrators (DSA). The courses were intended to address a set of issues raised by recent graduates, part-time students and the local business community who were building small-scale applications, usually databases and spreadsheets, with a limited skill set. Although computer literacy courses had introduced the business students to basic IT skills their business program up to this time had not progressed beyond this.

The aim of these courses was to introduce students to the individual and organisational issues of end-user developed applications (UDAs), and to help students to develop good working practices and robust applications using 4GLs. Both courses were designed to provide students with an active introduction to the problems of end-user development of databases and spreadsheets. Students were faced with problems that required them to handle complexity and ambiguity.

This paper examines the evolution of two, second year courses introduced to explore end-user issues with business students. It examines why end-user issues are significant, and explores an approach developed over the two-year period to encourage students to become more active learners and to learn to reflect on their choices and actions.

---

Material published as part of these proceedings, either on-line or in print, is copyrighted by Informing Science. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission from the publisher at [Publisher@InformingScience.org](mailto:Publisher@InformingScience.org)

## End-user Developers (EUDs)

There are many examples of individuals within organisations who identify an opportunity for developing an application to support their own work. Typically, in the past this would have been a

spreadsheet application but we are now finding more and more instances of end-users developing small databases using 4GLs. The growing number of applications produced for business by non-IS developers, generally known as end-user developers (EUDs), has raised a number of critical issues for organisations who are only just starting to realise the positive and negative impacts that end-user developers can have on the organisation.

The introduction of UDAs not only affects the individual or department using the application, it has repercussions throughout the organisation. At the individual level the EUD has the opportunity to develop a software application useful to them and/or their colleagues and normally in a shorter timescale than the IT department. It also offers the individual the opportunity to be innovative and discover new ways of undertaking their role.

These opportunities also bring with them the risk that the EUD may spend too much time in the development of the application, at the expense of their main role, and if it takes too long it may be just as expensive as using an IS professional. There is also the risk that the EUD will produce an application that suffers from a range of common faults and is thus meaningless, and sometimes dangerous, to the organisation. McGill (2000) notes a number of common faults with UDAs including incorrect design, inadequate testing, and poor maintenance. Inadequate documentation is another common fault. Other researchers have noted errors in user-developed spreadsheet applications (Panko and Halverson 1996, Teo and Tan 1999, Janvrin and Morrison 2000, Panko 2000).

It is also important to the organisation that applications are developed in the context of the overall business requirements. Thus EUDs need to be able to fully understand and analyse the business problem, ensure that they are using accurate, complete and relevant specifications, and avoid making assumptions about the organisation's activities.

In exploring the range of skills required by IS professionals, Ross and Ruhleder (1993), suggest that IS educators should provide students with an awareness of a wide range of technical, social and organisational concerns as well a range of IT skills, teamwork and proactive learning to cope with a rapidly changing technology environment. Hobby (1996) noted the need for EUDs to be given some design and implementation training. Although EUDs are unlikely to follow a traditional systems development life cycle approach, the approach adopted should be structured and organised.

End-user computing has not only increased responsibilities for managers in their own departments, it has increased the responsibility of the organisation to establish procedures and good working practices. 'It has also increased the responsibility of the end-user developer, to make sure they apply good working practices when they develop systems for themselves and their colleagues.' (Monday, 2000).

## **The Courses**

The two courses (also described as subjects in some universities) aim to provide students with an understanding of the wider organisational issues of end-user development of small-scale business applications using wizards in spreadsheet and database applications. Each course is 4.5 units and is one of 24 x 4.5 unit courses students undertake to complete the degree program. Semesters are 14 weeks long, with 13 weeks of tuition followed by a one-week self-study period before examinations commence. The first EUD course (DMA) is scheduled in year 2 of the degree program and has a pre-requisite of a first year IT course. The second EUD course (DSA) follows DMA.

### ***Data Management for Administrators (DMA)***

In this course students explore the processes of analysing the information requirements of a business unit and design and build a small database using the wizards in Microsoft Access. This course explores not only practical database skills but also issues relating to the analysis, design and implementation of a data-

base using a 4GL. The technical focus is on using applications generators (wizards) and does not extend to SQL. The course is designed around a simple problem within a case organisation

### **Decision Support for Administrators (DSA)**

In this course students explore decision making in organisations, individual and group decision makers and some of the technology that supports individual and group decision making. Students analyse a more complex problem and develop a small decision support system using Microsoft Excel.

In both courses students are required to analyse a business case, scope the problem, and plan and manage the project as part of a group. They are also required to learn the software, and ultimately choose the appropriate features that allow them to build a user-friendly application. This includes identifying the scope and limitations of the wizards and whether, in fact, the software provides a sensible solution to the problem. Ross and Ruhleder (1993) suggest that students do little experimenting and little evaluating to help them determine whether a given task is better pursued with or without technological assistance.

## **Student Population And Study Modes**

Each course is delivered to large numbers of students in several delivery modes and delivery patterns. The student population for each course by mode of study is shown in Figs 1 and 2.

<b>Fig 1: Student population 2000</b>				
	Onshore Internal	Onshore External	Offshore	Total
DMA	100	40	280	420
DSA	80	20	300	410

<b>Fig 2: Student population 2001</b>				
	Onshore Internal	Onshore External	Offshore	Total
DMA	100	40	300	440
DSA	80	10	300	390

Onshore internal students attend 3 hours of lectures, tutorials and practical sessions. Onshore external students receive hard copy study materials and access to resources on the course web site. Offshore students study in their own country (Hong Kong) at a partner University. Lecturers visit the country for one week at the beginning of the semester and one at the end to deliver mass lectures. Students attend 4 x fortnightly workshops with local tutors employed by the partner University. All students can access lecturers via email, telephone and a web-board throughout the course.

All students are issued with a Course Information Booklet that sets out details of the course objectives, method of delivery, learning outcomes, graduate qualities, assessment details, timelines and policies governing their study. This forms the basis of a learning contract with the students. In addition, a study guide is provided to all students to help them to plan their learning and the stages of the project. The University recommends 10 to 12 hours per week of study for each course taken.

## **The Approach Adopted In 2000**

### ***Case Study***

Both courses are taught via case studies that are based on a business. They are either written in consultation with local businesses, drawn from the industrial experience of staff working with the program or from past students. Case studies are used to tell a story, raise issues for discussion, contain elements of conflict, lack a definitive answer, encourage students' thought processes, require a decision to be made and are reasonably concise (Gross Davis 1993, cited in Kreber, 2001). It was important that students reflected on their choices, approaches and actions from the outset so the business case was used as a vehicle for learning as well as for assessment.

### ***Role Play***

A role-play approach is used to simulate the interaction between the parties to the business case. It allows students to tackle 'live' projects, adopt a role and view a situation from the viewpoint of their role, present arguments, defend their viewpoints and work in teams to develop group skills (Cutler and Hay, 2000). For onshore students, the lecturer and tutors role-played the business; for offshore students the lecturer was the business. The student groups were the end-user developers.

Because students study in a variety of modes (internal, external, offshore) face-to-face role-play with all students was not possible and had to be adapted for use with each group of students. Students were required to work with the business to solve a problem and develop a small-scale database (in DMA) and a spreadsheet decision support application (in DSA). Thus they were involved in communicating with the business, identifying the problems of the organisation, eliciting the information needs of the business and then designing the database and spreadsheet for the business. Students had to learn to handle conflict and ambiguity.

### ***Understanding the Business and Clarifying the Problem***

In both courses onshore internal and offshore students were required to work in groups. In 2000 not all onshore external students had access to the Internet thus group work could not be enforced. Students were allowed to choose their own group members, though group sizes were limited. They were given time to get to know each other early in the semester to facilitate the forming, storming and norming stages of the project. Simple problem solving exercises were used to encourage students to participate.

Once groups were formed they were required to determine the goals of the group, how each member could best contribute to the goals of the group, to identify any likely problems, strategies for avoiding problems, plan a schedule, allocate tasks and establish good working practices. The study guides had been carefully scoped to provide reasonable time scales for each aspect of the project and students were encouraged to follow the structure provided.

In both courses student groups were given a specified length of time in which to clarify the business needs which helped them to get started early in the semester. Student groups submitted questions as an assignment and were evaluated on the quality of the questions they asked. Questions were submitted to the 'business' via email. Questions and answers were then summarised and posted on the web board for all students. Given the large numbers of students the overhead in time was vast for the lecturer running the course and the supporting tutor.

In DMA students then produced a data model, which they used to check the business rules and requirements. The students were thus able to design the rest of the database (inputs, queries, outputs, interface) based upon a stable table design.

In DSA students were required to produce a fully formed problem statement that detailed the current state of affairs, desired state of affairs and central objectives in order to provide them with a clear definition of the problem facing the business (Marakas 1999). They were then required to design a DSS that would inform the decision makers in their endeavour to determine how the problem might be solved. Students were familiar with the concepts of optimisation and satisficing.

### ***Learning the Software***

To enable students to design an application in the most effective way it was essential that they advanced their understanding of the software at an early stage in the program. Thus the program schedule was designed to help the students learn to manage their time. The practical sessions for onshore internal and offshore students were not software training sessions in the traditional sense. Students worked through training examples and brought any problems they had to a help desk. This resource was available to help students to resolve problems and to discuss and question the range of alternative approaches available in Access. External students were provided with a telephone help line for software queries and key issues explored through the internal help desks were summarised and despatched to them.

### ***Designing and Implementing the Spreadsheet and Database Models***

At each stage of the project onshore internal students exchanged experiences, problems were shared and designs demonstrated and critiqued by the group in tutorial sessions. Sessions were summarised and notes made available to external and offshore students, but obviously the richness of the debate was lost in the hard copy. External and offshore students either posted or emailed questions and designs for feedback by the business (tutor). As organisational issues were raised by student comments or actions, these were discussed in tutorials, by email and subsequently documented on the web board. Not all external and offshore students chose to participate in these activities.

## **Course Review 2000**

The course was reviewed on a continuous basis by the course co-ordinator and tutors. Student feedback was gained via email and tutorials throughout the course, from a formal Course Evaluation Questionnaire and in a final examination. The standard university course evaluation instrument, which provides a simple numerical measure of students' impressions of the course without explanation or discussion of the issues that might inform a student's assessment, was supplemented with a range of questions that required students to reflect on their learning, their contribution to the learning, problems they had encountered, how they had resolved them, to what extent they felt they had achieved the learning outcomes and developed the graduate qualities.

Typically, students tended to start by trying to tackle what they perceived to be the problem rather than the actual problem. Students also failed to see that the software was a tool to support decision making and not the end in itself. They saw the tool as the focus and wanted to build an application before they had scoped the problem. Rather than work through the study guide methodically, many chose to adopt a 'suck it and see' approach to learning the software and in both courses students did not initially know enough about the software to implement their database appropriately.

An interesting finding was that students were loathe to discard data that was irrelevant to the problem they were solving because it had been supplied in the case study. In feedback students commented '*but you [the tutor] provided it so we assumed it must be important, we've never been given irrelevant materials before.*' Thus the review at each stage of the project was important in ensuring that students developed an application that was appropriate to the business. However, in adopting this approach, when the students were required to examine the tools and techniques they had used to develop their applications and the problems they might encounter if they adopted an inadequate design their answers showed a lack of any

real understanding. They also showed a poor understanding of their role in the organisation as end-users and the implications of their chosen working practices.

In the final applications submitted for assessment students demonstrated good design skills for screen layouts and reports. The applications were not totally robust but they had achieved a reasonable level of development in the 13-week time frame. Some databases and spreadsheets had poor validations, unhelpful error messages and poor workflow. The emphasis was sometimes on including as many features as possible at the expense of the needs of the user whereas, in fact, the simpler the application the more user friendly they were. Some students viewed the spreadsheet as an 'easy' tool but failed to protect cells, produced formulae that contained numbers and used complicated if statements.

In terms of project management, the real problem lay, not in groups managing the group members as expected, but in managing their time and in the students feedback the majority of students noted a problem of time management. Although they were advised of the time they would need to learn the database software and the importance of working to the study guide they commented they thought it was *'the tutor doing the usual rah, rah, rah'* and adopted the approach they had used for previous courses – start the week before it was due in. They also stated that the course took considerably longer than any other course they studied, but when asked to add up the hours they had committed to the course it came to much less than the university's recommended study time with few exceptions.

A large number of students had not worked with case studies or role play previously. Their feedback showed that they had sometimes found it very hard *'especially when the business kept changing its mind'* but they felt it had helped them to develop the skills they would need to respond to difficult situations though one of the parents didn't agree. At a school function, the tutor was approached by a parent who stated *'it is unfair and unreasonable for a tutor to change their mind half way through an assignment'*. When the tutor explained that this can and does happen in the workplace and it's good for students to have the opportunity to handle this type of situation in a relatively safe learning environment the parent didn't agree. In DSA, the second course, students felt well equipped to deal with the vagaries of the 'business'.

## Student Recommendations 2000

At the end of each semester students were asked to make suggestions as to how the course could be improved. Very mixed responses were received from such a large and diverse student group.

*'You can't do anything for us, you've provided clear details of the learning objectives, assessment requirements, recommended timelines and provided step-by-step exercises to help us to build up a picture of the organisation. It's really up to us to learn to take responsibility for ourselves.'*

A common suggestion was for tutors to allocate marks to every stage of the process. The building blocks were in place and review took place each week but marks were not attributed to each group's work and not all groups had completed as per the schedule. Given the lecturer workload formula allows a maximum of 1.25 hours per student per course for marking, it would be impossible to mark individual pieces of work each week without the need for staff to carry a huge overhead in marking time.

Another very common recommendation was to make tutorials and practicals compulsory for internal students, provide an example of a similar assignment for students to follow, provide faster feedback to the first assignment (questions to the business) teach the software in computer labs, provide easier textbooks and readings and reduce the amount of reading required. The offshore students, with English as their second language, found the theory texts and readings difficult.

## The Approach Adopted in 2001

In semester 1, 2001 a number of the students' recommendations from the previous year were adopted. The texts and readings were replaced with a single text. Help desk support was abandoned and formal practical sessions were introduced for internal students, though the software was still not taught in the traditional 'training' method. Students were required to learn the software prior to class and bring any problems with them for discussion. All classes (lectures, tutorials and practicals) were still optional in that students were expected to attend but this was not formalised in the learning contract (Course Information Booklet).

In addition five major changes were introduced into the courses. Firstly, the case study approach was used as in the previous year but the case used in DSA was an organisation that had adopted a laissez-faire approach to end-user computing.

Secondly, a role play approach continued as in the previous year except that a web discussion board was introduced for onshore students and this was used to liaise with the business. The offshore web board at this time did not provide a discussion forum and questions were submitted via email.

Thirdly, questions to the business were posted to a discussion board and not submitted as an assignment as in 2000. This was introduced in response to students' requests for faster feedback.

Fourthly, students were allowed to follow through with inaccurate data models (DMA) and inaccurate problem statements (DSA). Students were advised they could seek feedback from the business at any stage in the project. Rather than confirm that students had successfully completed each stage and should move on to the next stage, students were allowed to make this decision. Reflection took the form of tutors and the 'business' helping students to question what they were doing, why they were doing it and how they were doing it.

Finally, in semester 2, 2000 and semester 1 2001, a number of student recommendations had not been implemented because past experience had demonstrated the recommendations would not achieve the desired outcome. However, at the end of DMA 2001 when 90% of internal students suggested that their time management skills could be improved with compulsory tutorials and practicals, they were introduced into DSA.

The above changes were prompted by a number of issues - the students' poor understanding of the importance of a stable data model in the design of their database; their poor understanding of organisational issues and their responsibilities as end-users developing applications; and that students continued to fail to manage their time and their studies. Students had identified a prop, compulsory tutorials and practicals that would help them to manage their time. At the same time, other props were gradually being removed to allow students to take responsibility for their actions.

## Course Reviews 2001

Student feedback was gained from formal Course Evaluation Questionnaires, via email and tutorials throughout the course and via a final presentation for internal students (DMA) that required students to reflect on what they had learned from tackling the project, as individuals and as members of a group, and via examination questions.

Although a replacement text had been introduced which proved to be satisfactory for students with English as their first language, those students with English as a second language still found the text difficult.

About half the internal students still opted to attend only a limited number of practical workshops without prior preparation although there was a core group of regular attendees who prepared and contributed well each week. The general comment from students was '*attach a mark to it and we will do it*'. Approxi-

mately half the offshore students attended workshops. In DMA 10% of external students used the telephone help desk and in DSA there were no enquiries from external students.

The case study, based on a live company and describing a laissez-faire approach to end-user computing, was a success. Because students were required to design and build a DSS that would help the organisation to evaluate a number of options that would allow them to manage their end-users more effectively, students found *'the case organisation provided an insight into the real world and the choices and responsibilities of the company and its end-users'*. In the end of semester assessment students demonstrated an excellent understanding of end-user issues for themselves and the organisation.

Questions to the business were not assessed as in 2000 and the quality of the questions dropped. Not all students contributed to the discussion board. External students chose not to submit questions or comments though they did use the web board to download the questions and other materials. The quality of questions from externals the previous year had been very good, but unless a mark is attached to the work they are not prepared to contribute even though it informs the assessed work.

In DMA students were allowed to progress with inaccurate data models although they were advised that time spent in producing an accurate data model would save considerable time later in the development process and lead to a more robust database. Once students began to design outputs, inputs and queries they realised their data models were inaccurate. The initial response was to solving each problem as the error messages appeared. However, after a great deal of time wasting students eventually reviewed their data models and found the rest of the implementation was considerably quicker. The same approach was adopted in DSA and it took students less time to recognise that understanding the problem situation was important.

Internal students acknowledged in semester 1 that their time management was poor and recommended compulsory tutorials and practicals for semester 2. Needless to say, this did nothing to improve their time management skills and students acknowledged this at the end of the year. Some did state that their time management at work had improved. The general feeling from students is that they have a number of commitments, which requires them to prioritise their lives. With few exceptions, university study was ranked last after work, family, sports and social life.

The recommendation for week-by-week assessment was not implemented given the marking time constraints. However, DSA was restructured so that a small group of students each week took responsibility for developing further the tutorial group's understanding of the problem they were solving. Each group had to understand the prior weeks' work as it fed their contribution.

## Final Student Feedback

In addition to the feedback documented above, a general classroom discussion was also undertaken with internal students at the end of DSA. It became evident that this was particularly important as it provided students with an opportunity to listen to the views of other students – both positive and negative. It was in this forum that one of students stated *'I thought I'd hated every minute of it, it was really hard, but I've learnt such a lot, the course was brilliant.'* One of the aims of the course was certainly to challenge students though it wasn't the intention to traumatise them.

Many of the students commented *'I've never done anything like this before, it was really good'* and *'It's made me realise I can achieve a lot more than I thought I could'*. Many commented that it had *'helped them to recognise they must take more responsibility for their own learning'*. There were naturally those who felt *'the course was too difficult'* and *'preferred to write essays that could be completed in a day'*, though interestingly this was relatively few.

Their final recommendation is that students who had already completed the course should come along to the first lecture in 2002 to stress the importance of starting early and developing on a week-by-week basis.

'The tutor was right, it's not possible to leave the preparation and assessment till the last minute.' 'We wasted valuable time ... we started work on it later than we should have.' 'If we had been working for [the company] we would have cost them money due to our time wasting.' 'You should let us come to the first lecture and tell the students, they might listen to us ... er, a hundred students, maybe not.' They have agreed to come along.

## Conclusions

The approach adopted had risks attached to it given the current climate of a 'customer driven' market place. Students were allowed to fail in that they were allowed to progress through each stage whether or not the activities had been successfully completed. They were also allowed to determine how much support they wanted to use. It was certainly risky for the tutors given the university's course evaluation instrument provides a simple numerical measure of students impressions of the course without explanation or discussion of the issues that might inform their choice.

There were greater problems for external students in the second delivery. They chose not to engage with the online discussion and to question the business. In the main poor quality applications were submitted by external students this year though it is not possible to determine whether there is a relationship between the two issues. It is proposed to continue to use the discussion board in 2002 but to assess the questions. Thus all students will be required to contribute.

In the second delivery students demonstrated a good understanding of data modelling approaches and problems they would encounter if data models were not accurate. They also demonstrated a good understanding of problem solving and the organisational issues of developing applications as end-user. There were students who wanted a perfect solution and weren't happy that they had produced what they perceived to be a sub-standard 'product'.

Working with the students for 2 semesters provided a better opportunity for tutors to build a rapport with the students and for students to develop an understanding of the approach used. It also provided tutors with a good opportunity to watch students develop, and to see their growing confidence in tackling unfamiliar problems, though this was more evident with internal and offshore students where tutors met students face-to-face.

It is acknowledged that the approach adopted in 2001 may not be as successful with a different group of students. The courses next year will run with minor modifications as indicated above. Strategies to encourage external students to participate more actively in role play will be explored. It may be, of course, that some external students choose their mode of study because they prefer to work on their own and this also needs to be investigated.

## References

- Cutler, C. and Hay, I. (2000) 'Club Dread': applying and refining an issue-based role play on environment, economy, and culture, *Journal of Geography in Higher Education*, 24,2,179-197
- Hobby, J. (1996) Degrees of Excellence, *Computer Weekly*, 02/08/1996, 32
- Janvrin, D. and Morrison, J. (2000), Using a structured design approach to reduce risks in end user spreadsheet development, *Information and Management*, 37, 1-12
- Kreber, C. (2001), Learning Experientially through Case Studies: A Conceptual Analysis, *Teaching in Higher Education*, 6,2,217-228
- McGill, T.J. (2000), User Developed Applications: Can End Users Assess Quality?, *IRMA International Conference: IT Management in the 21<sup>st</sup> Century, Alaska*, 106-111
- Monday (2000), The reality of teaching large groups of local and international business students to develop end-user applications, *IRMA International Conference: Managing IT in the Global Economy, Toronto*, 362-365

## A Strategy for Developing End-user Developers

Panko, R.R. (2000), Survey of Developers, [accessed on line 17/7/00 at: <http://panko.cba.hawaii.edu/ssr/tables/dvelprsr.htm>]

Panko, R.R. and Halverson, R.P. Jr. (1996), Spreadsheets on Trial: A Survey of Research on Spreadsheet Risks, *Proceedings of the Twenty-Ninth Hawaii International Conference on System Sciences*, Maui, Hawaii, 326-335

Ross, J. and Ruhleder, K. (1993), Preparing IS Professionals for a Rapidly Changing World: The Challenge for IS Educators, *Proceedings of the 1993 Conference on Computer Personnel Research*, 379-384

Teo, T.S.H. and Tan, M. (1999), Spreadsheet development and 'what-if' analysis: quantitative versus qualitative errors, *Accounting, Management and Information Technology*, 9, 141-160

## Biography

Ann Monday lectures in end-user computing for decision support at undergraduate level and information systems for competitive advantage at masters level in the School of Accounting and Information Systems at the University of South Australia, where she has been based since 1997. She worked for a bank and a large pharmaceutical company before joining academia. She previously worked in the UK tertiary sector for 17 years teaching a range of information systems subjects at undergraduate and masters levels. Her current research interests are in the areas of education, email policy, supply chain management, best practice and enterprise systems within the Australian wine industry.