IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITP4401

The Reality of Teaching Large Groups of Local and International Business Students to Develop End-User Applications

Ann Monday
School of Accounting and Information Systems
University of South Australia, North Terrace, Adelaide, South Australia 5000, Email: Ann.Monday@unisa.edu.au

ABSTRACT

There has been a significant change in the needs of our undergraduate business students in the last few years as more and more business graduates are developing PC applications for their own use or that of their department or organisation. The traditional business computer literacy subject that covers an introduction to hardware and software is no longer sufficient to meet the needs of many business graduates but neither is there always scope, or need, for business undergraduates to undertake the depth of learning offered to information systems and computing undergraduates. There is ample evidence to suggest that businesses are only just realising the cost to them of business graduates developing applications. This paper discusses a new subject designed to provide students not only with the practical skills necessary to build a small-scale database using a 4GL but also to recognise the wider departmental and organisational issues related to end-user development of applications. The paper explores the subject design and how the subject was delivered to 400 onshore and offshore students.

INTRODUCTION

End-user computing (EUC) offers countless opportunities to administrative and business managers who realise the need for up-to-date and accurate information. For the purpose of this paper Edberg & Bowman's (1996) definition of end-user computing will be used: 'User-developed applications (UDAs) are defined as any computer-based application for which non-IS professionals (end-users) assume primary development responsibility.'

Since the proliferation of PCs in the early 80s businesses have seen a continuing growth in the use of applications created by end-users for themselves and others in their departments. Frenzel (1992) argues that UDAs have grown because they offer greater user control and increased flexibility and responsiveness for the user. They encourage innovation, reduce the workload of the IT department and enable the user to achieve what they want in a much shorter time-scale. Thus they offer greater control to the user, allowed changes to be implemented quickly and the effects to be seen immediately. McGill (2000) notes that UDAs now form a significant proportion of organisational information systems.

There is evidence however to suggest that businesses are only just realising the cost of end-users developing their own applications. Janvrin and Morrison (2000) suggest that, when identifying and evaluating risks within organisations UDAs are often overlooked. McGill (2000) cites UDAs that may be incorrect in design, inadequately tested, and poorly maintained. Thus the cost to the organisation can be quite considerable unless some measure of control is exerted on the person developing the system and the department managing the process. End-user computing has increased responsibilities for managers in their own departments, and in their involvement in establishing procedures and good working practices at corporate level. 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.

In the early days of PCs, end-user development tended towards the development of spreadsheet applications for decision support. Thus it is not surprising to find that a large proportion of the research in end-user developed applications has been directed towards the development of spreadsheet applications, their use in organisations and risk analysis based on the error rates found in these UDAs.

Chan and Storey (1996) investigated the use of spreadsheets within the organisation to determine the relationship between usage, tasks, proficiency and satisfaction. The outcomes of this research were 'proficiency has a greater impact on the tasks than the tasks have on proficiency; users do not often use the commonly available spreadsheet features; users prefer software packages they know and understand rather than the best package for the task; user proficiency was not related to the importance of the decision made as a result of the spreadsheet analysis'.

Edberg and Bowman (1996) recognised that 'UDAs represent a considerable risk to organizations since users who create applications frequently have little or no training in development methods'.

Other research examined the risks associated with spreadsheet errors. Teo and Tan (1999) found that '…spreadsheet applications were used by an overwhelming 91% of a sample of end-users', whilst Janvrin and Morrison [4] comment that '...as many as 44% of all end-user spreadsheets contain at least one error'. Since 1987 researchers have shown that the problem of spreadsheet errors in UDAs is increasing. Teo and Tan (1999) state that 'This growing concern over spreadsheet errors can be attributed to the increasing popularity of spreadsheets to support important financial analysis, budgeting, and forecasting applications.' They have also seen '...a tendency for end-users to view spreadsheets as simple tools and to be overconfident about the error-free nature of their spreadsheets.'

Panko (2000) has undertaken many studies over the past 2 decades into error rates in spreadsheet production and the subsequent research into this topic. Panko and Halverson (1996) cite studies that look for errors in end-user developed spreadsheets, located in considerable numbers. They have also determined that 'user controls do not seem to approach the level of control that professional programmers have found to be necessary in a similar application'. Teo and Tan (1999) also found that 'inadequate care is taken to design spreadsheets, which in turn makes error detection and correction even more difficult' and the quality of the application is reduced further.

This paper appears in *Managing Information Technology in a Global Economy*, the proceedings of the Information Resources Management Association International Conference.

More recently there has been an increase in the number of end-user developers building small-scale databases. However, there has been, until recently, little research undertaken into end-user development of database applications. The lessons learned from research into end-user developed spreadsheets offers useful insight into end-user development of other applications. With the increased use of database software with 4GL (application generator) ability, it is not unreasonable to assume that the above problems might also arise in database development.

Ross and Ruhleder (1993), in exploring the range of skills required by IS professionals suggest that IS educators should impart not only the technical skills to students but also provide them with an awareness of a wide range of technical, social and organizational concerns. The impact that UDAs are having in organisations would suggest non-IS professionals, who develop applications for their own and their immediate colleagues use, cannot ignore these issues.

Ross and Ruhleder (1993) highlight 4 critical issues, in the education of IS professionals: development of technical skills with application within the business arena; the need to teach students to effectively work in collaborative, team-based settings sharing their technical expertise with persons who obtain a wide range of technical abilities; the need to develop the sensitivity to understand and, when possible, anticipate the organizational impacts of information systems and the need for students to become proactive in learning and adapting to rapidly changing technological environments.

EUDs are certainly required to have some technical skills they can apply to the day-to-day problems arising in the organisation. Business graduates are commonly required to work in a team-based environment with persons of varying degrees of technical knowledge and will be required to adapt quickly to changes in software and technology as the company upgrades to come in line with new technology. There is also the risk for businesses when staff become 'super-users' who, in the long term, could be called upon consistently by other, less experienced, staff members. In this instance the staff member would become a "surrogate IS professional" and may find this increased workload interferes with the daily tasks for which they are employed.

The need for end-users to be given some design and implementation training is highlighted by Hobby (1996) where PC users were transformed into end-user developers of databases using Microsoft Access. Initially the trainers concentrated on the teaching of the skills of Access however it did not take long for them to realise that a small amount of systems analysis training was also required.

Given that organisations are now starting to realise the high costs of staff PCs it will become more and more difficult for organisations that do not acknowledge the need to make staff more aware of the organisational impact of the choices those staff make in terms of software usage.

SUBJECT DESIGN CONSIDERATIONS

The University offers a range of undergraduate business degrees. Over the past three to four years we have seen a growth in the number of business graduates seeking help to develop small-scale databases, particularly those graduates who have found employment in small and medium sized organisations. These graduates had been required to complete one single semester computer literacy subject as part of their introductory business year. Feedback over a period of time from past and present students, local businesses and professional organisations highlighted a growing need for business graduates with a greater understanding of the opportunities afforded by 4GLs, and competency in understanding the business needs and developing small-scale applications for local users.

One IT literacy subject had been sufficient to provide a very basic introduction to the standard PC software (WP, spreadsheet and database). However, in the majority of instances recent undergraduates can demonstrate at least basic proficiency in word processing and spreadsheet although in most instances database skills were limited. The feedback highlighted the increased development of database applications by many of the business graduates as well as other employees within the organisations. In most instances, these staff demonstrated little understanding of the concepts of problem solving, information gathering, analysis, design and implementation issues for databases. Nor did they recognise the implications of the process used in the development of the application, the quality of the applications developed, or the wider organisational issues.

Feedback, in particular, from graduates and part-time students has shown that they are being required to build small-scale databases, primarily using Microsoft Access software or similar 4GL. The computer literacy courses had introduced students to the software's applications generators or wizards and this was their chosen route in developing their applications. It was only when they started to explore the more advanced features of the software that they began to recognise the potential and the limitations the wizards offered them. It was usually as they identified its limitations that graduates would seek help from past tutors. In some instances, the applications they were trying to build could not be achieved effectively, solely by using the wizards; in other instances the problems lay in structuring tables and in recognising appropriate features for the problem they were trying to address.

Although a number of the graduates became proficient in the tools of the software, they had only a limited understanding of design principles in relation to tables, inputs including data validation, outputs and the interface. Thus the applications developed tended to fail in terms of 'user friendliness' and failed to achieve the level of accuracy and efficiency expected in data input and output. Winter, Chudoba and Gutek (1997) suggest this is likely to be caused, partly by the lack of attention paid to the role of IS literacy in helping the end-user to be efficient and effective.

Graduates also found, in many instances, they were building database applications not only for their own use but also for their immediate colleagues. They were therefore not simply building an application for tasks they were reasonably proficient in but also faced the challenge of gathering information, analysing, designing and building applications around processes and information needs they were not completely familiar with. A need to understand the broader picture quickly became evident. Rollier (1993) identifies awareness of the problems encountered and techniques used in identifying information to be collected and in collecting it to be a prime objective for their students. In addition he identifies the ability to collect data and design appropriate structures for storage and retrieval.

In Banks (2000) he states his aim is 'to encourage students to think, explore and challenge, rather than absorb [his] views of the world'. One of the aims of this subject was to start students on this path with a view to achieving the learning outcomes and thus the graduate qualities identified by the University, which include effective communication and problem solving, lifelong learning and autonomous and collaborative work.

This is a second year, first semester subject. The subject has been designed to provide students with an active introduction to the problems of end-user development of databases.

Students explore the processes of analysing the information requirements of a business unit and design and build a small database. This subject explores not only practical database skills but also issues relating to the analysis, design and implementation of a database using a 4GL. The technical focus is on using applications generators (wizards) and does not extend to SQL. Students are required to learn the software, and ultimately choose the appropriate features that allow them to build a user-friendly database. They are also required to identify the scope and limitations of the wizards and whether, in fact, database software provides a sensible solution to the problem. Ross and Ruhleder [9] 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. All of these issues are examined in the context of the organisation.

The subject is delivered to approximately 420 students, 35% onshore (local and overseas studying in Australia), and 65% offshore (based primarily in Hong Kong). The subject format varies, and is primarily determined by the location of the student. Onshore internal students attend lectures, tutorials and practicals for 13 weeks. External students are supplied with a 'package' of materials and follow week-by-week sessions detailed in a study guide. For offshore students, lecturers visit the country for one week at the beginning of the semester and deliver half the lectures to a mass audience. Students then attend fortnightly software workshops with local tutors throughout the semester. In the final weeks of the semester the lecturer makes a second visit for feedback and revision sessions (again in mass lecture format). All students can have additional access to lecturers via email, telephone and a web-board throughout the semester. The University recommends 10 to 12 hours per week of study for each subject taken. In addition, a study schedule is provided to help students to plan their learning and the stages of the project.

TEACHING/LEARNING APPROACH

The subject is taught via a case study either written in consultation with local businesses, drawn from the industrial experience of staff working with the program or from past students. Because the case study was to be used by students with English as their second language, overseas students studying onshore, who had already developed a working relationship with the lecturer, were asked to highlight any problems and the case study was revised. Careful thought had to be given to the choice of case study to address the wide audience. It was found that the majority of students worked in the service sector with a smaller number working in manufacturing organisations thus a rotation of organisation types is scheduled into the program. It was important that students reflected on their choices, approaches and actions from the outset so the case was used as a vehicle for learning as well as for assessment.

Role play, in different forms, was used with each group of students. The case, typically, revolved around the information needs of a department within a larger business, or a small business. Students are required to work with the business to develop a small-scale application. Thus they are involved in communicating with the business, identifying the problems of the organisation, eliciting the information needs of the business and then designing the database for the business.

Onshore, internal students had the greatest richness in terms of learning through role play, mainly because they received instant feedback (via facial expressions, body language as well as information) through interviews and discussions with the business and through presentations to the business as the system developed. The local onshore students were much more relaxed in their approaches to the business after the initial introductory meeting whereas the overseas students maintained a very formal approach throughout. This was evident too in the approach adopted by the offshore students apart from in one instance where an email was sent which said "don't waste my time, just tell me what to do, and how to do it". The comment provided a useful vehicle for discussing the problems of dealing with a business that was unsure of what they really wanted and who needed guidance from the developer. It is acknowledged that role play is difficult, and it was found to be more so for the overseas students. It did, however, give the students the opportunity to develop their communication skills, learn to seek clarification of their understanding from the business and understand the concerns of the business and the problems they were facing. With support, encouragement and practice even the more reserved students began to contribute to the role play activities.

It was more difficult to set up role play for the offshore and external students. The University has a policy of equal access and, at present, students are not required to have access to the Internet or email. Not all external, onshore students have access to the Internet or email and many offshore students have access only through their local university. The vast majority of these students are studying part-time. Role play for onshore external students was via the telephone or email. For offshore students the email was used. For frequently asked questions, answers were posted on a web board for all students to access.

Students were given a specified length of time in which to clarify the business needs which helped them to get started early in the semester. Questions and answers were then summarised and posted on the web board. Given that the majority of questions from offshore students arrived via email the overhead in time was vast for the lecturer running the subject and the supporting tutor. However, consideration had to be given to the fact that offshore students attended workshops once a fortnight and thus would be able to access the web board and download all the information.

At each stage of the project onshore 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.

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 prior to the session and also explored the help facility in Access. The practical sessions were then used 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 materials from sessions were summarised and despatched to them.

With 400+ students, many not available electronically, it proved impossible to monitor whether students kept up to date with the recommended learning schedule. When the assignment disks arrived complete with databases there were several that showed students had started the practical work the week before submission, or even the day before submission. This raises a number of issues in terms of how the assessment might be structured. Staged assessment would help with this problem but with 400+ students this was physically impossible. Those students who started early and took the time to learn the software thoroughly were reasonably confident when it came to applying their skills to their assignment case. It was interesting to note that in the main these were students in career posts who could see the potential of databases for their own work.

Assignment feedback was provided in the form of a summary of key issues to all students, plus individual comments on students' work. Feedback lectures were also provided for onshore internal and offshore students. Onshore external students received a letter discussing the feedback more thoroughly. Offshore assignments have to be marked within seven days of receipt, which raises another major issue in terms of establishing and training a marking team.

'Lack of validation and misuse of validation and GUI controls was the major weakness across the assignments. None of the databases proved to be one hundred per cent robust but several achieved a good level of development. A number of the students assured us that the wizards were not capable of achieving the full requirements of the users.' (Barker and Monday, 2000), but this was not true. However, once students had explained the options they had explored, and thus demonstrated they had examined the software, a range of alternatives was discussed with them.

It was generally during the later stage of the semester that students started to realise the implications to the organisation of what they are developing in terms of the time involved, understanding the business problem and developing something that is useful to the business. It was at this stage that they suddenly realised how much data would be processed, how well designed or poorly designed their interfaces were, how the data might be used elsewhere in the organisation and the impact that would have if it wasn't accurate, up-to-date, etc. Although none of the applications were perfect, given the time scales of the project, a good quality was achieved by many students.

CONCLUSIONS

There are many issues that still need to be addressed, not least of all how to provide a quality learning experience for all students and how the manage a large group more effectively and efficiently. In the feedback students commented on the amount of time and quality of effort that academic staff had put into the learning process. However, the overhead of staff time was enormous.

Many students rose to the challenge of this subject, although a number of students found the subject difficult. Some students could not see beyond the mechanics of the database and wanted to be told which features to use and when to use them. Many underestimated the time it would take to complete the project. The subject had been designed in building blocks but required the students to manage the blocks on a weekly basis. When students added up the time they had invested in the subject, only a very few had exceeded the University's recommended study time, and in many instances they came far short.

The learning experience for onshore external and offshore students lacked the richness of that available to onshore internal students. The underlying aim was to encourage students to develop a good working relationship with the business by gaining a better understanding of the problem situation and to understand the impact of their actions for themselves and the organisation. This was much harder to achieve with onshore external and offshore students. It is clear that we need to reconsider our learning approach for large groups of students studying in a variety of modes and from different cultural backgrounds.

REFERENCES

Banks, D. (2000), Teaching Information Systems Policy: Electronic Sophism?, *Proceedings of the IRMA Conference*, Anchorage, Alaska, May

Barker, S. and Monday, A. (2000), Business Students in Information Systems: Wizards or Apprentices?, accepted for the Australasian Computing Education Conference (ACE 2000), 4-6 December, Melbourne, Australia

Chan, Y.E. and Storey, V.C. (1996), The use of spreadsheets in organisations: Determinants and Consequences, *Information and Management*, 31, 119-134

Edberg, D.T. and Bowman, B.J. (1996), User-Developed Applications: An Empirical Study of Application Quality and Developer Productivity, *Journal of Management Information Systems*, 13,1, 167-185

Frenzel, C.W. (1992), *Management of Information Technology*, Boyd & Fraser, 294

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.

McGill, T.J. (2000), User Developed Applications: Can End Users Assess Quality?, *IRMA International Conference: IT Management in the 21st Century, Alaska*, 106-111

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

Rollier, B. (1993), The Database Project: Maximizing its Value, *Journal of Information Systems Education*, 5, 2

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

Winter, S.J., Chudoba, K.M. and Gutek, B.A. (1997), Misplaced resources? Factors associated with computer literacy among end-users, *Information and Management*, 32, 29-42