Computer-based tutorials: cost functions and software durability

C. Faye, N.W. Scott & B.J. Stone
The University of Western Australia
http://www.mech.uwa.edu.au/dynamics/

Abstract

A computer-based tutorial system (UWA-CPCS) has been developed which is diagnostic, monitored and networked. At the time of writing, this system had been used in three academic years of teaching: 1995–1997. This tutorial system satisfies many common student needs and allows greatly reduced staff numbers in the classroom. The detail of the form of this system and its educational outcomes are the subject of another paper in these proceedings (Faye & Scott, "Cost-effective computer-based tutorials"). The purpose of this paper is to compare the costs of setting up and running the computer-based tutorial system to the costs associated with traditional tutorial methods.

1 Introduction

To evaluate any new innovation in education we must consider both the 'input' to the innovation (for example the dollar cost of setting it up and operating it), as well as the 'output' (the educational outcomes). It will be appreciated that it is usually easier to quantify the costs than it is to quantify, or even rank, the outcomes. In this paper we ignore the issue of whether the studied tutorial system was an effective educational tool, and concentrate on cost issues.

2 Brief description of the innovation

This paper describes educational development in a particular course at The University of Western Australia, Engineering 100 (Dynamics).

For many years tutorials in this course were of the traditional small-group form: about 20 students with one tutor. In some years up to 15 staff and postgraduate tutors ran parallel tutorial classes. Students had two one-hour lectures and two one-hour tutorial classes each week during the academic year. Students were expected to solve several exam-standard questions each week in the tutorial classes. These were to be handed in to the tutor who would mark them and return them the next week. No formal credit was given for completing the tutorial problems, although the marks were recorded and were occasionally used as evidence of student diligence in examiners' meetings. It was observed that attendance at the tutorials was good at the start of the year, but gradually fell away to a very low rate at the end of the year. Presumably students did not see the tutorials as a valuable study aid and were choosing to invest effort in other activities.

In 1995 Scott & Stone replaced these traditional small class tutorials with an innovative computer-based system (Devenish 1995). This system was networked, monitored and diagnostic. Each student was set a sequence of 200 engineering problems during the year, with unique problem parameters (such as important lengths, velocities and so on). The answer to each problem was a number with units e.g. "3.2 m/s"; these answers had to be typed on the computer screen. If an incorrect answer was entered the computer system was programmed to try to give a diagnostic response rather than a simple "right/wrong" reaction (Scott 1994). These diagnostics were based on observations of typical student error in past years. The computer system provided tools for staff to monitor the progress of the whole class as well as individuals (Scott 1996a). There were regular deadlines and the work counted for 20% of the year mark in the unit. The lecture sequence and style were not changed: there were still two one-hour lectures per week.

In 1996 and 1997 the computer system was extended to include an integrated messaging system (Scott 1996b). Students could attach a question to a problem, and staff (who were on-line) would often respond within a few minutes. A database of comments, queries and responses was thus built up for each of the 200 problems. This meant that staff would only have to answer a given student question *once*, a great saving of time. This messaging layer was called "the forum".

It was observed that students approved of the new tutorial system, giving it good reviews in anonymous surveys. Performance in the formal examinations was also equivalent to, or better than, that of previous years. These results were encouraging enough that the system was used again in 1996 and (in a Web-based form) in 1997 (Scott 1997).

3 Costs

In determining the dollar cost of setting up and running a tutorial method it is not always possible to be precise. For example, although 'on paper' staff might be expected to spend 3 hours per week in the tutorial class, it is possible that additional hours are needed to print course materials, keep student records, satisfy visiting students and so on. In the case of a computer-based tutorial system there are significant costs which are hard to quantify. At UWA the computer room used was a joint facility with another department. How should the purchase and running cost of the computers be shared between the two departments? The analysis presented here is thus meant to be indicative rather than authoritative.

For clarity a *tutorial method* (for example traditional small class tutorials or the computer-based tutorial system) will be called here a *program*. In this paper the economic analysis is intended to be representative of 1996, although some data is taken from other years where necessary.

Opportunity costs are used to give a clearer indication of the differences in costs between traditional and computer-based tutorials. Costs can be divided into start-up costs and operating costs. Start-up costs are incurred only once over the life of the project while operating costs are continuously incurred over the life of the project. The list of opportunity costs identified are described below.

3.1 Start-up costs

The costs of developing the computer package used in the computer-based tutorials are the only start-up costs involved in the introducing these tutorials into learning institutions. No such costs are associated with traditional tutorials. Making-up the total development cost are the three components described below.

i) System development: The development of the system supporting the computer-based tutorials was completed over a two year period (1993-1994). The best estimate of the time spent in the development yields a development cost of $30,000. This is based on the salary of the developer at the time and the hours spent outside work in the develpment. It was considered if a more realistic amount should be used if the opportunities for the development of computer-based tutorials was limited to institutions of higher learning. However, evidence exists in the form of numerous papers describing systems developed for CAL by universities and technical colleges, to suggest that these opportunities arise frequently. Thus it was concluded that basing the system development cost on grants given to the developers is reasonable.

ii) Development of questions: the cost involved in devising and writing the questions to be used in the computer-based tutorials was based on the number of hours it took an experienced lecturer to devise and write questions for the UWA CPCS. The questions were designed to be more novel than questions taken from the text book. The rate of $50 per hour was used to reflect the academic's experience and expertise. Additional questions are expected to be added intermittently over the life time of the project with insignificant costs associated.

iii) Licences: around $500 was paid for all the software used - namely the licence for the language the computer package was written with. This cost also included other miscellaneous start-up costs.

3.2 Operational costs

By far the largest cost component involved in the operation of a tutorial is the cost of the tutors. In fact, this is essentially the only cost involved in traditional tutorials. Tutors are responsible for teaching students during formal tutorials and are expected to make time for students seeking additional help outside these set times. Tutors are also responsible for marking under the traditional system. Marking is made redundant in computer-based tutorials, where on-the-spot marking is completed by the computer. However computer-based tutorials have some additional costs: the expense of printing handout sheets, of maintaining the computer laboratory and of student monitoring.

i) Tutor Costs: in the traditional tutorial method tutors were expected to conduct two tutorials a week with an extra hour every week assigned to marking. The time spent informally tutoring students is a cost incurred in both tutorial systems but is usually regarded as a hidden cost as academic tutors are not officially paid for this work. However, costs are incurred because the time spent with the students represents time that might have been spent completing chargeable work that must be done some other time. Informal tutoring costs were based on a yearly estimate of the time spent in this way obtained from each tutor.

Computer-based tutorials have reduced tutor costs but there is an additional cost of answering questions asked on the forum. Answering the vast majority of the queries has been the responsibility of one tutor who has estimated that this task requires 1 hour every week.

Under the traditional tutorial method the total number of hours staff had to be paid for was about $3 \times 15 = 45$ hours per week. Under the computer-based tutorial system this was reduced to as little as 10 hours per week.

Tutors can either be staff or post graduates. Different rates apply to each. Post graduates (with honours) are paid $53.61 per hour for the first tutorial given each week and $35.74 per hour for the second tutorial. Marking, assumed as routine, is charged at $19.48 per hour. (All figures taken from The University of W.A.'s Casual Teaching Rates: effective 16/10/95). The staff rate was formulated from a typical annual wage, plus an administration charge, of $55,000 and the proportion per year that is spent on activities related to tutorials. The proportion of time devoted to tutorials (i.e. teaching and/or marking) was based on a 35 hour week and a 52 week year. This led to an hourly rate of $30.22.

Each computer was assumed to be in operation 8 hours a day (9am - 5pm), 5 days per week, 39 weeks a year. From summary information obtained for 1995, the average "live time", or the time each student spent logged into the computer, was 2 hours per week. Assuming this figure gives a true indication of the time students needed on the computer, the number of students that can be allocated to a computer can be calculated. While technically, the number of hours a week a computer is available for use (by any student, for any course work) is 40 hours, each student must attend to other university activities such as lectures. This reduces the time that students have access to the computers and a more realistic time of 28 hours a week has been assumed. Thus, 14 students can be allocated to each computer.

ii) Printed Questions: under the computer-based tutorial students were given a printed form of the assignments for each week. This was done to reduce the amount of time students would have to spend in front of the computers. A total of 40 pages were printed per person and the printing cost per page was $0.04.

iii) Computer Laboratory: the idea of opportunity costs meant that expenditure on furniture (chairs and tables), lighting and cleaning were ignored. These costs should be much the same for both traditional and computer-based tutorials.

The costs that were included were the costs of the computers and the associated electricity costs. New computers were valued at $3,000 and are depreciated over 3 years in accordance with the Australian Taxation Guidelines. Therefore, the equivalent annual cost of a computer is $1,000 per year.

Domestic electricity rate of 12.75 cents per kW.hr apply. Thus, the hourly rate for a 250 W computer is 3.1875 cents. Computers were operated 40 hours per week for 39 weeks per year for a annual load of 1,560 hours every year.

iv) Monitoring: this includes monitoring of the computer system as well as the students' performance. The system needed to be monitored in case of software 'crashes' and other bugs which prevent optimal operation. System monitoring costs were expected to be minimal as few problems arose in 1996. This mimics the trend which is expected in the normal operation of an established system. Students falling behind or who were not preforming satisfactorily, were identified and were contacted by faculty. The cost of this activity was estimated at $750 (25 hours per year) and was considered fixed.

3.3 Cost functions

All the variable costs can be expressed as a function of student enrolment in Dynamics 100. Incorporating these variable costs with the fixed costs of operating a tutorial, a cost function can be derived. This cost function can reflect the economic sensitivity of the tutorial method on student enrolment and can also show which tutorial is expected to cost less (and by how much) at different levels of student enrolment. By examining the annual computer-based tutorial operating cost function, it can be determined how many years it will take to recover the initial development costs. The annual savings, derived from using the computer-based tutorial instead of traditional tutorials, can be used to pay off the start-up costs.

Finally, to test the sensitivity of the cost functions of both tutorial methods, two scenarios were investigated and the operational costs under each tutorial method considered for both scenarios. The two scenarios (A and B) represented two possible extremes.

Condition A was:

1. a maximum number of students were allocated to each tutorial: 20 students in a traditional tutorial and 50 students in a computer-based tutorial;
2. only the least expensive tutors are employed, i.e. staff;
3. electricity costs are based on the time that the computers are used in direct relation to the Dynamics 100 unit, which is, on average, 2 hours per week per student. It is assumed that the computers are being used for other activities and costs can be shared with those responsible for the other activities.

Condition B was:

1. the maximum number of traditional tutorials per week are held - 16 tutorials;
2. the smallest number of students are allocated to the computer-based tutorials - 30 students;
3. only the most expensive tutors are employed, i.e. postgraduates;
4. 30 computers are bought to be used solely for Dynamics 100. Therefore, costs cannot be shared and the electricity cost is based on the total time that the computers are in operation, i.e. 1,560 hours per year.

4. Results

i) For condition A (where s is the number of students):

<u>Cost function for traditional tutorial</u>,

$$\mathbf{C_t(s) = \$7,200 + \$226s}$$

<u>Cost function for computer-based tutorial</u>,

$$\mathbf{C_{cb}(s) = \$45,063 + \$48.69s}$$

Break-even student enrolment = 121 students

ii) For condition B:

<u>Cost function for traditional tutorial</u>,

$$\mathbf{C_t(s) = \$7,200 + \$130.5s}$$

<u>Cost function for computer-based tutorial</u>,

$$\mathbf{C_{cb}(s) = \$10,424 + \$103.92s}$$

Break-even student enrolment = 214 students

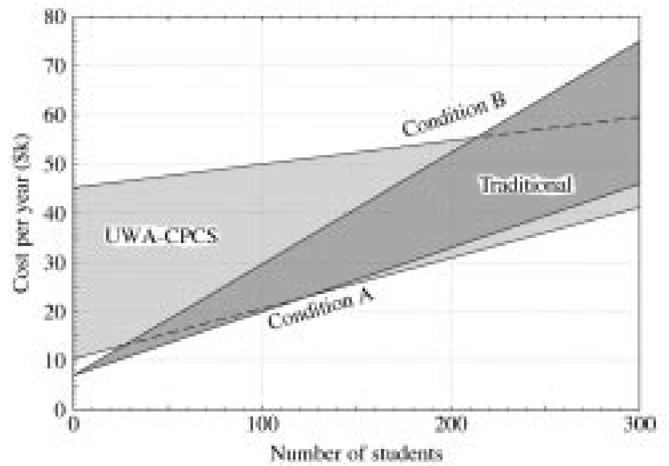

Figure 1 Operating cost functions

A few obvious comparisons can be made from looking at these cost functions:

- Regardless of the conditions assumed, comparatively large fixed costs and small variable costs are associated with computer-based tutorials.
- The differences between the fixed costs and the variable costs for each tutorial method are not as large when condition A applies rather than condition B.
- The fixed costs for traditional tutorials are constant regardless of the scenario assumed.
- For the traditional tutorial method the condition B has a higher slope than the condition A. However for the computer-based tutorial method the condition B has the *lower* slope.
- The traditional tutorials cost less at lower student enrolments.
- Computer-based tutorials cost less at higher student enrolments.

Two plots of the total annual operating costs for each tutorial method at different enrolment levels are shown in Figure 1. The first plot are costs for condition A while the second plot shows costs for a condition B. The plots clearly show the relationship between cost and student enrolment for each tutorial method.

There is no initial start-up cost for traditional tutorials but for computer-based tutorials, initial start-up costs equal $45,000. Below is a table which indicates the number of years needed to recover the initial start-up cost of a computer-based tutorial under the two conditions.

Table 1 Time to recover start-up costs

s	Condition A	Condition B
0	-14.11	-1.20
50	-24.01	-1.57
100	-80.39	-2.26
150	59.63	-4.04
200	21.75	-18.95
250	13.30	7.04
300	9.58	2.97
350	7.48	1.88

This table suggests that computer-based tutorials are not cost effective until a certain level of student enrolment. It also shows that as student enrolment increases above this critical level, the time to recover the initial investment decreases at a decreasing rate.

5 Discussion: software durability

Thomas (1994) has stated that making programs durable is as beneficial as making programs of high quality or low cost. A durable program was defined as "software which remains in use over a number of years despite changes in curriculum, teaching staff and computer operating system" (p 65). It would seem logical that durable computer software will save in operating and maintenance by being reliable and by reducing the cost of updating the system. Thomas also stated that durability was related to quality since "... better quality increases the chance of long term acceptance and utilisation of a product" (p 65); and he specified certain issues that need to be addressed if durability is to be achieved. The importance of academic portability, cross machine portability, useability, reliability and maintainability were explained and ways to successfully deal with these issues were given.

1) Academic portability relates to the computer package use under different instructors. An instructor may decide to dispense with a computer package if:

- it does not fulfil stated teaching objectives;
- it has become redundant due to a change in the curriculum;
- the teaching style has gone out of favour; or
- the new instructor does not like the content of the package, or does not feel comfortable using the package due to the 'not invented here' syndrome.

The tutors of Dynamics 100 have not voiced any major reservations about the use of the computer package, although apprehension has been expressed by tutors of other engineering units. The UWA-CPCS has been introduced into another university in WA – Curtin University, and while the tutors there have not expressed any major dissatisfaction with the computer package, cooperation between the two universities has not been entirely free of problems. Thus, the computer package would seem to have academic portability except for the 'not invented here' syndrome which is rife in tertiary institutions.

2) Cross machine portability takes into account the dynamic nature of the computer industry. It is very likely for the hardware and even the operating system on which the computer package was based to change. Obviously, designing programs that are capable of running on one or more commonly used operating systems is desirable. In addition to the portability of the software between systems, Goodwill et al. (1995) also considered the availability of systems, and the cost and availability of software licences. The 1996 version of the UWA-CPCS was written for Macintosh™ computers and would need to be converted for use on the more common (IBM compatible) P.C.s to gain wider acceptance. At the time of writing one author (Scott) is working on this project.

3) Utility is concerned with the interface between the application and the user. It was noted that "an inappropriate, outmoded or inflexible interface may well cause a teacher to abandon the use of a teaching application and students to become frustrated and dissatisfied". The ease of writing the software and subsequently adding new questions will also affect how appropriate the teacher views the application; while the students' approval of the application will depend on how easy it is for them to use (Goodwill et al. 1995). Goodwill also considered the time required (if any) for students to learn how to use the software. Thomas (1994) gave a few suggestions of how to make the interface more user friendly by designing it to give a degree of consistency: using standard screen formats and colours, having menus which are placed in fixed positions were some of these suggestions. Under the UWA-CPCS neither the students nor the tutors needed any training and in fact both seemed very comfortable using the computers within a few minutes of first using the software.

4) Reliability and Maintainability. In 1996, the UWA-CPCS was very reliable and needed very little maintenance due mainly to the system in which the tutorial was placed.

Thus the 1996 version of the UWA-CPCS can be considered durable except for cross machine portability and the 'not invented here' syndrome. However, these are not major points of contention as the system was designed to be applicable to a wide range of engineering and science tutorial subjects, and the software conversion is well under way.

Analysis of the cost functions developed in section 3.3.3 led to the data presented in Table 1. See Figure 2

The parts of the curves in Figure 16 that have negative recovery times represent enrolment levels where the operating costs of the computer based tutorials are higher than those of the traditional tutorial method, and it is not possible to recover the start-up costs of the computer system. It is interesting to note that under condition B we require a higher level of enrolment in order to break even. However, once this requirement has been met, recovery times are then smaller than under condition A. In the case of a university intending to re-invent the UWA-CPCS the implication is that the development costs will only be recovered if enrolments are 250 students or more and the situation remains constant for a long period of time, perhaps 15 years. In this scenario there is no room for additional development or maintenance of the software or course i.e. the software would have to have an unrealistic level of durability.

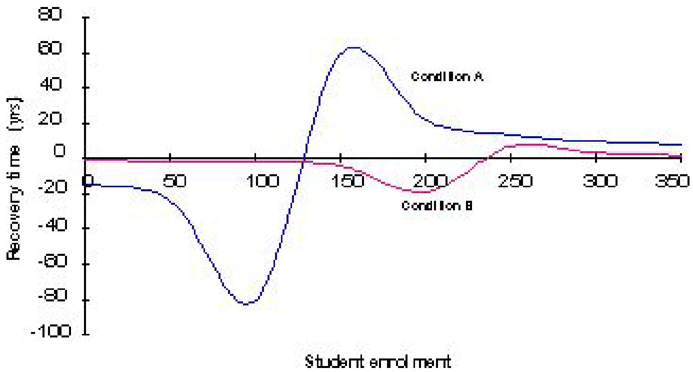

Figure 2 Time to recover start-up costs, taken from Table 1.

From this we can come to a new definition of software durability. Since – under condition A – it may not be possible to recover development costs through in-house use only, durable software *must* be adaptable to other course content, and it must be saleable to other institutions.

6. Bibliography

Brown S.W., McManus J.F. and Lowe, D.W. 1985, 'A Model for Evaluating and Acquiring Educational Software in Psychology'. Paper presented at the Annual Convention of American Psychological Association, Los Angeles, CA.

Carey, John. 1984, 'An Economic Assessment of Electronic Text', Report Number Six of the Electronic Text Report Series, prepared for the Electronic Text Consortium.

Dean, Robert L. 1989, 'A Model for Evaluation of the Cost-Effectiveness of Computer-Assisted Instruction in a University', in Journal of Research on Computing in Education, Spring 1989

Devenish DG, Entwistle RD, Scott N & Stone BJ (1995), 'An assessment package with diagnostic facilities', Proceedings of the International Conference on Computers in Education, Selected Applications Track Papers, Singapore, December 1995, pp 101-107.

Edmonson, William F. 1986, An Instructional Cost Estimation for the XYZ Community College.

Lewis, Darrell and Kallsen, Lincoln A. 1993, 'Using Multiattribute Evaluation Techniques for Assisting Reallocation Decisions in Higher Education.' Paper presented at Annual Meeting of the Association for the Study of Higher Education, Pittsburg PA.

Levin H. 1993, Cost-effectiveness: A primer. Beverly Hills, CA: Skidmore College.

Levin, H. 1994, 'Costs and cost-effectiveness of computer-assisted instruction.' IFG Project Report No. 84-A21. Stanford University Institute for Research on Educational Finance and Governance.

Scott NW, Devenish DG, Entwistle RD & Stone BJ (1994), 'Computer-based error detection in engineering dynamics education', Proceedings ASEE '94 Conference, Edmonton, Canada, pp 2481–2485

Scott NW & Stone BJ (1996a), 'Continuous monitoring and class performance', Proceedings of ASEE '96, Washington DC, 23-26 June, session 2220

Scott NW & Stone BJ (1996b), 'An integrated computer environment for collaborative learning in engineering dynamics', Proceedings of HERDSA '96, The University of Western Australia 8-12 July, pp 755-764

Scott, Nathan. 1996, 'A Study of the Introduction of Educational technology into a Course in Engineering Dynamics: Classroom Environment and Learning Outcomes', unpublished PhD thesis of The University of Western Australia [the full text is available on-line: http://www.mech.uwa.edu.au/nws/nws_phd/]

Scott NW & Stone BJ (1997), 'A flexible Web-based tutorial system for engineering, maths and science subjects', 1st Asia-Pacific Forum on Engineering & Technology Education, Monash University, Melbourne, Australia, 6–9 July 1997, pp 141-146

Thomas, Ruth. 1994, Durable, Low-Cost Educational Software, Pergamon Press Ltd, Great Britain.

Wholeben, Brent E. 1984, Reducing Higher Education Budgets Through MAM, ISEP, New Orleans.

C. Faye

Claudia Faye received a 2B honours degree from The University of Western Australia, Department of Mechanical & Materials Engineering, in 1997. Her final year honours thesis was the subject of this paper.

N.W. Scott

Dr Nathan Scott is a Lecturer in the Department of Mechanical & Materials Engineering, The University of Western Australia. He spends a great deal of time in front of a computer, developing and maintaining computer-based resources such as animations and tutorial systems. He is always willing to correspond by email: nscott@mech.uwa.edu.au

B.J. Stone

Professor Brian Stone has held the Chair in Mechanical Engineering at the University of Western Australia since 1981. He obtained his doctorate from the University of Bristol in 1968. He has been writing teaching software since 1987, some of which is now used at universities throughout the world. In 1997 he was named the best Engineering teacher in Australia by a federal committee. His research interests include vibration suppression and computer simulation of dynamic systems.