

Chapter 6

SOFTWARE EVALUATION CRITERIA - THE USERS

Derek Clarke (IIS)

6.1 Introduction

To be able to evaluate the requirements that define a "good" computer package for irrigation calculations we have to identify the purpose of the software and how practical it is for that purpose. The practicality of the software can be defined in terms of the software's functionality, availability and cost.

- *Functionality:*

- The software should provide a useful, time-saving and acceptably accurate solution to a specified task or problem.
- The software should be tested with a wide range of data sets with several trial users and should be stable and predictable in its behaviour.
- The user interface should be effective in explaining to the user the data requirements and describing the sequence (or sequences) of calculations. At the same time the user interface should not be too complicated.

- *Availability:*

- Potential users should be able to obtain copies of the software from a well organised "sales desk" in the organisation that promotes the software.
- In-house models and packages produced for specific projects are often not suitable or relevant to other schemes unless the package is configured by the vendor.
- Many research papers describe new models and packages but these are rarely made available for use to other organisations.

- *Cost:*

- Prices for modern PC software such as databases and word-processors rarely cost more than \$500.
- The cost of the software development is often very high and it is difficult to recover these costs unless many copies of the software are sold.
- The majority of individuals who require irrigation software have a restricted budget. From experience most engineers from tropical countries consider that \$50/copy is a "good" price for a program, and \$100/copy is "too high".
- Software costs can be a significant restraint unless it is part of a centrally funded software strategy. It is surprising that the cost-effectiveness of the software is frequently undervalued or ignored when there is always a perceived need for more or newer computer hardware.

6.2 Who uses irrigation software ?

There are four main categories of users, defined by the aims of the users, their ability to use the software and their ability to obtain the software.

- *High-technology group*

This is made up from irrigation practitioners who are involved in high-cost commercial irrigation. These users often have definite requirements and are willing to pay a lot of money for software because their business depends on it. Typical of this group are farmers in the USA who often will buy in design expertise from irrigation equipment manufacturers.

- *Researchers and scientists*

Often with a lot of theoretical knowledge, this group is able to appreciate detailed investigation and understand complex models such as finite-element simulations of unsaturated flow. This group is often tolerant of software that is more difficult to use.

- *Developing countries group*

This covers a wide range of irrigation activities including design, management and research. This group often has a restricted budget and cannot afford advanced programs usually have computers that are somewhat older. (It is interesting that in the computer industry an "old" computer is often the new one that was bought 2 years ago).

- *Trainees*

Several organisations provide training in irrigation. Most courses include some aspect of computer use. The author is involved in computer training for 1-year MSc and short courses for irrigation professionals. Such staff are often mid-career engineers who are sponsored to renew or develop their technical skills.

6.3 Problems and pitfalls with software

Users have become familiar with some large complex pieces of software (e.g. EXCEL, WORD) which are well-tested, full of useful (and not so useful) features, have a good user interface and seem easy to use. Users tend to expect all software to behave perfectly and will tend to believe the results of the calculations "because the computer says so".

A user will expect that there is a computer program that will solve all of his problems, whatever the problems may be. I once received a telephone call from an engineer who wanted a full design package for irrigation. ("All I want is a program that asks me for the numbers and then I can type them in and the program will do the design and print out the diagrams and costs.")

Many potential users see a panacea in computer software. It is clear that in many countries staff in research centres prefer to stay in their offices "working" on computer models. On one visit I was told that the office is preferable because it is air-conditioned, more reports can be produced and, anyway, there are snakes in the fields. There is a trend in many academic institutions to sit at the computer (it is cheaper than laboratory or field work).

Many users can now process larger amounts of data in more and more complex models without considering the practical implications of the assumptions made in the computer programs. Finite-difference and finite-element programs can now run easily on present-day PCs, but these models are frequently built using theoretical situations and require calibration before they can be applied. In one research centre I was asked why CROPWAT did not have the crop coefficient data for a specific citrus fruit growing on a sandy soil. I suggested that the researcher try to derive the k_c data for the crop and send it to FAO to add it to their k_c files. This idea was rejected because the FAO k_c values were assumed to be correct and no other values would be accepted.

6.4 Experiences with MSc-course students

Each year a group of 20-30 students attend the 1-year MSc course in Irrigation Engineering at Southampton University. Approximately 60% of the students are from tropical countries and have several years' experience in irrigation; the remainder are usually new graduates from European countries.

At the start of the course each student completes a computer-experience questionnaire to identify his training needs. An analysis of these questionnaires between 1987 and 1995 has shown that more students are getting some experience with computers, although the experience is usually restricted to the use of one or two packages only.

Figure 6.1 shows that in 1987, 70% of students had never used a computer, but by 1993 this had dropped to only 10%. This indicates a growing availability of computers. (In the 1980's up to 50% of the students had no practical computer experience, although the author did find some students with "excellent" grades for computer courses which had been carried out entirely on paper!)

Figure 6.2 illustrates that the majority of incoming students had experience in the use of two or three packages only. The programs most frequently used were word-processors and spreadsheets. Interestingly, there has been a shift away from program "models" such as specific hydraulic packages (19% of students had used these in 1991, but none had used them in 1995).

Experience in computer programming in the main scientific languages is shown in Figure 6.3. This shows a dominance of Basic whilst FORTRAN, originally the main scientific programming language, is less common. The majority had attended a one-term university programming course. A significant trend is that the number of people with good computer

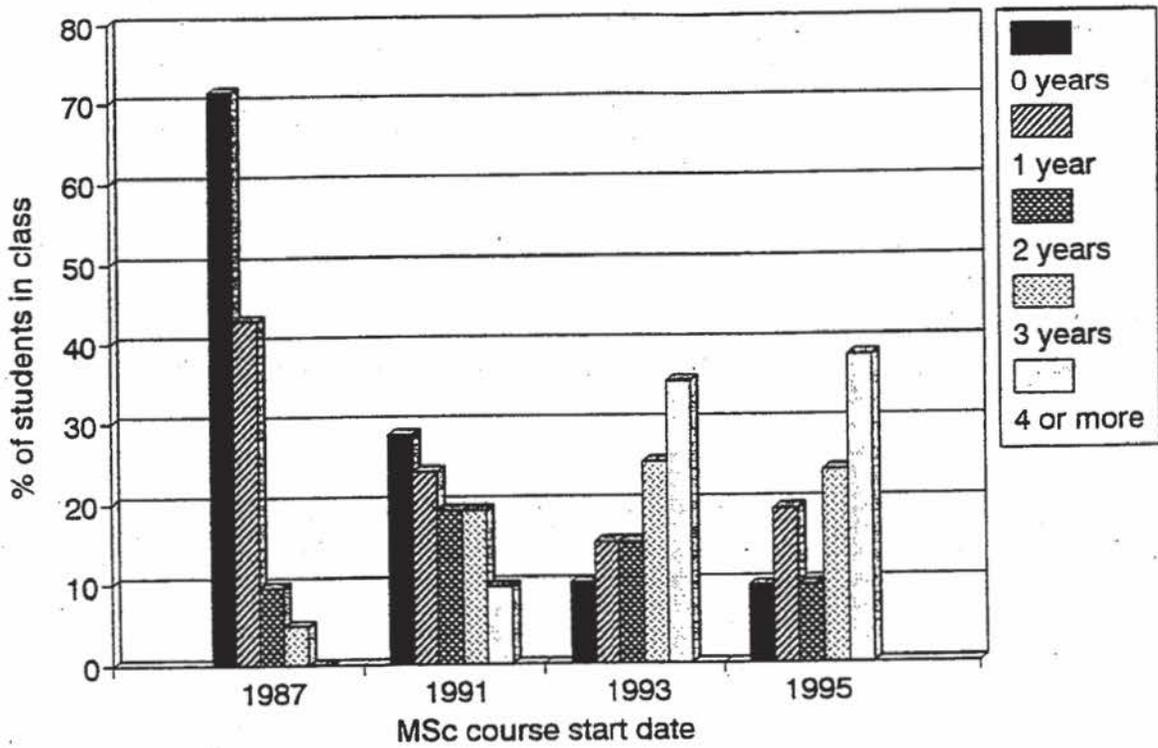


Figure 6.1 Years of computer experience before starting MSc course

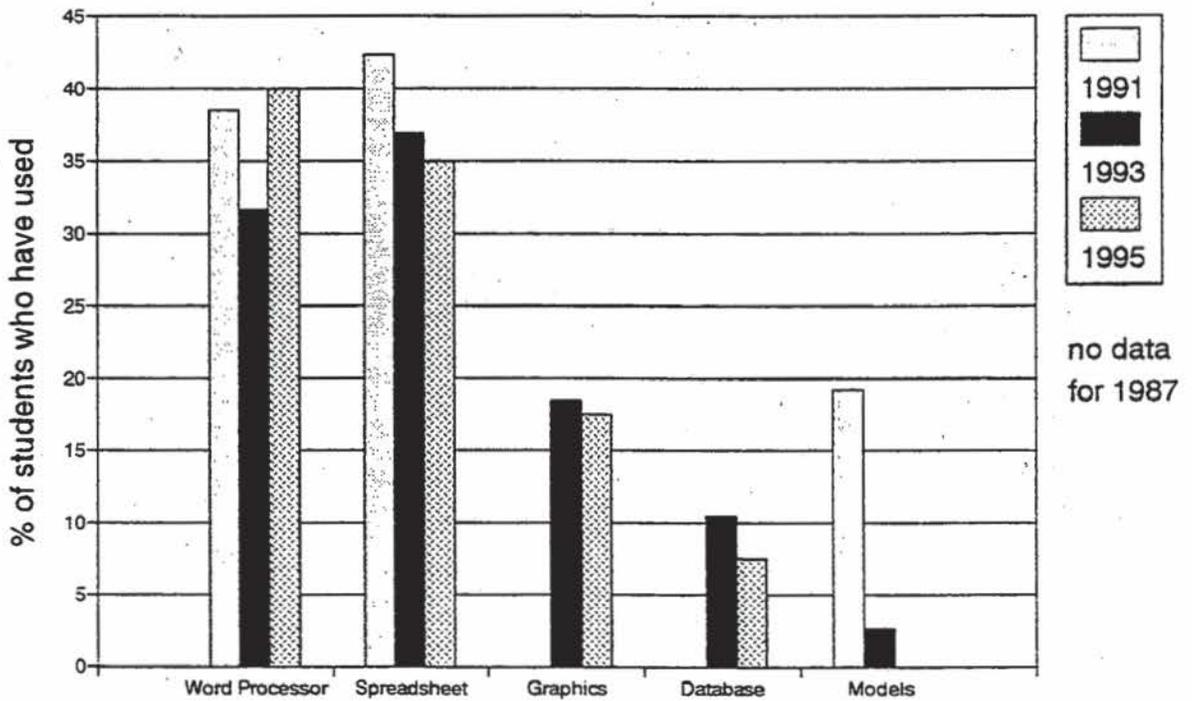


Figure 6.2 Packages used before start of the MSc course

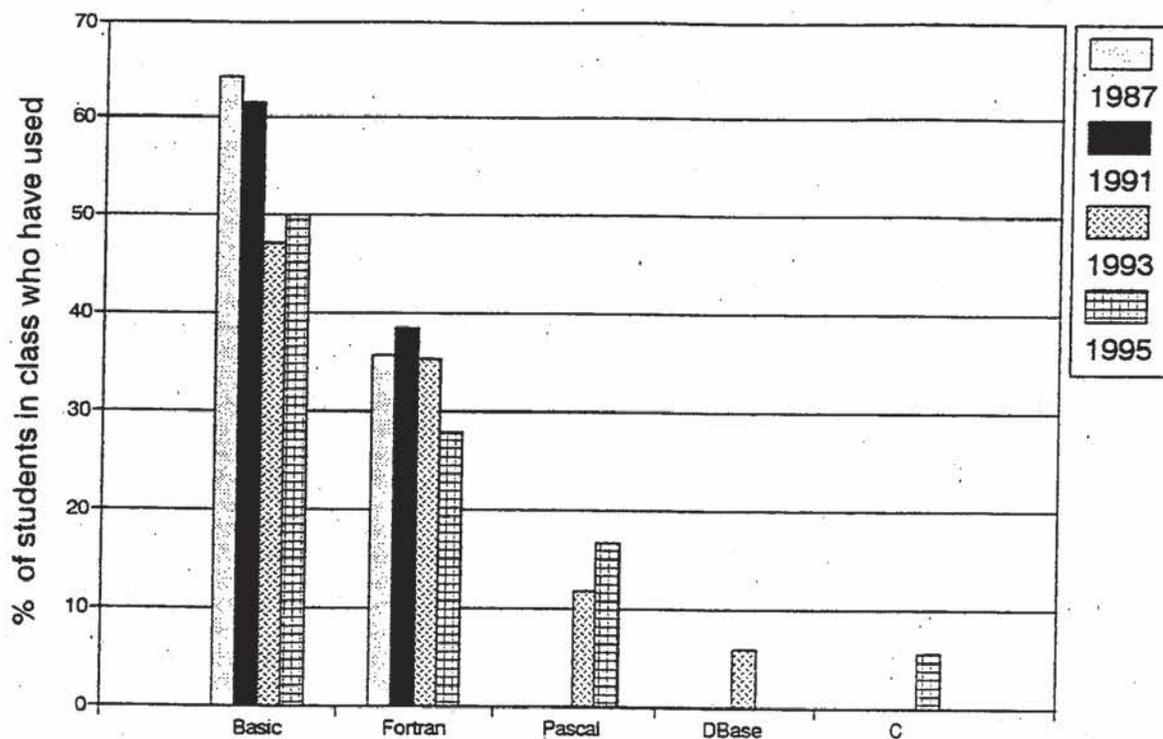


Figure 6.3 Programming experience by MSc students arriving at Southampton

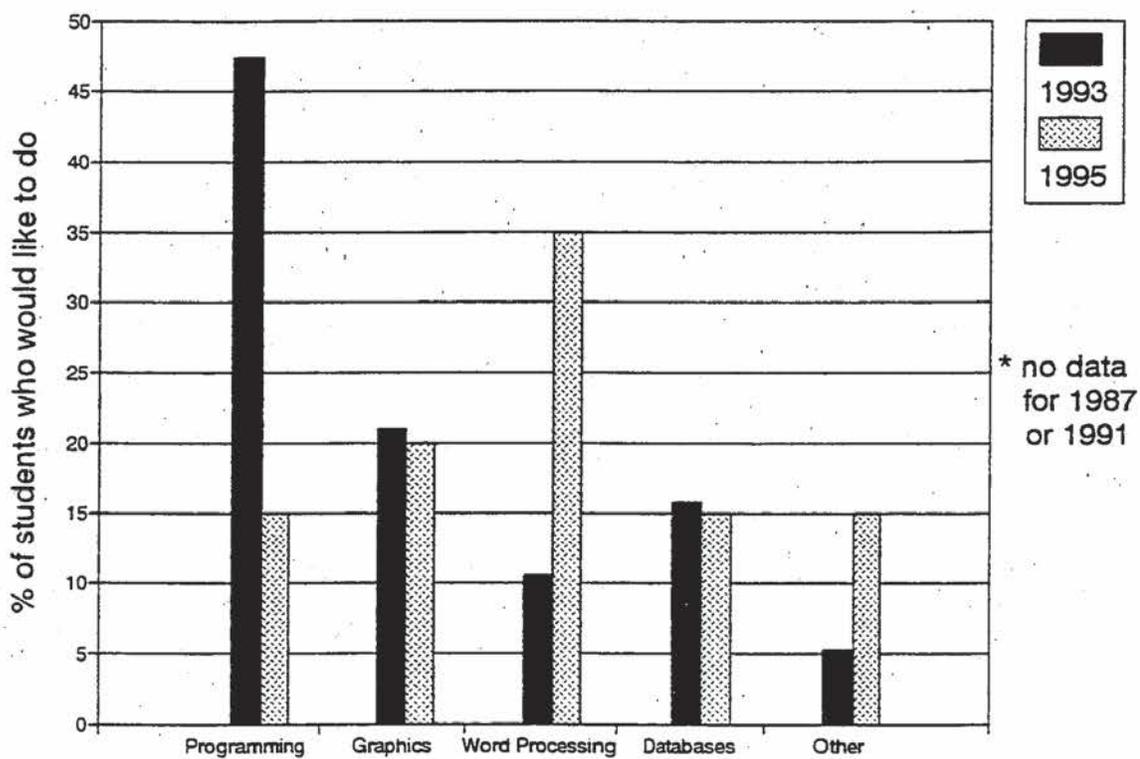


Figure 6.4 Desire for computer training of MSc students arriving at Southampton

programming experience is low and is falling each year. This is reflected by the (limited) data in Figure 6.4 which shows that the desire for training in programming is falling whilst enthusiasm for word-processing is increasing. This leaves us with the question - who is going to write the irrigation software in the future ?

6.5 Irrigation software - catering for the users in the future

It seems that there will be fewer irrigation staff with computer programming experience in the future. Writing a computer program involves having a good understanding of the theory of the method being used and the ability to interpret the results from a program in a critical manner.

Therefore, our future users will have to be taught how to identify flaws in a program rather than slavishly believing the output from a package because it was written by a well-known organisation. These people will need software that is easy to use, tested and calibrated and is well-documented. Hence I suggest that a good software package should be designed with the following points in mind:

- What is the expected technical knowledge of the user?
- Will they have the necessary hardware?
- Will the software be available at a reasonable price?
- Will it install easily (memory, EMS/XMS, ANSI.SYS etc.)?
- Can the user manage with only the manuals (no training available)?
- Will the interface trap obvious data input errors?
- Will the software be appropriate for the intended user (theory too advanced)?
- Is it feasible to collect all the data required by the program?
- Will the user have the time to collect the data and to process them?

From the author's experience with staff from many countries, we cannot expect the typical user to have a very advanced knowledge of computer systems. A common problem encountered is that "program xxxxx worked well until program yyyyy was installed and it changed the AUTOEXEC.BAT. Since then program xxxxx has never worked even though we have a powerful computer." There are many possible reasons for these problems (such as changing a DOS PATH or installing memory-hungry network drivers) but, without an experienced person being available, it is almost impossible to sort the problem out. We should be considering these potential problems when designing the software and the manuals.

Some time back, a review of MS-DOS 5 was made amongst a University Computing Department help desk support staff. They were asked what was the biggest problem with this new release of DOS. One response was

"...that the software was perfect, it is just the users who are the problem."