How to manage the database development for an ecological project, and how not to

F H YUNG

Abstract

Nowadays, it is still commonplace to see ecologists developing database systems in an *ad hoc* and uncontrolled manner. I illustrate this point by describing a hypothetical example. This paper deals solely with methods of managing a database development, as a software development, and not the mathematical techniques used in designing it, such as the normalisation of tables. I then advocate the correct methodology and outline the reasons behind it.

Introduction

An ecologist is well trained in all the biological science aspects of designing and managing an ecological project, from experimental design, field work and finally to the analysis of the data. He will make sure that the data collected can test his hypothesis. However, as information science advances with computers, parallel to the running of such a project, there is always an underlying sub-project, the design and construction of a computer database, which is for storing the data collected from field work. It is important that this database is well designed so that it is easy to extract information for subsequent statistical analysis and mathematical modelling. This demands the database designer to have a sound knowledge of the relational database theory, a specialised mathematical technique in information science. The design issues are the topic of a separate paper by Yung et al (1997) to be submitted shortly and is not discussed here.

Given that you have a competent designer, he may not be an experienced software development manager. Database design and the management of the over all software development are two different topics and are equally important. This paper deals solely with the methods of the latter.

Some ecologists may think that this is an overkill. This is not so, as evidenced by our real example here: Phase 1 of an ecological project has a budget of AUD $ 650,000 over two and a half years, involving six scientists full time. We found it worthwhile to spend about 5 man-months on the database development. This amounts to about AUD $ 20,000 in value. It is envisaged that phase 2 is going to span the next three years and the same database can be used throughout.

A bad approach

There was no 'One, two, three, and away', but they began running when they liked and left off when they liked, so that it was not easy to know when the race was over. (Lewis Carroll, Alice in Wonderland).

When Lewis Carroll described the Caucus Race in Alice in Wonderland, he meant to provide us with some entertainment in reading it and to stimulate some thoughts, I guess. I am sure he did not mean to provide us with a methodology to develop databases for scientific use.

In fact, it is a recipe for disaster if one uses this approach to manage any project. So, I have told you how not to manage a database development.

In order to illustrate the point, here I describe a hypothetical but common scenario. An ecologist is given half a million dollars to monitor an endangered desert fauna species over a number of years. It involves trapping, re-trapping, tagging and the recording of reproductive behaviour with a reasonably complicated data gathering process. Having been trained in a more traditional biological science school and having not yet been exposed to the power of modern database techniques, he goes on to set up a database by himself in dBase or Paradox. He does this in quite an *ad hoc* manner, creating tables as he goes. He knows almost nothing about relational database theory and the importance of Normalisation of the tables. That is, he is not making use of the preceding 15 years of advancement in information science

After a few years of data collection, he gradually realises that his database is very difficult to use. It takes a long time of thinking just to design a query to extract the desired information. After talking to a few experienced information scientists, he is convinced that after all, information science has become an established science in its own right, just like biology, chemistry and physics. So he appoints a keen young lady, who is an inexperienced contract programmer, to re-design his database. He tells her, "I still have a sum of money left in my budget. Please do whatever you can until the money runs out.". This is a remarkably common situation in which we scientists may find ourselves.

Half a year later, the young lady has spent all the allowed budget solely on writing a few hundred pages of codes. She tells the ecologist that she has put in a lot of work, and it certainly has been worth his money. Unfortunately, due to lack of experience on both sides, they have not realised that the amount of programming codes written is not necessarily a measure of the quality of the system.

The new database system is only partly debugged and is still not working. There is no budget left to write the User Documentation. By this time, the contract programmer has been offered a better job overseas under the recommendation of the ecologist and has to leave immediately. In short, the ecologist is left with the new system, which is practically unusable. Furthermore, there is no Programmer Documentation, making handing over to a succeeding programmer uneconomical and arduous. Thus the ecologist is back to square one and forced to stagger on with the old dreaded system. Time and money have been spent in vain!

The most immediate reason for such failure is that if the software development is brought to a halt at an unplanned time before completion, it will be at just such an arbitrary state. The ecologist may not even know whom he should blame.

It would be clear that the following methodology, which I advocate will overcome all the problems above.

The correct approach

Proper management techniques for software development have been widely used since the 1960's. Off the shelf modern methodology packages are available now (APT 1993). The principles are the same as for managing typical construction projects, like in engineering and architecture.

Historically, the method of project scheduling including a Critical Path Analysis (Taha 1992), was invented especially for making the first atomic bomb in the Manhattan Project, so that we might know when we could end the War. After changing human history, these techniques have now become standard tools in operations research and are not to be dismissed without due respect.

Therefore, to manage the development of a database or of any software, the manager should use proper project management techniques, which divide the project into a number of phases, namely: (1) Initiation, (2) Analysis, (3) Design, (4) Construction, (5) Implementation. We should not confuse these phases with that of the ecological project, which may have quite different phases. I give a brief description of these phases in the Appendix.

There are no hard and fast rules for dividing a project into phases. Different ways suit different types of projects. The principles are *control* and *communication*. The phases mentioned may suit most cases in an ecological research environment. The manager should not hesitate to depart from them as he sees fit.

Each phase should end with a documentation.

Prototyping should be used whenever possible. Here is an outline of prototyping. It is outlined as follows. Once the construction phase starts, a simple computer database should be constructed first, containing only the bare skeleton of the system. It may include the tables, with only a few typical Forms for data entry and updating, without any unnecessary details. This system is an early prototype of the proposed system. It should be used for demonstrating to the ecologist, the User. Only if he is satisfied, the programmer should proceed to the next stage of development. If he found some unexpected behaviour, they would have to correct it by back-tracking to the Design phase or even to the Analysis phase. Such prototyping detects misconceptions and communication failures early and is thus invaluable. Control and communication cannot be overemphasised in project management.

Discussion

The phased approach ensures that when the budget runs out, the user will at least have proper documentation. If time allowed, he might also have a partially working and usable system. The programmer's successor will be able to continue the development smoothly. Should the programmer suddenly resign, the software development should not have to start from scratch. It is worth noting that Programmer Documentation is even more important than the equivalent of a civil engineering construction project. This is because no two persons think alike; the variability in software design styles is greater. When the budget runs out, the last task done should be more likely the writing of a piece of documentation rather than a piece of code. Thus, effective management of software development does not necessarily lead to the completion of the development, but it can guarantee that all the resource put in is optimally spent.

In fact, the phased approach, although it may be rather informal, should be used, even if the biological scientist chooses to develop the software himself. This will stop him setting moving targets. The most effective way is for him to work in conjunction with an experienced information scientist, who does the software development management for him.

Surprisingly, some ecologists are still using the 'Lewis Carroll' approach to set up a database.

Appendix Phases in a software development project

I assume an information scientist, known as a Systems Analyst, Analyst for short, is managing the database development for the ecologist.

Phase 1 Initiation

This starts from the approval of the ecological project and the decision that a computer database is needed. At this stage the ecologist has already decided what data to collect. For an independent software development project, a phase called Conception comes before this phase. In ecological applications, it is already included in the design process of the ecological project and does not appear as a separate phase here.

The Analyst does a preliminary analysis of the ecologist's needs and does a feasibility study before he tentatively chooses a suit of hardware and software from the alternatives. Then he determines the approximate scope of the project and makes an estimation of the completion time, accurate up to only about 100 %.

The document produced from this phase can be called the System Proposal.

Phase 2 Analysis

The Analyst and the ecologist will analyse in detail all the User Requirements of the database, namely data entry, updating, data output and reports required.

It is best that they review the data collection process and field sheets at this point. They decide on data validations required and the system acceptance criteria. Security and disaster recovery standard are also determined. At this stage, completion time should be estimated again to about 50 % accuracy. This includes estimating how long the next two phases may take (design & construction).

Then they must come to agreement and produce the document for this phase, the Functional Specification. It need not be formal, but needs to be clear and written in User Language without information science jargon.

Phase 3 Design

The Analyst finalises the choice of hardware and software.

Based on the Functional Specification from the previous phase, he works out how to achieve all the User Requirements. He directs the database designer, who may be himself, to design the tables, which holds the data, so that they fulfil the requirements of the Relational Database Theory.

He may choose to test all the computer techniques required to make sure they work as he expected.

He should also revise the facility and personnel requirements and describe exactly how the operation of the system fits into the ecological project. Factors like computer literacy of the data entry staff are considered.

Then the Analyst and the Designer specify in terms of computer techniques how to achieve all the User Requirements in the document of this phase, the Design Specification.

Phase 4 Construction

The Analyst constructs the complete database system. As the computer techniques have been all tested above, there should not be major problems. Prototyping technique described in the text should be used through out this phase to maximise communication, until the acceptance criteria are fulfilled.

The User Guide and the Programmer Documentation should have been completed.

This phase ends when the ecologist writes a memo, the User Acceptance, to the Analyst saying that the system is satisfactory.

Phase 5 Implementation

The system goes into production. the Analyst may have to perform the data take-up or data conversion from an existing, possibly manual system. Any minor problems ironed out and corrected must also be documented.

Acknowledgment: I acknowledge N S Lander and A R Chapman for encouraging this work, Allan H Burbidge and Norm L McKenzie for helpful suggestions on the earlier draft and Stephanie H T Yung for expression editing.

References

Taha H A 1992 Operations Research, An Introduction, 5 th Ed., Macmillan.

APT Software Development Methodology (Jan 1993), Release 5.1, developed by Execom Ltd, Perth, W. A.

Yung F H, De Tores Paul & Halse S A 1997 Importance of relational database normalisation in ecological projects. Journal of Royal Society of Western Australia (to be submitted).