15

A rigorous framework for measuring development productivity and estimating the effort of multimedia courseware

I. M. Marshall
W. B. Samson
P. I. Dugard
A. Castell

University of Abertay Dundee
United Kingdom

ABSTRACT

Over the last few years impressive claims for increased productivity have been made for the development of multimedia courseware. Perhaps multimedia effort estimation has proved to be unreliable in accurately predicting development effort. The authors propose that the problem can be clarified by careful definition and rigorous use of terminology to measure productivity or estimate effort. Starting point is the organizational context and the end product complexity. A waterfall model for multimedia courseware development is used to define stating and finishing points for measurement purposes. On this basis elapsed time, development time, effort, learner time, productivity and effort to learner time ratio are defined along with appropriate units.

Main conference themes: costing, methodologies, quality
Educational areas:
Study topics:
Secondary keywords: authoring systems, computer assisted instruction, courseware, multimedia, software engineering

J. D. Tinsley et al. (eds.), *World Conference on Computers in Education VI*

INTRODUCTION

When a new multimedia courseware authoring tool or development system is announced it is normally associated with a claim for improved productivity. In less than a decade there has been an apparent eightfold improvement in productivity reported in the literature [1]. This improvement has been achieved while at the same time the visual quality and complexity of the end product has increased. Is this increase in productivity real, does it reflect variability in the product, the development process or is it a feature of inconsistent definitions applied to measure productivity?

Associated with these questions is the unreliability of results from existing multimedia courseware development effort estimation methods. Six methods for estimating the cost or development effort of multimedia courseware have been identified by Marshall, Samson and Dugard [2]. Jay [3] evaluating one of these methods found that it produced effort estimates within 10% of the actual development effort for only three of the 9 projects tested. The other six estimates ranged from 86% too low to 70% too high. Similar variability in results have been obtained for the other estimation methods against actual project data [4].

These problems could be clarified and perhaps resolved if a greater insistence was placed on the careful and accurate definition of terms and the rigorous application of this terminology to measuring and discussing both productivity and effort estimation. In this paper we will begin to define the terminology by adopting a top down approach to describe the process and products of multimedia courseware development.

DEFINING TERMINOLOGY

Organizational context

There are significant differences in the organizational context in which multimedia courseware is developed. In a commercial environment courseware is either developed for an external client or for the general training and educational market to make a profit. Noncommercial developers tend to create multimedia courseware for internal educational, training or research use without the primary profit motive. Within both environments the courseware can be developed by individual authors, teams or by consortiums. Figure 1 describes the two organizational contexts in which multimedia courseware is developed.

In a noncommercial environment an author may develop multimedia courseware in their own time for the author's own use. Noncommercial team and consortium projects tend to be based around internal or external funding

where a budget has been provided for a fixed period. One of the major differences between noncommercial and commercial environments is that the desire to maximize profit is significantly reduced. A commercial developer will try to minimize development and maintenance effort to maximize profit. It is this difference when combining results from noncommercial and commercial developments, which is likely to produce unreliable measures of productivity or effort estimation.

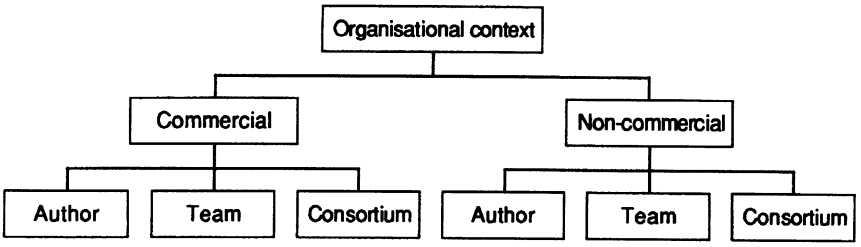

Fig. 1 Organizational contexts of multimedia courseware development

Courseware product quality

There are wide variations in the quality of multimedia courseware produced in both organizational contexts. A low quality product could be produced by a noncommercial author entering simple text and graphics into a hypertext system for classroom use. In this situation less than perfect standards of presentation, robustness and presence of bugs may be acceptable in the finished product. A commercial developer producing the same courseware would have to develop a higher quality product which is well presented, robust, and designed and tested to ensure it requires little ongoing maintenance.

It is essential to define consistent quality standards to describe multimedia courseware. To this end we define that multimedia courseware should involve computer control and presentation of text, or graphics, or video, or audio or animation for an educational or training purpose. The end product developed should include all of the following elements:

- software which controls the learning environment;
- computer based presentation of one or more media;
- learner and/or computer based assessment of the learning process;
- learning support materials;
- installation materials and software;
- learner record system.

These end products should be produced in an acceptable commercial quality and be in the form of camera ready or master copy quality. This means that productivity gains cannot be claimed for the production of a noncommercial prototype. Using this end product quality definition each multimedia courseware effort model developed would be primarily designed for commercial use. The model could be applied by noncommercial developers provided appropriate allowances were made for the differences in end product.

Development lifecycle

Having defined the quality of the end product it is important to agree the development lifecycle which produces the multimedia courseware. In their analysis of existing effort estimation methods Marshall, Samson and Dugard [2] found that even when the methods defined a development lifecycle, these used different starting and finishing points. For example Schooley [5] specifically excludes training needs analysis which Bergman and Moore [6] specifically included.

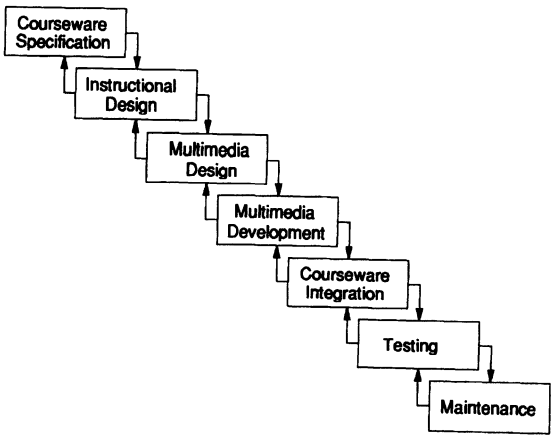

Fig. 2 The waterfall model of multimedia courseware development

Figure 2 shows a waterfall lifecycle model for multimedia courseware development. This is based on a number of existing Instructional Design and Courseware Engineering models [7]. Although highly stylized and perhaps de-emphasizing the concurrent and iterative nature of multimedia courseware development it is proposed as a basis for future discussion on what is included in a multimedia development.

In the courseware specification phase the detailed objectives and outline content of the course are defined along with target audience and additional information about the course to be delivered. Following this phase the overall instructional design is developed based on discussions with subject matter experts, courseware designers and the client. Once the output of this phase has been agreed with the client, then detailed design is undertaken for the media to be used in the final course. Additional experts such as media specialists, graphic artists, design editors and programmers may be added to the team at this point.

In the multimedia development phase the graphics, audio, video, sound and program structure is developed. The courseware integration phase brings the various multimedia elements together on one platform. The testing phase involves pilot testing of the courseware with learners. Problems identified by the evaluation of the courseware during the testing phase are fed back into the model to improve the quality of the final product. Once testing is complete the product is amended to solve any problems and it is then ready to be published. The final phase is the maintenance phase in which the courseware is only altered when serious problems are detected or the client requires changes to update the content and method.

For the purposes of measurement we use this development lifecycle to define the starting point at the beginning of the instructional design phase and the finish point at the completion of the testing phase. The reason for excluding the courseware specification phase from the agreed development lifecycle is that in a commercial development the duration of this phase can vary considerably between projects. The client and developer are typically involved in negotiations as to what can be developed. This can be protracted or instantaneous depending on the difference between the client's expectation and the developers' cost base. In addition these negotiations may be multilevel and complex because of the innovative nature of multimedia development. By the end of the courseware specification phase the developer and client will have agreed the major objectives, content, delivery platform, method and media for the project. Similarly the duration of the maintenance phase is difficult to define accurately. Estimation of maintenance effort for multimedia courseware is a separate issue which may be worthy of further research.

Elapsed time

Having defined the development lifecycle and identified precise start and finish points elapsed time can be defined. Elapsed time, as the name suggests, measures the total time in days from the start of the instructional design phase to the end of the testing phase.

Development time

Development time is the basic measure of the number of hours taken to develop the multimedia courseware from the start of the instructional development phase to the end of the testing phase. During this period development time is measured in productive working hours consumed by the multimedia courseware project excluding weekends, holidays and lunch breaks. Development can be related to development time by the following formula.

Development time = Working days per day x (Elapsed time - (Nonworking days))

Figure 3 shows the development time in terms of the number of developers required during the project.

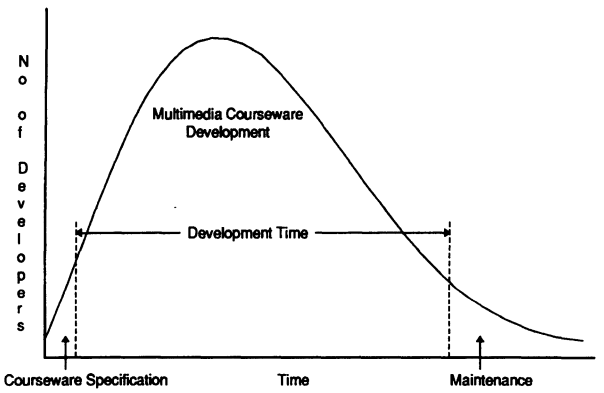

Fig. 3 Multimedia development lifecycle and development time

Agreeing the start and end points in a development lifecycle is important because excluding phases has a significant impact on development time of multimedia courseware. Table 1 indicates the effect of excluding lifecycle phases on development time.

The percentage allocated to each phase of the development lifecycle is based on figures quoted by Imke [8] for courseware development. The hours quoted for each phase are based on these percentages for a nominal 300 hour development. Project A describes a project in which all phases in the development lifecycle excluding the maintenance are recorded. In Project B the hours for Courseware Specification and Testing are excluded. This reduces the recorded development hours to 89% of the original Project A. In Project C the time allocated for the Instructional Design phase is also removed. The effect is

to reduce development time to 55% of the original Project A. Again, by not recording all phases in the development lifecycle it is possible to reduce the apparent development time without any real improvement in productivity.

Table 2 Effect of excluding lifecycle phases on development time

Phase	*%*	*Development Time*		
		Project A (hours)	*Project B (hours)*	*Project C (hours)*
Courseware Specification	7	21		
Instructional Design	34	102	102	
Multimedia Design	25	75	75	75
Multimedia Development	22	66	66	66
Courseware Integration	8	24	24	24
Testing	4	12		
Total Hours	100	300	267	165
	% of Total	100%	89%	55%

Effort

In the literature effort is seldom used as a measure in multimedia courseware effort estimating, although it is the normal measure in software engineering. Effort is measured in developer hours and provides a measure of the total amount of productive work expended in developing the course.

$$\text{Effort} = \sum_{\text{Instructional design}}^{\text{Testing}} \text{Development time on phase x Number of developers used}$$

Cost

Cost describes in monetary terms the effort of producing multimedia courseware.

Learner time

Existing courseware effort estimation methods all use the time a learner spends to complete the course as the main cost driver. In the literature the amount of time spent by the learner using the courseware is normally called delivery time; to prevent confusion with development time we propose to describe this measure as learner time. Unfortunately there is no current definition as to how it should be calculated. Some developers base their estimates on how long they feel an average student will take to complete the course. Other developers define it, once the product has been developed, by testing on an average student. Yet other developers suggest using the average learner time of a

sample group of learners. Given that one of the strengths of multimedia and computer based courseware is the selfpacing of instruction the range of learner time for a group of learners can vary considerably.

Jay, Bernstein and Gunderson [9] investigated the use of other base measures for courseware development, but found similar problems with other potential measures such as screen interactions and frames. While learner time has problems, it is widely used and accepted by the industry. To reduce these problems it is essential to define a standard method for measuring learner time. Prior to the development it should be based on the time an average learner will be expected by the developer to complete the course. Once the courseware has been developed learner time should be measured using the mean time taken for a representative group of 20 learners for whom the material was developed to complete the course.

Productivity

Symons [10] defined a range of productivity measures for software size estimation. The basic measure of productivity is given by dividing the output by the input. The output measure is the number of hours of learner material delivered and the input is the development effort. Productivity is thus measured by the following equation.

Productivity = Learner time / Effort

Productivity is measured in units of 'average learner hours per developer hour'. With current development methods productivity will normally result in a value of less than one.

Effort to learner time ratios

We are proposing to rename the widely used 'development to delivery time ratio' as 'effort to learner time ratio'. The original definitions for development to delivery time ratios were described in terms of 'average man-hours expended per CBI lesson hour' [11]. Recent authors have dropped the effort element in favour of the less descriptive 'hours of development for each hour of instruction' [1]. To reduce ambiguity, the following definition should be used.

Effort to learner time ration = Effort / Learner hours

Using this definition, the ratio should be measured in 'average developer hours per learner hour'. Inverting the equation for effort to learner time ratio produces the equation for productivity.

Effort to learner time ration = 1 / Productivity

Table 2 gives examples of the relationship between productivity and effort to learner time ratios.

Table 2 Productivity and Effort to learner time ratios

Average delivery (developer hours)	*Development effort time (hours)*	*Productivity (average learner hours per developer hour)*	*Effort to learner time ratio (average developer hours per learner hour)*
0.5	100	0.005	200:1
1	100	0.010	100:1
2	100	0.020	50:1
20	100	0.200	5:1
100	100	1.000	1:1

The effort to learner time ratios are perhaps easier to remember than the proposed productivity equivalent. While it is likely to remain popular due to its historical significance and acceptability to industry, we would encourage the adoption of the proposed productivity measure to ensure compatibility with software engineering metrics.

Whichever of the two measures is used it is important to employ both a consistent definition and means of measuring learner time. Apparent improvements in productivity can be made by using a pessimistic value of learner time. Table 3 gives an indication of the effect on the effort to learner ratio of using optimistic, actual and pessimistic values for learner time.

Table 3 Effect of learner time on development to learner time ratio

	Effort to learner time ratio		
Effort (Developer hours)	*Calculated using an optimistic value of Learner Time (1 hour)*	*Calculated using the actual value of Learner Time (3 hours)*	*Calculated using an pessimistic value of Learner Time (5 hours)*
100	100:1	33:1	20:1
200	200:1	66:1	40:1
300	300:1	100:1	60:1
400	400:1	133:1	80:1

This table reinforces the need to ensure that great care must be taken in measuring learner time to prevent inaccurate claims for productivity improvement.

CONCLUSION

Formally defining the terminology used to describe multimedia courseware development allows more objective comparisons to be made of productivity and forms a more predictive basis for multimedia courseware development effort modelling. Defining the organizational context and end product complexity of multimedia courseware is the beginning for establishing standards for measurement. The waterfall model for multimedia courseware development, while highly stylized, enables the stating and finishing points for productivity to be defined and allows effort estimation measurement purposes to be made more transparent. The waterfall model also supports a consistent definition of elapsed time, development time, effort, learner time, productivity and effort to learner time ratio. Based on these definitions of multimedia courseware productivity data for effort estimation can be objectively measured.

REFERENCES

1. Orey, M., *et al* (1994) *Development efficiency and effectiveness of alternative platforms for intelligent tutoring for the mobile subscriber radio-telephone terminal.* Computers and Education, **22** (4) pp. 301-313.

2. Marshall, I. M., Samson, W. B. and Dugard, P. I. (1994) *Courseware—How much will it cost to develop?* in 2nd All Ireland Conference on the Teaching of Computing. Dublin, Ireland.

3. Jay, J. (1988) The CBT analyst. Journal of Computer-Based Instruction, **15** (1) pp. 34-45.

4. Marshall, I. M., Samson, W. B. and Dugard, P. I. (1994) *Multimedia courseware. Never mind the quality, how much will it cost to develop?* in Proceedings of Association for Learning Technology 94. University of Hull, Hull.

5. Schooley, R. E. (1988) *Computer-based training (CBT) cost estimating algorithm for courseware (CEAC),* in Interservice Industry Training Systems Conference, Florida.

6. Bergman, R. E. and Moore, T.V. (1990) *Managing interactive video/multimedia projects.* Educational Technology Publications, Englewood Cliffs, NJ.

7. Tennyson, R. D. and Elmore, R. L. (1993) *Integrated courseware engineering system,* in NATO ASI on Automating Instructional Design, Development & Delivery. Springer-Verlag, Berlin.

8. Imke, S. (1991) *Interactive video management and production.* Educational Technology Publications, Englewood Cliff, NJ.

9. Jay, J., Bernstein, K. and Gunderson, S. (1987) *Estimating computer-based training development times.* Research Institute for Behavioural and Social Sciences, Alexandria, VA.

10. Symons, C. R. (1993) *Software sizing and estimating: Mk II FPA (function point analysis).* John Wiley & Sons, Chichester.

11. Hurlock, R. E. and Slough, D. A. (1976) *Experimental evalaution of PLATO IV technology: Final report.* Navy Personnel Research and Development Center, San Diago, CA.