Process Management for Geographical Information Systems Development

Stephen G. MacDonell[1] and George L. Benwell

Computer and Information Science

University of Otago

Abstract

The controlled management of software processes, an area of ongoing research in the business systems domain, is equally important in the development of geographical information systems (GIS). Appropriate software processes must be defined, used and managed in order to ensure that, as much as possible, systems are developed to quality standards on time and within budget. However, specific characteristics of geographical information systems, in terms of their inherent need for graphical output, render some process management tools and techniques less appropriate. This paper examines process management activities that are applicable to GIS, and suggests that it may be possible to extend such developments into the visual programming domain. A case study concerned with development effort estimation is presented as a precursor to a discussion of the implications of system requirements for significant graphical output.

[1] Address correspondence to Dr Stephen MacDonell, Department of Information Science, University of Otago, P.O. Box 56, Dunedin, New Zealand. Fax: +64 3 479 8311 Email: stevemac@commerce.otago.ac.nz

Software Processes for Geographical Information Systems

A software process is a continuous set of activities oriented towards software production and evolution, from the identification of an initial concept through the analysis and definition of requirements, design of software and other components, coding, testing, implementation, system operation, maintenance and enhancement, to eventual system retirement [19]. Although traditionally this process was depicted as a clearly delineated sequence of distinct non-overlapping phases (for example, see [6]), more realistic models have since appeared. These process models tend to show the iterative overlapping nature of activities, events, transitions, decisions and states involved in the development of various software products [7,12].

A development methodology for GIS is required for the efficient development of such systems. It is not apparent in the literature, or in practice, that such methodologies have been readily adopted. This is despite the extensive application of methodological approaches commonly used in informatics. The advances in GIS methodologies have been in realising that such systems, at least in one sense, are atypical; they are one of the few management information systems that consolidate data originating from a number of organisations at different levels of administration [23]. Love [13] advocated that a sound methodology was necessary and emphasised the importance of technical and humanistic perspectives. There remains a problem, however, and that is that while such methodologies may indicate the steps or phases of work to be done, they do not directly measure or indicate the amount of effort required.

Management of the Software Process

Participants (or ‘stakeholders’) in the software process have differing points of focus: customers see the product as vital, as they want a system that will enable them to work more effectively; developers consider the process more important, as they want to work in the most cost-effective manner. In order to direct resources to both perspectives, contemporary process management has adopted the view that a quality software process will produce a quality software product.

Development project failure is a direct result of ineffective process management. Some of the symptoms of ineffective management are:

- insufficient communication and reporting between users and developers
- differing expectations of system functionality
- development in isolation of organisational aims and objectives
- failure to assess risks, costs and benefits
- poor estimation of effort and duration
- development without adequate change procedures
- short cuts in development taken as a result of schedule constraints
- poor coordination among developers.

Techniques and methodologies have been proposed as (partial) solutions to some of these problems. For example, the issues of insufficient communication and differing stakeholder expectations may be overcome through the use of a prototyping process that enhances communication. Similarly, effective software configuration management [18] provides appropriate change procedures for use during systems development and evolution. These solutions are as applicable to the development of GIS as to any other system type.

Another of the symptoms of project failure listed above is that of poor estimation of effort and duration. This issue has been investigated in detail with particular reference to GIS.

Case Study - Effort Estimation

Identification and measurement of the determinants of software development effort are clearly desirable if the process is to be effectively managed (as opposed to simply being observed). A variety of methods exist for this activity within the business systems domain, the most widely used being function point analysis (FPA) [1,20]. Use of the Mark II FPA approach [20] produces as its output a number of function points - a measure of system functionality - based on two components: a measure of information processing *size* as derived from a weighted sum of the inputs, processing and outputs of each system transaction; and an adjustment factor derived from the *complexity* of the software and the environment in which it will operate.

Within the business systems domain, the industry standard calculation of the information processing size (generally based on transactions manipulating attributes in data models) in unadjusted function points (UFP) is:

$$\text{Size in UFP} = 0.58{*}N_I + 1.66{*}N_E + 0.26{*}N_O$$

where N_I is the number of input data elements used in system transactions

N_E is the number of data entity references made by system transactions

N_O is the number of output data elements produced by system transactions

the weightings 0.58, 1.66 and 0.26 have been derived from nearly 100 business systems.

This equation therefore accounts for the *relative* contributions to development effort associated with building software components to control the formatting and validation of input and output data items, and the database-centred processing required to transform inputs into outputs.

The technical complexity adjustment (TCA) factor is a weighted function of the sum of twenty characteristics that assess the impact of such requirements as distributed processing, user training and system portability to overall system complexity (see [20] for a fuller discussion of the TCA formulation).

Overall adjusted system functionality in Mark II function points is therefore:

Functionality = UFP * TCA

When development effort data is collected as part of a managed software process, those responsible for the control of software development can begin to use FPA as an effort estimation method for future projects. This is based on the calculation of productivity figures from recent typical projects, resulting in an average productivity rate for the organisation (in function points per effort unit). The determination of an average productivity figure such as this then enables effort forecasting. When a specification for a new system is produced, its scope can be calculated in terms of function points. Applying the average productivity rate to this scope results in a prediction of overall development effort.

In order to illustrate the worth of such an approach to the management of GIS development, a case study was conducted over a period of two years. This study enabled a retrospective effort estimate to be derived for the development of a prototype Hazards Register information system [2] required by a local city council. The system was to support local authority activities associated with the granting of building permits and resource consents. System scope was determined based on the original requirements specification documents ($N_I = 71$, $N_E = 15$ and $N_O = 40$) and an effort estimate was determined using an industry-standard productivity rate of 0.1FP/person-hour [20]. The estimate was found to be approximately 15% under the actual expended effort figure. Although a lower difference would clearly be desirable, this degree of difference as a first-cut prediction is reasonably accurate, particularly when a limit of 30% has been suggested as adequate for estimation in the business systems environment [22]. More importantly, the FPA-based estimate compared very much more favourably that the original estimate guessed at by the developers. Based on the results of this work, and on the lessons learned from it (see the following sub-section), further case studies are about to begin.

Lessons Learned

Although the case study effectively demonstrated the potential of a process management technique such as FPA in the GIS domain, it also highlighted some specific differences between business systems and GIS that have implications for process management in this environment. In particular, the inappropriateness of the relative weightings associated with the components of *business* system transactions became apparent.

The component weightings are normally calibrated to each development environment, so as to ensure that scope assessment and subsequent effort predictions are relevant to each environment. The case study briefly described above, however, had no historical data available to enable calibration and consequently the business-oriented standard weightings were initially adopted. Clearly, however, relevant figures obtained from GIS development could be expected to be more useful. Two approaches were therefore made (several months apart) to the recipients of the GIS-L listserver, the international mail server for those interested in the use and development of GIS, in an attempt to obtain actual project data from GIS-based system developers. Disappointingly but not unexpectedly, these requests produced a nil response. This may be simply a reflection of the reluctance of developers to provide systems for analysis. It is equally as likely, however, that developers simply did not have the data requested, particularly in terms of effort records, as this unfortunately remains a common situation in the systems development domain.

A different approach was then tried. A new request to GIS-L asked developers to rate the difficulty associated with code design and development for the three components of transactions in a GIS environment. This request produced twenty-seven responses, with overall weightings proving to be quite different from those obtained in the business systems domain (see Table 1).

Table 1. Differences in component weightings for business and spatial systems

	Input	Process	Output
Weighting - Business	0.58	1.66	0.26
Weighting - Spatial	0.81	0.84	0.85

Although the weightings are based on anecdotal rather than empirical evidence, it is considered that they are still likely to be more appropriate than the business system weightings, particularly as the differences are quite significant. In the calculation of system scope for the case study described above, then, it was the set of spatial weightings that was used. Comments that accompanied developer ratings reiterated these results - many respondents further suggested that the values for output would be substantially different depending on whether output was to be textual or graphical.

As a result of the above research it has been concluded that there is insufficient data available on GIS effort and factors that influence its determination. There is a lack of data models, system design documentation, data upon which to determine, adjust or modify the weights, nor has there been adequate investigation of other factors that may influence GIS measurement. It may be important to understand such variables as map components, number of layers, colour options, and graphical versus textual elements. These issues will require attention if effort estimation figures are to achieve any level of acceptance in GIS circles.

Current Research

Some of the conclusions reached as a result of the case study described above have wider implications. In particular, requirements for graphical output from any system type (that is, not only GIS) may demand reformulation of techniques for effort estimation.

Wider Implications - Effort Metrics for Graphical Output Software Development

With the advancement of graphical capabilities in terms of information presented on screen or paper, computer systems are able to provide extensive graphical information of value to users. For example, business data on market shares can be presented in a pie chart, scientific time series trends on tidal variations can be illustrated in line plots, real-time changes in river levels can be shown as screen-based scale models, and population densities can be depicted in a choropleth map. Multimedia systems, by their very nature, make extensive use of graphical components such as images, animations and video clips. The question of interest here, however, is what does the picture cost? That is, is it possible to measure, calibrate and predict the effort required of developer(s) to create the software that provides this graphic-based output? A related question is: is the effort devoted to the development of software to control output different than that required for the software that handles input and processing?

Work oriented towards effort assessment specifically oriented towards software for graphical output is not extensive, for a number of reasons:

- in the past the only option for output (particularly screen-based) was in the form of text, as hardware limitations restricted the use of graphical output
- software metrics have tended to give little or no consideration to the *form* of the output, as the *quantity* of information presented was considered to be more important (e.g. see [20])
- the rise of spatial information systems and multimedia systems, with their inherent dependence on graphical output, has been a relatively recent phenomenon.

Thus assessment of development costs from the perspective of output has been concentrated on the *amount* of data reported. This has been quantified under some counting schemes as measures of the number of data items reported per screen, or the number of fields and headings per report. Measures of this type are certainly acknowledged as important. It is suggested here, however, that the *form* of reporting is likely to have a further impact on both the cost of development and the 'value' of the output to the recipient. Some work investigating the value aspect has been undertaken, with several studies dealing specifically with comparisons of text-based and graphically-based representations [4]. Related areas of research are also of relevance, particularly those examining map complexity [14,15], screen-based widget layout [17,21] and usability analysis [5]. As far as can be determined, however, empirical effort or cost determinations and comparisons for such environments are yet to be examined. Given the increasing use of graphically-oriented software systems it is felt that a significant research opportunity exists. As the next step in this research, variations on the FPA approach to provide greater relevance to geographical systems is now discussed.

Component Weightings and Data Collection

Given the findings of the case study it would appear that the component weightings adopted for the calculation of business system scope in Mark II FPA are indeed inappropriate for geographical information systems. The anecdotal evidence indicates that the relative contributions of input, processing and output are much closer than is the case for business systems and that, in general, the development of code to control input and output in spatial systems is more difficult than for business systems. In fact, the dominance of input and output is made even more significant by the fact that the actual values of N_I and N_O are normally much higher than that for N_E. This is due to the fact that the input and output measures are concerned with numbers of data items manipulated, whereas the processing measure considers entity accesses. In most management information and decision support systems, within which class business and geographical information systems fall, the number of data items used in and produced by the system, in terms of screen-based data entry and soft- and hard-copy reporting, is greater than the number of entity accesses required in processing. If the case study figures are adopted an illustrative comparison of the contribution to scope of each component type can be performed, as follows:

Business weightings:		Spatial weightings:	
Input -	0.58*71 = 41 (54%)	Input -	0.81*71 = 57 (55%)
Processing -	1.66*15 = 25 (33%)	Processing -	0.84*15 = 13 (13%)
Output -	0.26*40 = 10 (13%)	Output -	0.85*40 = 34 (32%)

In this case, the absolute contributions to system scope (and ultimately to development effort) of software development for processing and output are virtually reversed, indicating that the development of software for output control is particularly more difficult for spatial systems whilst the development of processing code may be less influential.

In order to more accurately determine appropriate weightings data *must* be collected as a matter of course within a process management framework. This requires a level of process maturity currently beyond some organisations, but is acknowledged as one of the necessary steps if software processes are to be improved [3]. There exists a cycle of avoidance that must be overcome if appropriate weightings, and consequently more accurate measures of scope and effort, are to be obtained. This cycle has at one point developers and managers who will not use measures and models (such as FPA) as they are not convinced of their worth. If the measures and models are not used, then there is no point in collecting the relevant data. If the data is not collected, the worth of the measures and models cannot be assessed. Clearly, then, goal-based data collection must be undertaken as a first step to enable more effective process management.

Input Assessment

The impact on software development effort specifically related to code built for handling input processing in spatial systems is likely to be similar to that applicable to business systems. Query handling, data entry processing and associated validation procedures are generally required irrespective of system type. It may be that some adjustment is needed to cope with the input and maintenance of spatially referenced data, particularly where this requires input via digitising. However, the process of data collection/acquisition in the spatial systems domain is a distinct area of research and practice and is therefore not expressly considered here. Attention in this work is directed to the querying and manipulation of existing data to obtain textual and/or graphical output.

Processing Assessment

The processing of data from a database involves the programs that extract, manipulate and deposit data from and in a database. In the case of a GIS it can be assumed that there are three basic forms for a data repository. First, is that which supports a raster format where the data will be stored in a tree structure or even in very flat ASCII files. Second, is the standard relational model (here implied to include object oriented paradigms) which would include most contemporary systems. Third, is the advanced relational models and the emerging 'spatial database structures' such a Oracle7 MultiDimension and developments in spatial database engines soon to appear from the leading vendors. When these structures are compared to those used in business applications there is little to no difference. All three structures above may be used in a purely business environment (that is, one which is less graphically oriented). It is therefore not considered that significant changes will be required in the processing component of the assessment and estimation process as described previously. Notwithstanding this conclusion, there are two points that need further discussion. First is the relative difficulties of using raster or vector representations of reality and second, is the advances being made in spatial data modelling.

The choice of a raster or vector structure will affect both the effort required to create a system and the fidelity of its representation of reality. If could be argued that a raster-based system will require far less effort but will not represent reality as accurately as a vector system. This argument starts to diverge from the theme at hand. But, while that is true, the estimation formula is not about accuracy, it is about effort. If the raster form takes less effort then so be it. The formula will reflect the lower effort, the user must understand and justify the fidelity.

Advances in spatial data modelling will impact on the determination of processing scope (as measured in FPA with N_E). Research by Firns [11] and Feuchtwanger [10] have laid the foundations for new data modelling formalisms. This will bring about some degree of divergence from 'the standard' models used in business [9]. This will do no more than create a modified 'spatial' version of the UFP formula which, after all, was the intention. If there is a requirement to convert between formula models then additional research will be required. Presently, there seems no reason or justification for such conversions.

Output Assessment

The significant difference between a 'traditional' system and a GIS is the form of the output. Regardless of whether the GIS is raster or vector, a significant amount of output will be graphical. While this is stated to be a significant difference it is acknowledged that this difference will diminish as programming and/or display concepts generally become more graphical. This trend is now observable and rapidly expanding with tools such as visual programming, multimedia and the world wide web. It is fair to say the amount of graphical programming and display will increase. It is with this background that the output component of the estimation formula is examined in the context of GIS.

It is true to say that GIS output is mostly graphical, though textual output is also important. After all, some maps may be no more than appropriate placement of textual attributes on a screen. In order to understand the effort required to produce these graphical outputs it is considered appropriate to examine the literature relating to map complexity [8,15,16]. Map complexity, according to Monmonier, relates to the number of map elements which result in a pattern that 'appears to be intricate or involved'. In the context of this paper that definition is inappropriate. For example, given the same data (that is the same entities and the same database retrieval) there are any number of different visual complexities. The thrust of this paper is more related to the complexity as it relates to the number of map elements and the effort required to place them on a screen or hard copy output. The ‘map complexity’ is therefore a function of entities and effort of placement, not intricacy or difficulty in understanding. These are certainly very important factors to be considered in the design of a GIS, but here it has little direct impact on effort. Nonetheless, there will be, intuitively at least, some degree of positive correlation between an intricate map and large effort. (excluding the novice cartographer who takes no time at all to create the most horrid and intricate map).

It is suggested that, when effort is involved, the following will be the important factors to be considered in the measurement of graphical output from a GIS:

- *the number of discrete graphical entities*; this is more concerned with measuring the number of separate entities such as height, hydrology etc rather than the number of contours or the number of rivers. In addition it is not concerned with the number of data points per contour (or river, as examples).
- *the style of the output*; is it a line drawing, a filled polygonal rendition or a solid model.
- *the nature of the output;* is the output static in place or position and is it a still or video.
- *the cartographic completeness of the map;* is the output crude or is it of high cartographic quality.

Any effort estimation formula will need to consider the influence of these and perhaps other spatial concepts at the output stage.

Conclusions

Research to date in process management for GIS suggests that, whilst similar software processes or even sub-processes may be required for business and spatial systems, there are some characteristics particular to GIS that make current techniques for effort estimation (and perhaps other process management activities) inappropriate. The results obtained from the case study highlighted the distinction between spatial and business systems, particularly in terms of the impact on effort associated with the development of code required to control system output.

Although some progress has been made towards determining the causes of or reasons for such distinctions it is clear that further research is needed in order to more fully ascertain the determinants of development effort in the GIS domain. Moreover, it is suggested that this research may become more widely applicable, given the ongoing move to graphically-oriented systems such as multimedia systems and visual programming environments.

References

1 Albrecht, A.J. Measuring application development productivity. In *Proceedings Joint SHARE/GUIDE Application Development Symposium*, 1979 pp. 83-92.

2 Aldridge, C., Benwell, G., Turnbull, I., Henderson, J., Harris, M. and Tay, A. Dunedin pilot hazards information system - a system analysis and proposal. In *Proceedings of the Fifth Annual Colloquium of the Spatial Information Research Centre*, University of Otago, Dunedin, 1993 pp. 247-264.

3 Arthur, J.D., Nance, R.E. and Balci, O. Establishing software development process control: technical objectives, operational requirements, and the foundational framework. *Journal of Systems and Software 22* (1993), 117-128.

4 Benwell, G.L. and MacDonell, S.G. Assessing the graphical and algorithmic structure of hierarchical coloured Petri net models. *Australian Journal of Information Systems 2,* 1 (1994), 17-28.

5 Bias, R.G. and Mayhew, D.J., Eds. *Cost-Justifying Usability*. Academic Press, Boston, 1994.

6 Boehm, B.W. *Software Engineering Economics*. Prentice-Hall, New York, 1981.

7 Boehm, B.W. *Software Risk Management*. IEEE Computer Society Press, Los Alamitos CA, 1989.

8 Bregt, A.K. and Wopereis, M.C.S. Comparison of complexity measures for choropleth maps. *The Cartographic Journal* 27 (1990), 85-91.

9 Date, C.J. *An Introduction to Database Systems* Vol. 1, 5th Ed, Addison-Wesley, Reading MA, 1990.

10 Feuchtwanger, M. Towards a geographic semantic data model, PhD thesis, Simon Fraser University, Canada, 1993.

11 Firns, P.G. An extended entity relationship model applicable to the design of spatially referenced databases PhD thesis, University of Otago, New Zealand, 1994.

12 Lehman, M.M. Uncertainty in computer application and its control through the engineering of software. *Journal of Software Maintenance 1*, 1 (1989), 3-28.

13 Love, W.R. GIS design and implementation: a successful methodology. In *Proceedings of the 19th Australian Conference on Urban and Regional Information Systems*, Wellington, New Zealand, 1991 pp. 474-484.

14 MacEachren, A.M. Map complexity: comparison and measurement. *The American Cartographer 9*, 1 (1982), 31-46.

15 Mersey, J.E. *Colour and Thematic Map Design: The role of colour scheme and map complexity in choropleth map communication*. Cartographica Monograph 41, Toronto, 1990.

16 Monmonier, M.S. Raster-mode area generalisation for land use and land cover maps. *Cartographica 20* (1983), 65-91.

17 Protsko, L.B., Sorenson, P.G., Tremblay, J.P. and Schaefer, D.A. Towards the automatic generation of software diagrams. *IEEE Transactions on Software Engineering 17*, 1 (Jan. 1991), 10-21.

18 Ratcliff, B. *Software Engineering: Principles and Methods*. Blackwell, Oxford, 1987.

19 Sallis, P., Tate, G. and MacDonell, S. *Software Engineering: Practice, Management, Improvement*. Addison-Wesley, Sydney, 1995.

20 Symons, C.R. *Software Sizing and Estimating: Mk II FPA (Function Point Analysis)*. John Wiley & Sons, Chichester, 1991.

21 Tan, K.P., Chua, T.S. and Lee, P.T. AUTO-DFD: An intelligent data flow processor. *The Computer Journal 32*, 3 (1989), 194-201.

22 Tate, G. and Verner, J. Software costing in practice. In Veryard, R. *Information and Software Economics.* Butterworth Scientific, UK, 1990.

23 Zwart, P.R. The rise and decline of land information systems. In *Proceedings of the 12th International Cartographic Association Conference*, Perth, Western Australia, 1984 pp. 123-133.