

**South African  
Computer  
Journal  
Number 11  
May 1994**

**Suid-Afrikaanse  
Rekenaar-  
tydskrif  
Nommer 11  
Mei 1994**

**Computer Science  
and  
Information Systems**

**Rekenaarwetenskap  
en  
Inligtingstelsels**

**The South African  
Computer Journal**

*An official publication of the Computer Society  
of South Africa and the South African Institute of  
Computer Scientists*

**Die Suid-Afrikaanse  
Rekenaartydskrif**

*'n Amptelike publikasie van die Rekenaarvereniging  
van Suid-Afrika en die Suid-Afrikaanse Instituut  
vir Rekenaarwetenskaplikes*

---

**Editor**

Professor Derrick G Kourie  
Department of Computer Science  
University of Pretoria  
Hatfield 0083  
Email: dkourie@dos-lan.cs.up.ac.za

**Subeditor: Information Systems**

Prof John Shochot  
University of the Witwatersrand  
Private Bag 3  
WITS 2050  
Email: 035ebrs@witsvma.wits.ac.za

**Production Editor**

Dr Riël Smit  
Mosaic Software (Pty) Ltd  
P.O.Box 16650  
Vlaeberg 8018  
Email: gds@cs.uct.ac.za

---

**Editorial Board**

Professor Gerhard Barth  
Director: German AI Research Institute

Professor Pieter Kritzinger  
University of Cape Town

Professor Judy Bishop  
University of Pretoria

Professor Fred H Lochovsky  
University of Science and Technology, Kowloon

Professor Donald D Cowan  
University of Waterloo

Professor Stephen R Schach  
Vanderbilt University

Professor Jürg Gutknecht  
ETH, Zürich

Professor Basie von Solms  
Rand Afrikaanse Universiteit

---

**Subscriptions**

	Annual	Single copy
Southern Africa:	R45,00	R15,00
Elsewhere:	\$45,00	\$15,00

to be sent to:

*Computer Society of South Africa  
Box 1714 Halfway House 1685*

# Effect of System and Team Size on 4GL Software Development Productivity

G R Finnie

G E Wittig

Computer Science Department, University of Natal, PO Box 375, Pietermaritzburg 3200, South Africa

## Abstract

*Modelling the effect of size on software development issues has been the object of considerable research in the software engineering community. The size metrics form an integral part of software project cost estimation models. The effects of the interaction between software size, average team size, total development effort, and elapsed development time are examined and modelled. The research included the capture and analysis of empirical data from 15 commercial fourth generation language data processing developments. This paper reports on the results of this study and the equations which were developed to model the effect of average development team size on productivity and total development effort. In addition, models relating system size and development effort are discussed.*

**Keywords:** *Development effort, development team size, software development productivity, system size, total elapsed development time*

**Computing Review Categories:** *D.2.9, K.6.1, K.6.3*

Received: February 1992, Accepted: October 1993, Final version: March 1994.

## 1 Introduction

Effective software development requires that attention be given to the interrelationships and the trade-offs between the size of the system being developed, the average development team size, and the required schedule or elapsed time. A team size which results in the highest productivity may not be the optimum for some developments as it is necessary to determine the value of the system to the organisation, so that developers can estimate the benefit of implementing this "as soon as possible." Research results cited by Conte et al [4], and Jones [7] indicate that as the team size is increased to reduce the schedule, development productivity drops and production cost increases. Productivity research done by Jeffery [5] corroborates these results.

Research results reflecting the effect of system size on development effort are contradictory [4, 7, 8, 12]). CO-COMO [2] and numerous other models which were developed in 3GL environments predict a power functional relationship. The increasing non-linear growth in estimated development effort as size increases, implies that there are reducing returns to scale. Other models, for example the Walston-Felix [4], reflect increasing returns to scale and predict that development effort will increase proportionately less than the increased size of the system. Albrecht's Function Point Analysis model [1] assumes a linear increase in development effort as the size of systems increases. Jeffery [5] on the other hand states that initially increasing returns to scale can be expected as size increases but, beyond a certain point, decreasing returns to scale can be expected.

This research analyses these trade-offs in commercial fourth generation language (4GL) software developments.

## 2 Research Methodology

The research was conducted by reviewing some previously done productivity research [2, 4-8, 10, 12] and interviewing the information systems managers of 10 commercial organisations to develop a questionnaire to gather productivity data [14]. Fifteen organisations developing data processing application systems using 4GLs completed the questionnaire. The size of the systems ranged from a small 29 function point system to a system of 4669 function points. If uncommented 4GL source lines of code (SLOC) are used as a size measure, the systems ranged from 600 to 571 000 SLOC. The development effort required to develop the systems ranged from 40 to 81 270 development hours.

For this analysis the size of the systems was measured in unadjusted function points as it appears to be a more consistent measure than source lines of code [8, 9]. Development effort was measured in development hours. All activities, starting from the requirements specification stage and through to that stage where the product is ready to be delivered to the end-user are included in the development time. The time spent on documentation is therefore also included. Excluded from the development person-hours is the time required for the formal user acceptance tests, as well as end-user training. The definition of a person-hour is the actual time spent on the project, and also includes all time spent attending meetings directly related to the software development, but excludes times such as public holidays, leave, illness, and development staff training.

In the determination of the average size of the development team both analysts and programmers are included in the calculation, as well as project managers and program librarians directly involved with the development. It excludes users, secretaries, higher management, and other staff not directly involved with the development.

Productivity is defined for this analysis in the economic

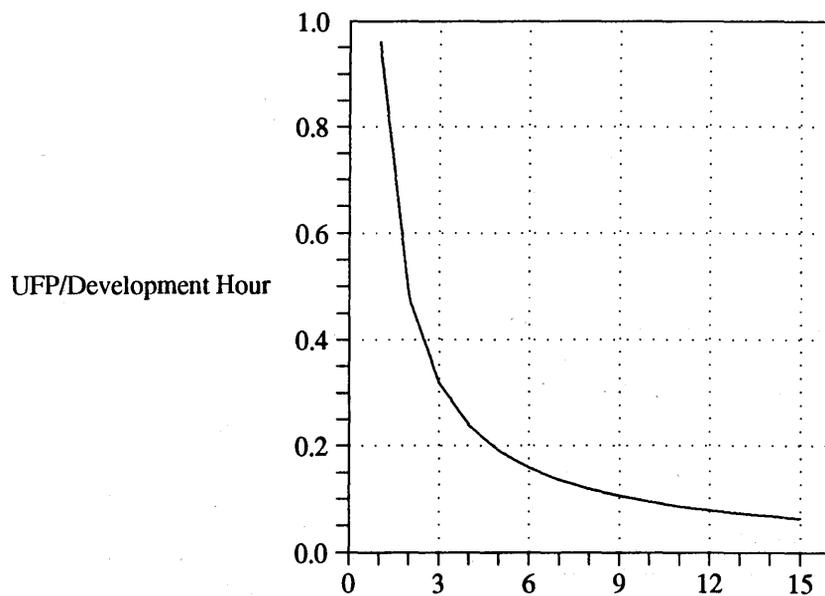


Figure 1. Team size effect on productivity

sense and is expressed as the amount of output produced per unit of input, and the unit used is unadjusted function points developed per development hour.

### 3 The Effect of Team Size Alone

One theory on productivity and average team size is that as the team size is increased and the number of communication paths increase at an almost exponential rate, the increased overhead due to this extra communication burden results in reduced productivity [3, 4, 11]. In contrast none of the Walston-Felix study on productivity [13], the COCOMO model [2] or Function Point Analysis [1] directly identify team size as a significant factor in productivity assessment.

Four functional relationship models were examined. These are:

- Linear:**  $Y = \alpha + \beta X$
- Power:**  $Y = \alpha X^\beta$
- Exponential:**  $Y = \alpha \beta^X$
- Reciprocal:**  $1/Y = \alpha + \beta X$

where  $Y$  = development effort (development hours)  
 $X$  = team size  
 $\alpha$  = constant  
 $\beta$  = coefficient of  $X$

The linear model relating team size to productivity has a poor fit, with a low coefficient of determination ( $R^2$ ) of 0.13 and a  $X$  coefficient which is not significant.

The power function model presumes a power function relationship between average team size and development effort but also fits the data poorly ( $R^2 = 0.35$ ).

The exponential model indicates an exponential decrease in productivity as the average team size is increased. This model indicates however that average team size can be presumed responsible for only 49% of the variance that occurred in the data.

The reciprocal transformation model appears to fit the data much better. The constant was excluded as there appears to be no theoretical justification for it. The adjusted

$R^2$  is 0.83, and the  $X$  coefficient is significant at 0.01. A special feature of this model is that productivity declines non-linearly as the average team size is increased until it becomes asymptotic with the  $x$ -axis. When the results are converted to the normal form, the equation is as follows:

$$\text{Productivity} = (1.04 + \text{A.T.S})^{-1}$$

where A.T.S = average team size.

The effect of average team size on development productivity is depicted graphically as modelled by the above equation in Figure 1.

### 4 Effect of System Size

Plotting development effort versus unadjusted function points (UFP) indicates the increase in development effort as the size of systems increase. A cursory examination suggests, for the sample data, that for the systems in excess of 3 000 function points the development effort increased sharply.

The same four functional relationships were used to investigate the relationship between system size and software development effort. In addition the residual plots and the predicted values plots were examined to try and establish whether any other functional relationship between development effort and system size exists. These four models are:

- Linear:**  $Y = \alpha + \beta X$
- Exponential:**  $Y = \alpha \beta^X$
- Power:**  $Y = \alpha X^\beta$
- Reciprocal:**  $1/Y = \alpha + \beta X$

where  $Y$  = development effort (development hours)  
 $X$  = size (unadjusted function points)  
 $\alpha$  = constant  
 $\beta$  = coefficient of  $X$

For the linear function, the regression of system size against

development hours gives :

$$Y = 9.06X$$

However, the low coefficient of determination (adjusted),  $R^2 = 0.45$ , does not indicate a good linear relationship between size and development effort.

In analysing development effort as an exponential function of size, the data has to be transformed to fit the linear regression model. The independent variable, unadjusted function points, is regressed against the natural logarithm (ln) of the dependent variable, development hours, giving the equation:

$$\ln Y = \ln \alpha + (\ln \beta)X \text{ or } Y = \alpha \beta^X$$

The coefficient of determination (adjusted) of 0.84 indicates a fairly good fit. Both the constant and the  $X$  coefficient are highly significant.

Substituting values from the regression analysis gives:

$$\ln(\text{Dev.Hours}) = \ln 5.19 + \ln 0.001 * \text{UFP}$$

or

$$\text{Dev.Hours} = 179.73 * 1.001^{\text{UFP}}$$

This model suggests that there is a constant growth in development effort as the absolute size of a system increases.

To analyse effort as a power function of size, the data again has to be transformed to fit the linear regression model which is achieved by regressing the natural log of the unadjusted function points (size) against the natural log of development hours (effort), which produces the equation:

$$\ln Y = \ln \alpha + \beta \ln X$$

This model is appropriately also known as the double-log or constant-elasticity model. The attractive feature about this model is that the slope coefficient measures the elasticity of  $Y$  with respect to  $X$ , that is the percentage change in  $Y$  for a given (small) percentage change in  $X$ . The adjusted  $R^2$  is 0.85, which suggests a reasonably good fit. Substituting values gives

$$\ln(\text{Dev.Hours}) = -2.06 + 1.41 * \ln \text{UFP}$$

which is converted to

$$\text{Dev.Hours} = 0.13 * \text{UFP}^{1.41}$$

Various reciprocal models were examined. The model which did fit the data well was the regression of the natural log of UFP against the reciprocal of the natural log of development hours i.e.

$$\frac{1}{\ln Y} = \ln \alpha + \beta \ln X$$

The model fits the data well (adjusted  $R^2 = 0.89$ ), and both the constant and the  $X$  coefficient are highly significant.

The transformed data regression model gives the equa-

tion

$$\frac{1}{\ln(\text{Dev.Hours})} = 0.37 - 0.033 \ln \text{UFP}$$

which, converted back to its original form results in

$$Y = e^{1/(0.37-0.033*\ln \text{UFP})}$$

## 5 Effect of Average Team Size and System Size

Intuitively it would be expected that the larger the system being developed, the larger the development team will be. The plot of the size of systems in unadjusted function points versus average team size suggests that in the research sample larger systems are generally developed with larger teams. The Spearman rank correlation coefficient of 0.74 between average team size and unadjusted function points appears to substantiate this as well. It is thus useful to examine the combined effect of these two variables on development productivity.

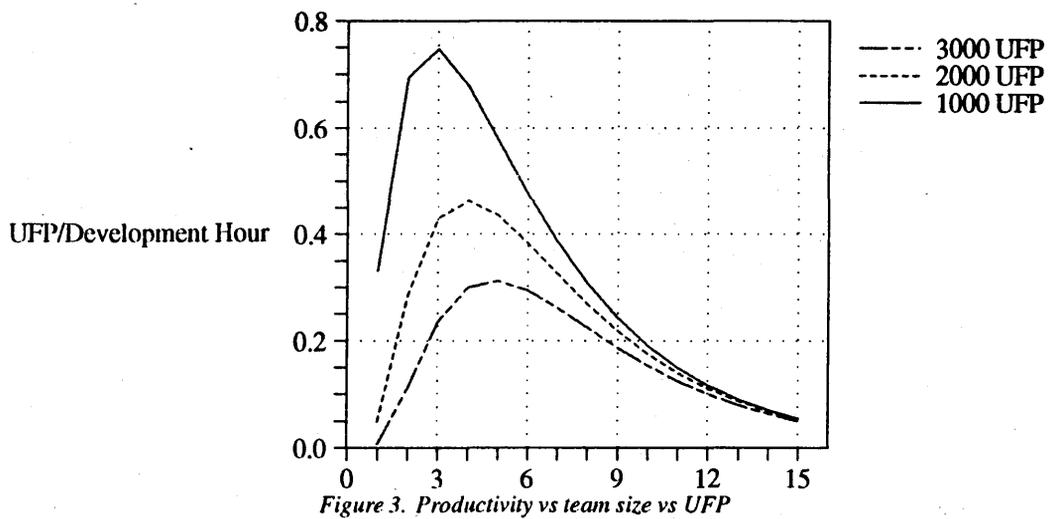
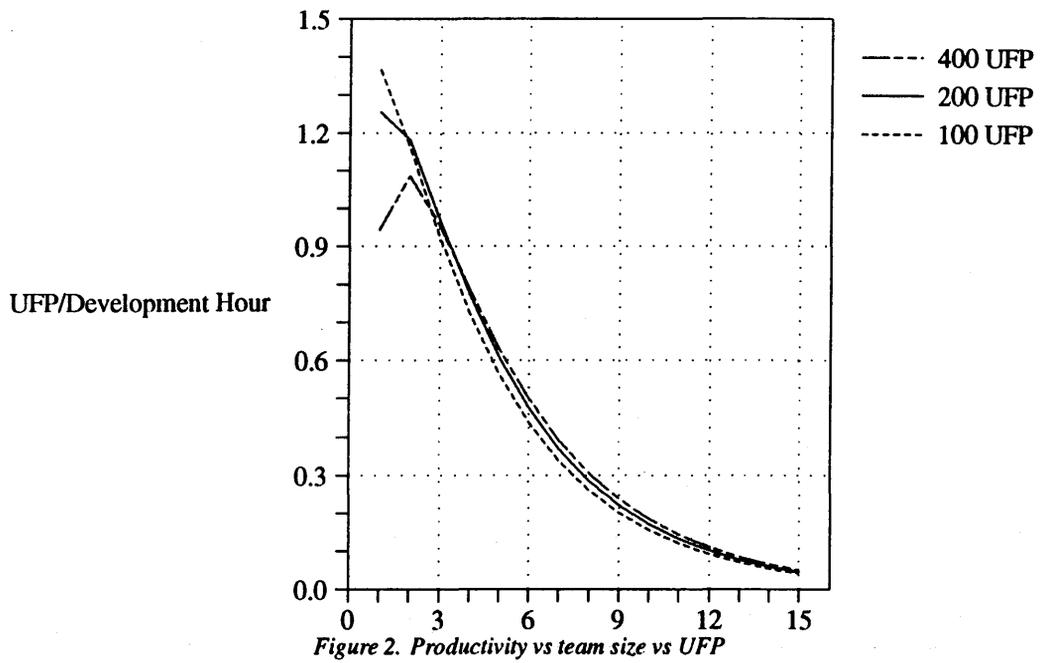
Numerous models were developed and tested, including the linear model, the semi-log model (power function model) and the reciprocal model. The model which performed well was developed after experience gained from developing the total schedule and team size trade-off models. Another variable, the average number of function points developed per team member was introduced.

For this model the average team size (ATS), the number of function points developed per team member (FP/M), and the natural log of unadjusted function points (ln FP) are regressed against the natural log of development productivity (function points developed per hour). The constant of this regression was not significant, and there appeared to be no theoretical basis for not excluding it. The adjusted coefficient of determination is 0.72, and two coefficients of the independent variables are significant at 0.05.

The log function model exhibits some interesting characteristics. The model is graphically depicted in Figure 2 where three systems of 100, 200, and 400 function points are modelled. The 100 and 200 UFP systems achieve their highest predicted development productivity when the development team size consists of only one member. For each additional member the productivity falls, until it appears to reach some asymptotic value. The 400 UFP system predicts an improvement in productivity as the team size is increased from one to two members but as the team size is increased further, the reduction in productivity follows a similar pattern to the previous two cases.

Figure 3 graphically depicts the same model, but for systems of 1 000, 2 000, and 3 000 UFP. In all cases the predicted productivity initially increases as the team size is increased and reaches an optimum size, before further increases depress productivity.

This empirically based model predicts different optimum development team sizes for different size systems. For example, the predicted optimum team size for a 100 function point system is one, for a 400 function point system is two, for a 1 000 function point system is three, and



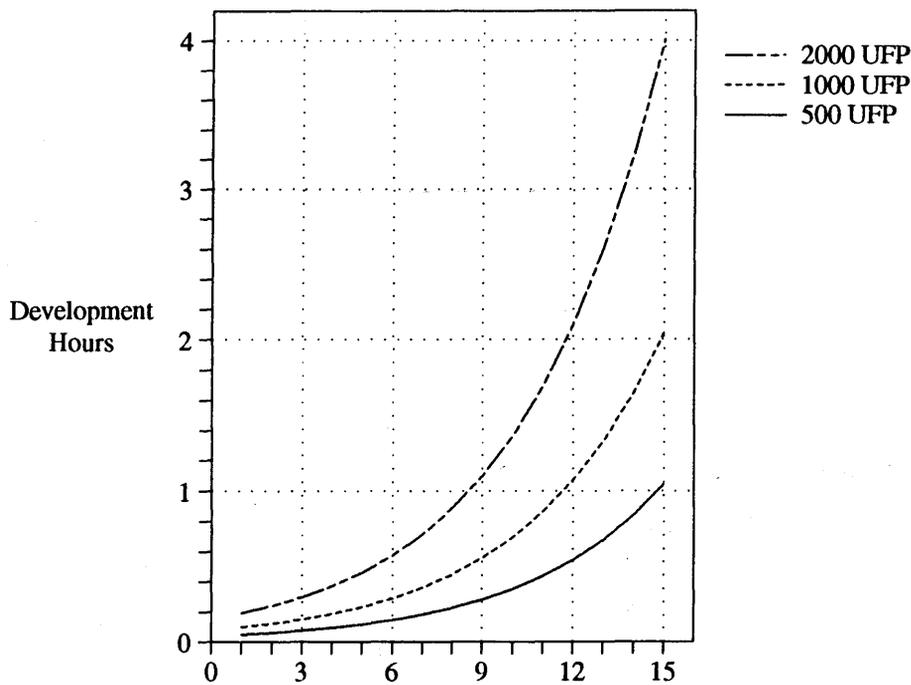


Figure 4. Development effort prediction model

for a 2 000 function point system is four. This trend of predicting larger optimum team sizes as system size increases supports the intuitive perception that larger systems are better able to employ the larger development teams.

## 6 Examining Schedule and Team Size Trade-offs

In software development there is an interrelationship between the total effort required to develop the system (development cost), the average team size, and the total development schedule. The research results examined in Section 3 indicated that development productivity dropped as the average team size was increased. To optimize a development it is necessary to consider the benefits which will be derived from a system, as well as its development cost. Often the earlier the system becomes available, the greater the benefits. To develop a system faster (i.e. to reduce the total development schedule) will require a larger development team, which may result in a drop in development productivity, and an increase in the production costs. Various functional relationships were investigated to examine the trade-offs of development team size and development schedule.

A linear model fitted the data well ( $R^2$  (adjusted) of 0.84). The double-log model also shows a good fit ( $R^2$  of 0.89) but the constant and the team size coefficient are not significant.

The model was varied by regressing average team size and the natural log of unadjusted function points against the natural log of development hours and by forcing it through its origin, thereby eliminating the constant, which produces a very good fit with an  $R^2$  adjusted of 0.99. Both  $X$  coefficients are significant at 0.01.

Numerous other models were developed and tested.

All appeared inferior to the above model, which also had the lowest MAE. Converting the results back into their original form results in the equation:

$$\text{Development hours} = 1.24^{\text{ATS}} * \text{FP}^{0.97}$$

In Figure 4 the predicted development effort of three systems of 500, 1 000, and 2 000 function points is shown. This increases non-linearly as the size of the development team is increased from 1 to 15 members. However a closer examination of these graphs reveals a problem with the model.

The management decision concerns the trade-off of total development cost (development hours) versus the benefit of installing the system sooner (Figure 5). As team size is initially increased the schedule decreases. Further increases in team size result in an ever reducing schedule, until the drop in productivity exceeds the benefit of a larger team, and the schedule actually increases. The problem of the model arises as it predicts in each case a shortest schedule for a five member team. Intuitively it appears unlikely that a team size of five would be the optimum team size for say the 29 function point system which consumed 40 development hours, as well as the 4 113 function point system which consumed 81 270 development hours .

If the above model is correct, then the reduction in productivity by increasing the team size to 15 members would result in the same reduction in productivity for both the 29 function point system as well as the 4 113 function point system. The intuitive argument is that larger systems are better able to optimally employ larger teams, and this is not reflected in the model.

To test this theory the model was adapted, and another variable, the number of function points developed per team member in each development, was introduced. For the model selected, the  $R^2$  (adjusted) is 0.99, and all

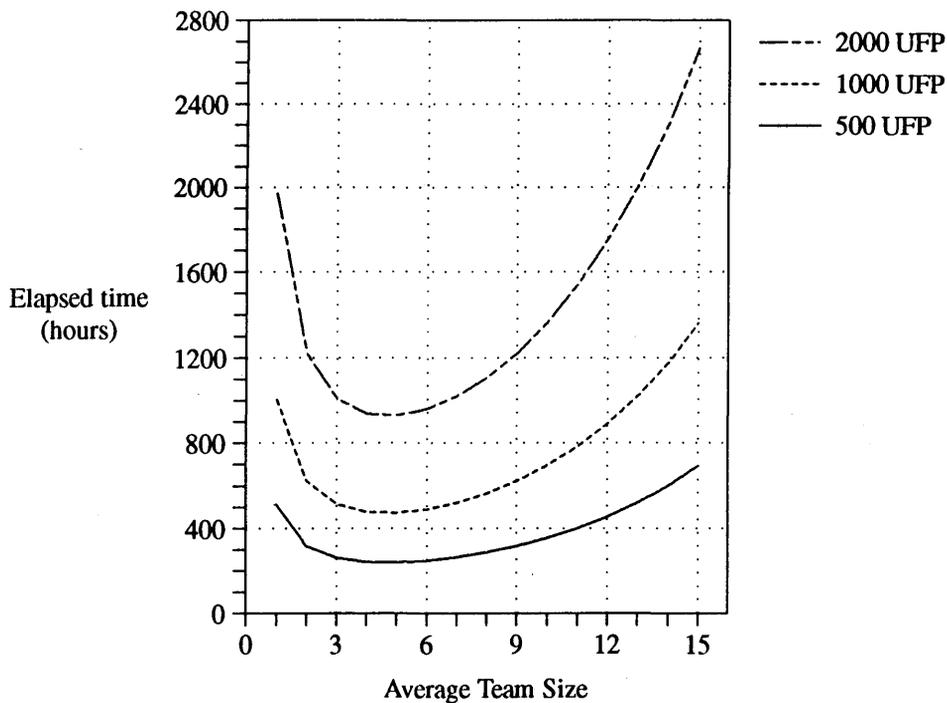


Figure 5. Elapsed time prediction model

coefficients are significant at 0.05. This gives the equation:

$$D.H. = 1.298^{ATS} * 1.002^{FP/M} * FP^{0.832}$$

where D.H. = total development hours  
 FP/M = function points developed per team member  
 ATS = average team size  
 UFP = unadjusted function points

Three systems of 500, 1000, and 2000 unadjusted function points are modelled using this equation in Figure 6. As the team size is increased there is initially a reduction in predicted development hours. The probable reason is that for each system size there is an optimum team size which achieves the highest productivity. The results in Section 5 indicated that for larger systems there is an initial improvement in development productivity as the team size is increased, until the optimum is reached, whereafter it falls. As long as the percentage decrease in productivity is less than the percentage increase in team size the total development time will be reduced.

In Figure 6 the initial large reduction in development time for the larger systems appears to be an anomaly. It may be that if the model has been refined and recalibrated on a larger data sample that this will be corrected.

In Figure 7 the elapsed time is modelled using the same basic equation as in Figure 6. For all systems there is an initial reduction in the predicted elapsed time or schedule as team size is increased. The size of the reduction in schedule diminishes until the shortest schedule is predicted, after which the predicted elapsed time actually increases despite employing a larger team. This appears to confirm "Brooke's Law" which states that adding extra staff to a late project may make it even later [3]. Note that the highest level of productivity and the shortest elapsed time do not necessarily occur at the same team size for the various

system sizes. It is only when the percentage decrease in productivity is greater than the percentage increase in team size that the predicted elapsed time will increase.

## 7 Conclusion

The research results indicate some of the trade-offs which occur in 4GL software developments and which should be considered in order to optimize each development according to its particular circumstances. An analysis of the effect of system size in isolation suggested that the most appropriate model to estimate the effect of size on development effort is the reciprocal function model. In the 4GL software developments which were studied, development effort continuously increases non-linearly as the size of systems increase, and can be modelled by the equation:

$$Y = e^{1/(0.37-0.033*\ln UFP)}$$

An analysis of the effect of team size in isolation indicates that productivity declines non-linearly as the average team size is increased until it reaches some asymptotic value. In the research data larger systems were developed with larger teams. When the combined effect of system size and development team size are modelled, the predicted productivity initially increases and then drops non-linearly until it appears to reach some asymptotic value. This indicates that a system's production cost will initially reduce as team size is increased until the optimum team size is employed, after which any addition will increase the development cost.

The inclusion of an additional variable, function points developed per team member, appears to improve the model, which is able to indicate the optimum team size to result in the shortest development schedule for different size systems. This does not necessarily coincide with the team

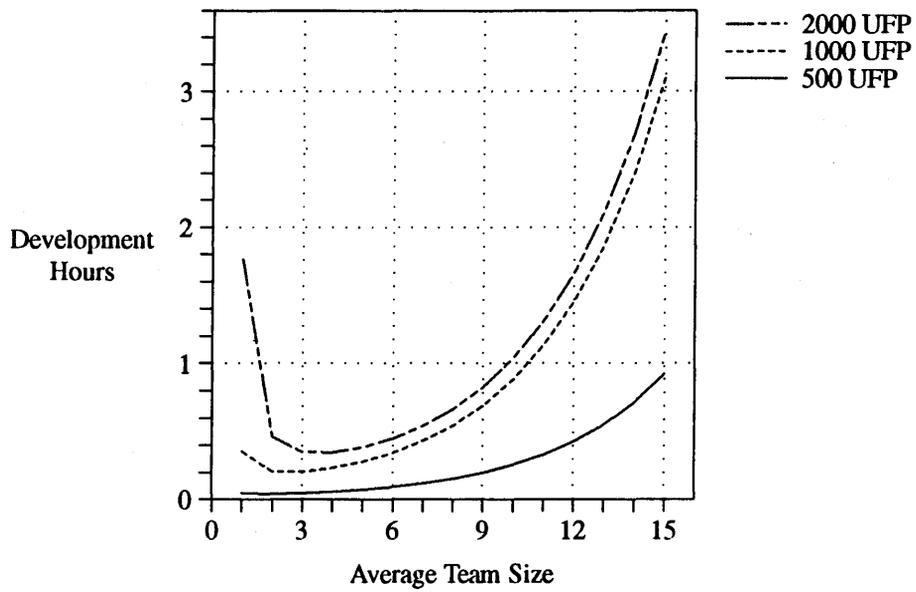


Figure 6. Development effort prediction model

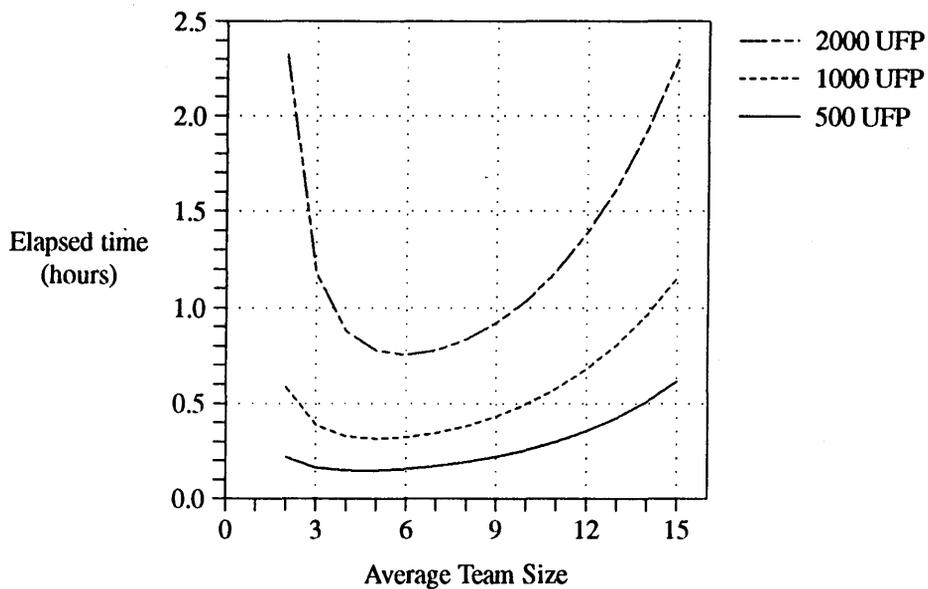


Figure 7. Elapsed time prediction model

size resulting in the highest productivity and thus lowest production cost.

As the development team size is increased and productivity increases the development schedule is initially reduced. When the team size is further increased the productivity drops, and when this reduction in productivity is greater than the increase in development team size, the development schedule increases as the team size is further increased.

This model facilitates an examination of the trade-offs involved in the system size, team size, planned development schedule, and estimated production cost (development effort).

Science Development toward this research is hereby acknowledged. Opinions expressed in this publication, or conclusions arrived at, are those of the authors and are not necessarily to be attributed to the Centre for Science Development.

## References

1. A Albrecht and J Gaffney. 'Software function, source lines of code, and development effort prediction: A software science validation'. *IEEE Transactions on Software Engineering*, 9(6):639-648, (1983).
2. B Boehm. *Software Engineering Economics*. Prentice-Hall, Englewood Cliffs, 1981.
3. F Brooks. *The Mythical Man-Month*. Addison-Wesley, Reading, 1975.
4. S Conte, H Dunsmore, and V Shen. *Software Engineering Metrics and Models*. Benjamin/Cummings, Menlo Park, 1986.
5. D Jeffery. 'A software development productivity model for mis environments'. *Journal of Systems and Software*, 7:115-125, (1987).
6. D Jeffery and G Low. 'Calibrating estimation tools for software development'. *Software Engineering Journal*, 7:215-221, (1990).
7. C Jones. *Programming Productivity*. McGraw-Hill, New York, 1986.
8. C Kemerer. 'An empirical validation of software cost estimation models'. *Communications of the ACM*, 30:416-429, (1987).
9. G Low and D Jeffery. 'Function points in the estimation and evaluation of the software process'. *IEEE Transactions on Software Engineering*, 1(1):64-71, (1990).
10. E Rudolph. Productivity in computer application development. Auckland University Research Report, Auckland University, Auckland, 1986.
11. S Schach. *Software Engineering*. Richard Irwin Inc., and Aksen Associates Inc, 1990.
12. I Vessey. 'On program development effort and productivity'. *Information and Management*, 10:255-266, (1986).
13. C Walston and C Felix. 'A method of programming measurement and estimation'. *IBM Systems Journal*, 1:54-73, (1977).
14. G Wittig. An analysis of 4gl software development productivity. M Com, 1991.

**Acknowledgements:** The authors wish to thank all commercial organisations who submitted data of their software developments. The financial assistance of the Centre for

## Notes for Contributors

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems, as well as shorter technical research papers. However, non-refereed review and exploratory articles of interest to the journal's readers will be considered for publication under sections marked as Communications or Viewpoints. While English is the preferred language of the journal, papers in Afrikaans will also be accepted. Typed manuscripts for review should be submitted in triplicate to the editor.

### Form of Manuscript

Manuscripts for *review* should be prepared according to the following guidelines.

- Use wide margins and 1½ or double spacing.
- The first page should include:
  - title (as brief as possible);
  - author's initials and surname;
  - author's affiliation and address;
  - an abstract of less than 200 words;
  - an appropriate keyword list;
  - a list of relevant Computing Review Categories.
- Tables and figures should be numbered and titled. Figures should be submitted as original line drawings/printouts, and not photocopies.
- References should be listed at the end of the text in alphabetic order of the (first) author's surname, and should be cited in the text in square brackets [1–3]. References should take the form shown at the end of these notes.

Manuscripts accepted for publication should comply with the above guidelines (except for the spacing requirements), and may be provided in one of the following formats (listed in order of preference):

1. As (a) L<sup>A</sup>T<sub>E</sub>X file(s), either on a diskette, or via e-mail/ftp – a L<sup>A</sup>T<sub>E</sub>X style file is available from the production editor;
2. As an ASCII file accompanied by a hard-copy showing formatting intentions:
  - Tables and figures should be on separate sheets of paper, clearly numbered on the back and ready for cutting and pasting. Figure titles should appear in the text where the figures are to be placed.
  - Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin, if they are not clear in the text.

Further instructions on how to reduce page charges can be obtained from the production editor.

3. In camera-ready format – a detailed page specification is available from the production editor;
4. In a typed form, suitable for scanning.

### Charges

Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect scanning, typesetting, reproduction and other costs. Currently, the minimum rate is R30-00 per final page for L<sup>A</sup>T<sub>E</sub>X or camera-ready contributions and the maximum is R120-00 per page for contributions in typed format (charges include VAT).

These charges may be waived upon request of the author and at the discretion of the editor.

### Proofs

Proofs of accepted papers in categories 2 and 4 above will be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Note that, in the case of camera-ready submissions, it is the author's responsibility to ensure that such submissions are error-free. However, the editor may recommend minor typesetting changes to be made before publication.

### Letters and Communications

Letters to the editor are welcomed. They should be signed, and should be limited to less than about 500 words.

Announcements and communications of interest to the readership will be considered for publication in a separate section of the journal. Communications may also reflect minor research contributions. However, such communications will not be refereed and will not be deemed as fully-fledged publications for state subsidy purposes.

### Book reviews

Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

### Advertisement

Placement of advertisements at R1000-00 per full page per issue and R500-00 per half page per issue will be considered. These charges exclude specialized production costs which will be borne by the advertiser. Enquiries should be directed to the editor.

### References

1. E Ashcroft and Z Manna. 'The translation of 'goto' programs to 'while' programs'. In *Proceedings of IFIP Congress 71*, pp. 250–255, Amsterdam, (1972). North-Holland.
2. C Bohm and G Jacopini. 'Flow diagrams, turing machines and languages with only two formation rules'. *Communications of the ACM*, 9:366–371, (1966).
3. S Ginsburg. *Mathematical theory of context free languages*. McGraw Hill, New York, 1966.

---

# Contents

## GUEST CONTRIBUTIONS

Ideologies of Information Systems and Technology <b>LD Introna</b> . . . . .	1
What is Information Systems? <b>TD Crossman</b> . . . . .	7

---

## RESEARCH ARTICLES

Intelligent Production Scheduling: A Survey of Current Techniques and An Application in The Footwear Industry <b>V Ram</b> . . . . .	11
Effect of System and Team Size on 4GL Software Development Productivity <b>GR Finnie and GE Wittig</b> . . . . .	18
EDI in South Africa: An Assessment of the Costs and Benefits <b>G Harrington</b> . . . . .	26
Metadata and Security Management in a Persistent Store <b>S Berman</b> . . . . .	39
Markovian Analysis of DQDB MAC Protocol <b>F Bause, P Kritzinger and M Sczittnick</b> . . . . .	47

---

## TECHNICAL NOTE

An evaluation of substring algorithms that determine similarity between surnames <b>G de V de Kock and C du Plessis</b> . . . . .	58
--	----

---

## COMMUNICATIONS AND REPORTS

Ensuring Successful IT Utilisation in Developing Countries <b>BR Gardner</b> . . . . .	63
Information Technology Training in Organisations: A Replication <b>R Roets</b> . . . . .	68
The Object-Oriented Paradigm: Uncertainties and Insecurities <b>SR Schach</b> . . . . .	77
A Survey of Information Authentication Techniques <b>WB Smuts</b> . . . . .	84
Parallel Execution Strategies for Conventional Logic Programs: A Review <b>PEN Lutu</b> . . . . .	91
The FRD Special Programme on Collaborative Software Research and Development: Draft Call for Proposals . . . . .	99
Book review . . . . .	102

---