19th Meeting of the Voorburg Group on Service Statistics

Ottawa, Canada
September 2004

SESSION II: IT RELATED PRODUCER PRICE INDEXES

FEASIBILITY OF ADOPTING FUNCTION POINT ANALYSIS FOR QUARTERLY SOFTWARE DEVELOPMENT PRODUCER PRICE INDEXES

Matt Berger

AUSTRALIAN BUREAU OF STATISTICS

1. Introduction

The Australian Bureau of Statistics (ABS) has recently undertaken research into the development of a price index for own-account software development. An outline of the proposed methodology using function points was presented in Berger (2003). Subsequent development activity has resulted in three key findings:

(i) the real price of software appears to be falling at around 2.2 percent per annum;
(ii) measurement of this price movement is not practical for a quarterly PPI, but may be more reasonable for inclusion in the national accounts as a price deflator following further research; and
(iii) there are outstanding issues with the measurement of function points to determine changes in labour productivity.

This paper briefly summarises these findings.

2. The ABS approach

As discussed in Berger (2003), the ABS approach to development of a price index for own-account software development is to incorporate function point analysis to attempt to measure changes in labour productivity in the software industry.

2.1 Function Point Analysis

Function Points are a unit of measure for a software product. They measure software by quantifying the functionality it provides to users based primarily on the logical design. Function Point Analysis is a method of breaking down complex systems (such as software) into smaller components, so they can be better understood and analysed. There are a variety of different methods used to count function points. The method referred here is based on the rules developed by Alan Albrecht (see Albrecht, 1979) and later revised by the International Function Point User Group (IFPUG).

2.2 Preferred approach

Working with the Australian Software Metrics Association (ASMA), the Australian representative of the International Software Benchmarking Standards Group (ISBSG), the ABS is using a data source which identifies completed stages of software development projects, together with their respective function points. Included with these data are labour expended (hours worked) and the duration of the project. These data are for own-account software development, and this forms the basis of current development work.

The aim of utilising function points is to develop a price index that measures price change over time accounting for changes in quality.

The preferred form of the index would measure real price received per unit output. Use of function points allows units of output to be measured to the same quality over time. When dealing with own-account software development, market prices received are not available. As an approximation for price we may instead use unit cost - that is, cost incurred per function point. This approximation is necessary since for own account software development there are no real transaction prices.

$$\text{Unit Price} \cong \text{Unit Cost} = \frac{\text{Total Cost}}{\text{Total Function Points}}$$

This index would then correctly reflect changes in both input costs and quality. However, accurate and complete costs data are not available, although labour data (in terms of hours worked) are provided. It is not possible to construct an index of this form using the data currently available.

2.3 Alternative approaches - overcoming limitations of data

It is noted above that for each software development project labour data are provided together with total function points. If the total cost expressed per unit of labour were available from some secondary source, it would be possible to measure unit cost, and therefore approximate unit price, by using a measure of the following form:

$$\begin{aligned}\text{Unit Price} &\cong \text{Unit Cost} \\ &= \frac{\text{Total Cost}}{\text{Total Function Points}} \\ &= \frac{\text{Total Labour}}{\text{Total Function Points}} \times \frac{\text{Total Cost}}{\text{Total Labour}}\end{aligned}$$

This form can be thought of as a combination of a wage cost index with a labour productivity index.

This approach requires a measure of total costs per unit of labour expended. In practice it is possible to approximate this by considering wage costs per unit of labour. The ABS produces a quarterly wage cost index, which is a pure price index designed to measure changes over time in wage and salary rates of pay. Index numbers for the wage cost index are compiled from hourly wage and salary rates and are unaffected by shifts in the distribution of employees across occupations and industries, and between full-time and part-time jobs. In the quarterly publication *Wage Cost Index, Australia* (ABS cat. no 6345.0), individual indexes are published for various combinations of State and Territory, public and private sectors, broad industry groups and broad occupation groups.

Utilising a wage cost index for the occupation of "Computer Professional" and labour data per function point it is possible to construct a price index for own-account computer software development.

2.4 Putting the method into practice

In 2003, the ABS acquired data pertaining to 663 software development projects conducted in Australia between 1989 to 2001. The data included the size of each project (calculated using Functional Size Measurement), the work effort expended on the project (in person-hours), and additional information on the project context, the type of product developed, the development environment, the development techniques and tools applied and information on the composition and experience of the team.

The aim of ABS analysis of these data was to explain changes in the number of *person-hours of work per function point delivered*, and the corresponding productivity measure (measured in function points delivered per 100 hours of work expended). Details of this study, including mechanisms to mitigate some limitations of these data, are contained in Rossiter (2004).

3. Results - observing real price falls in own account software development

Adopting the approach described above, with extrapolation forward for the measured rate of productivity improvement, the ABS constructed an experimental own-account software price index for the period 1997 to 2002 (Table 1 and Figure 1). This experimental index fell by eleven percent between 1997 and 2002 – or 2.2 percent per annum.

Table 1 Experimental price index for own-account software development showing decomposition into Wage and Productivity components

Year	*Wage cost index*	*Labour productivity index*	*Price index for own-account software*
1997	100	100	100
1998	104	106	98
1999	107	113	95
2000	110	120	92
2001	115	127	91
2002	120	135	89

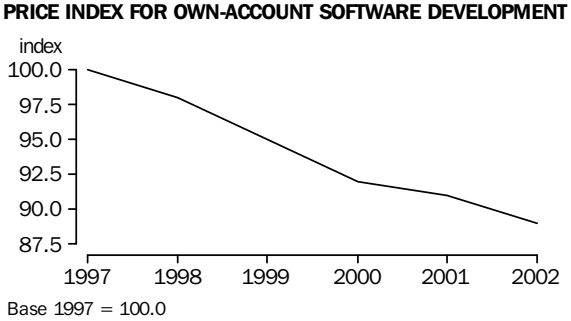

Figure 1

4. Use in a quarterly PPI

The ABS produces PPIs on a quarterly basis, primarily for use in the quarterly Australian National Accounts but also as a measure of upstream inflationary pressure. Unfortunately, the research undertaken above, and the processes outlined in Rossiter (2004), are not practical in terms of compiling a quarterly producer price index. The limitations of this approach are outlined below.

4.1 Necessary Assumptions

a) Unit cost is a good approximation to unit price
This approximation is necessary since own account software developments do not have real transaction prices.

b) Labour as defined in the function point source is consistent with the definition from the wage cost index.
The data available from the wage cost index are a price index for Computer Professionals. Like all groupings of occupations this group can be quite broad. Again, as demand for skills fluctuates, different sectors of this occupation will experience different wage cost

movements (for example, the demand for COBOL programmers for work on Y2K in the late 1990s). Homogeneity of wages within Computer Professionals will vary over time, and the extent to which this varies weakens the assumption upon which the computer software development is based.

c) Function Points

Function points are most frequently used for benchmarking software, and have some subjective aspect in their measurement. Any price index that utilises function points as a mechanism to capture improvements in productivity is using the assumption that function points accurately capture changes in productivity. It is possible that a different measure of function points would reflect a different degree of change.

Recent research (Rossiter (2004)) indicates that productivity varies according to factors such as the size of the project, the size of the development team employed, and the development platform and computing language used. Problems encountered included the uneven distribution of projects over time, large systematic and irregular variations in the mix of project characteristics over time, and inconsistencies in documentation and reporting standards.

4.2 Contemporary data

Contemporary data regarding function points are not currently available, meaning that construction of the labour productivity component is not feasible in real time. Latest function point data received are from the period up to the end of 2001, and measurement of productivity beyond this point has required some degree of extrapolation. The ABS does not revise quarterly PPIs and hence the inability to produce a timely measure of labour productivity limits its application in this area.

4.3 Labour costs, total costs and price

The method being developed makes the assumption that measuring changes in cost is a good approximation to changes in price. This assumption is discussed above. However, a more fundamental concern is that the data available measure wage costs rather than total costs. Wage costs are not the only input into computer software development, as there are substantial capital outlays in computer hardware and software, expenditure on staff training, insurance, costs for other working conditions and so forth.

Investigations to date suggest that total cost of software development is some function of wage costs, although there are currently insufficient data available in Australia to accurately model this function. A potential solution to this problem would be to model total costs as a function of wage costs using data from other markets (for example, software development in the United States). Any resulting function would then have the assumption that total software development costs in the Australia market have the same relationship to wage costs as in the United States.

4.4 Available data and Y2K

The time series of data utilised in development work to date covers the period 1989 to 2001. There is some concern that in addition to the data problems mentioned above that the tail of this series does not accurately reflect changes in productivity of own-account software development. The end of 2001 saw a large portion of programming resources, including those on own-account, being diverted to work on Y2K issues. This also saw demand

pressures on the supply of computer programmers. It is not clear whether these unique demand pressures together with the diversion of resources to particular types of programming effort is correctly reflected in the labour productivity measures produced for this series.

4.5 Outsourcing

This project has focussed on the development of a price index for own-account software development. However, towards the end of the series (late 1990's, early 2000's) Australia saw a shift away from own-account software development towards an outsourced model. It is not clear what impact this has on productivity for software development, and whether there are differences between own-account development and that experienced through the outsourcing process. Given the known shift towards outsourcing, this observation further suggests that future (and even contemporary) data on labour productivity may be quite scarce, which further limits the application of this method in a quarterly PPI.

5. Future work

Despite the limitations that recent findings have shown, particularly with regards to developing a quarterly PPI for own-account software development, research in this field is still promising. In particular, the findings have shown that it in the Australian context it is necessary to include real price falls for software development in the compilation of the Australian national accounts. Further, given that the national accounts are revised as more data become available, it is feasible to consider the future development of a price deflator for own-account software development for inclusion in the capital formation components of the national accounts in subsequent revisions.

Before such a process is undertaken, more frequent if not more timely data are required, and the issues regarding the impact of Y2K, outsourcing and future data availability need to be resolved.

6. References

Albrecht A. J., 1979, "Measuring Applications Development Productivity", *Proceedings of IBM Application Development Joint Share Guide Symposium*, Monterey, CA, pp. 83-92.

Berger, M. , 2003, "Producer Price Index for Software Development in Australia". 2003 Voorburg Group conference paper, Tokyo, Japan.

Rossiter, P. 2004, "Productivity Trends in Computer Software Development", Paper presented to the ABS Methodology Advisory Committee on 18 June 2004, Canberra.

Thanks to Tala Talgaswatta, Peter Rossiter and Shiji Zhao of the Analytical Services Branch, Australian Bureau of Statistics for their assistance in preparation of this paper.