18th Meeting of the Voorburg Group on Service Statistics

Tokyo, Japan
October 2003

SESSION 2: MINI PRESENTATIONS ON PRODUCER PRICE INDICES

PRODUCER PRICE INDEX FOR SOFTWARE DEVELOPMENT IN AUSTRALIA

Matt Berger, author
Michael Anderson, presenter

AUSTRALIAN BUREAU OF STATISTICS

1. Introduction

The Australian Bureau of Statistics (ABS) is undertaking a long-term development program to progressively extend the scope of the producer price indexes into the services sector of the economy. From September quarter 1998, a Computer Services price index has been compiled on a quarterly basis. Of the sub-industries that comprise Computer Services, Computer Consultancy Services contributes 86% to the Computer Services index. Pricing to constant quality has been a challenge for Computer Consultancy Services, (see for example McKenzie and O'Rourke 2002) primarily because of the rapid change in technologies underlying this industry and the consequential changes in productivity. Computer Software Development is a major component of Computer Consultancy Services.

The price index for Computer Software Development is still in a developmental stage and has not yet reached a point where it is ready for publication. This paper reports on progress in developing this price index.

2. Industry output

Software output includes custom software produced on order from specific users, computer programming services provided on a fee or contract basis and ready-made or prepackaged software sold on license to a number of users. The output of the industry is highly diverse. Prepackaged software covers a large heterogeneous range of software including systems software, applications software and other types of software such as games. Output also includes documentation, maintenance and training services. Custom software and contract programming are client specific, and the software and services provided will vary depending on the requirements of the client. Broadly, these outputs can be broken down into three groups: prepackaged, custom and own account.

With the development of computer technology, investment in computer software has increased rapidly in the last two decades. By way of example, in the United States expenditure on software by both government and business increased by 1400% between 1981 and 2001. Although Australia is not a major producer of computer software, a widespread application of computer technology has lead to a similar rate of investment in computer software. As a result computer software has become an increasingly larger component in the investment expenditure of both business and government sectors. The Australian national accounts estimates capital expenditure on software to be approximately 1.3% of GDP.

Recently the US BEA has published a paper, titled, *Recognition of Business and Government Expenditure for Software as Investment: Methodology and Quantitative Impacts, 1959-98,* suggesting different procedures and methods. For prepackaged software, the paper suggested using a combination of hedonic and matched model methods and, for business and government own-account software, it suggested using an index derived from the prices of inputs. A weighted average of prepackaged and own-account software has been used for custom software. The resulting price indexes are dramatically different. For example, the price index between 1959 and 1998 is -10.1% per year for prepackaged software, 4.1% for business own account and 4.7% for government own account.

An increase in the price of (own-account) computer software appears to contradict the economics and anecdotal evidence. It is generally believed that productivity in the software industry has increased significantly as a result of improved technology (ie. more powerful software and hardware tools) and competition from software developers. Higher productivity is likely to have a downward (rather than upward) pressure on software prices and a price index derived from input cost (ie. labour cost in particular) is likely to overestimate the price movement of the (own-account) software products.

3. Weights

Weights for all Computer Services price indexes use 1996-97 Input Output tables, together with data from the ABS Service Industry Survey for Computing Services (1998-99), which was released in *Computing Services Industry, Australia* (ABS cat. no. 8669.0). Computer Services contribute a weight of 13% to the Property and Business Services Division of ANZSIC (which accounts for 7% of GDP). Computer Consultancy Services contribute 86.3% to the Computer Services price index.

4. Index methodology

4.1 Problems with specification pricing

In the ABS price index for Computer Consultancy Services, the pricing of consultancy services are tiered by degree of complexity of the information technology solutions to be put in place as varying requirements for skilled resources, supporting hardware or software and time taken all have to be costed. The very fast pace of technological improvements and the constant evolution of business needs mean changes to terms and conditions of existing contracts necessitate vigilant monitoring to account for data quality issues.

Examples of specifications used in the ABS Computer Consultancy Services index are:
(i) Price for computer consultancy work falling into Tier 3 (as defined by the business), price per job;
(ii) A six month Federal government contract requiring a project manager and two systems analyst programmers (price per contract).

(iii) The ABS also prices hypothetical projects (a form of model-pricing) nominated for a certain time frame with the supporting input of different staff levels. Here, the hourly charge-out rates for each staff level are used for index construction.

Specification pricing has proved to be an effective measure of the price business are receiving for their service outputs; however, in application it is known that the quality of the actual service provided is not assessed. There is no doubt that the quality of these software development outputs has been continually improving over time, due to improved knowledge / productivity of IT professionals, and substantial improvements in other IT products used as inputs in providing the service (eg,. computer hardware, software applications for debugging code in creating new software, etc.).

The ABS is currently investigating other techniques to price computer software development to constant quality.

4.2 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 Ablbrecht, 1979) and later revised by the International Function Point User Group (IFPUG).

4.3 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. As noted above, direct comparison of prices received takes no account of the improvements in quality and as such would upwardly bias any price index in which they were used.

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.

4.4 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 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. This approach shows great promise. For some unresolved issues, however, see sections 5 and 6 below.

5. Pricing to Constant Quality

The approach being developed for own account software is an attempt to avoid the constant quality issues which arise when using specification and model pricing. There are however issues regarding constant quality for this new metric. In particular, the measure makes several 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 are by necessity a subjective measure. 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.

6. Price measurement challenges

The use of function points is showing promise for use in constructing a price index for own-account computer software. However there remain some outstanding measurement challenges.

6.1 Own-account software

The data available to the ABS are for own-account software development, and do not cover customised software development or prepackaged software. Alternative methods are required for these areas.

6.2 Timing of data

Australian producer price indexes for Service Industries are produced each quarter. It is therefore a requirement for any software development index that both function point data and wage cost data be available in a sufficient timeframe to be incorporated into a quarterly index. The current availability of data in Australia would require some lagging in order for both wages and function points to be incorporated in a timely manner. The impact of this compromise has yet to be fully investigated.

6.3 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.

6.4 Volatility

Software development is very heterogeneous by nature. The data available to the ABS at present consist of a relatively small sample size. Subsequently, changes in composition within sub-groups may dominate the movements of the productivity measure. Resulting measures may be quite volatile. Work is proceeding on increasing sample sizes where possible, and otherwise reducing the variation observed in these measures.

7. References

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

McKenzie, R. and O'Rourke, C., 2002, "Producer Price indices for Computer Services". 2002 Voorburg Group conference paper, Nantes, France.