Engineering Economy, a Required Component of Software Engineering

A Position Paper for the
First Workshop on Economics-driven Software Engineering Research

Steve Tockey
Construx Software
14715 Bel-Red Road #100, Bellevue, WA 98007 USA
stevet@construx.com

1: Introduction

Engineering economy has existed for over 100 years and is a recognized component of the traditional engineering disciplines such as Civil Engineering, Mechanical Engineering, etc. A course in Engineering Economy is at least an elective, and is often a requirement, in traditional engineering degree programs at most colleges and universities.

This paper takes the position that engineering economy is not only relevant to software engineering, but ought to be considered a required component of the software engineering discipline as well (e.g., appear as a required or elective course in the software engineering curriculum). The outline of this position paper is as follows

- An overview of engineering economy
- The relevance of engineering economy to engineering in general
- The relevance of engineering economy to software engineering in particular
- Experience in teaching engineering economy in a software engineering degree program
- Conclusions

2: An Overview of Engineering Economy

The dictionary defines "economy" as [1]

> *thrifty and efficient use of resources*

Engineering economy is applied microeconomics, where the fundamental question is [2]

> *Is it in the best interest of the enterprise to invest its limited resources in a proposed technical endeavor, or would the same investment produce a higher return elsewhere?*

Table 1 shows the topics generally found in an engineering economy course or textbook.

Time-value of money (Interest)
Economic equivalence
Inflation
Depreciation
Income taxes
Decision making among alternatives
Decision making under risk and uncertainty
Evaluating replacement alternatives
Evaluating public activities
Break-even
Optimization

Table 1. General topics in engineering economy

In relation to the topics to be considered at the workshop:

- Interest, economic equivalence, inflation, depreciation, income taxes, and decision making under uncertainty address how tradeoffs are made differently under different economic constraints
- Decision making under risk and uncertainty relate to the role of revenue/opportunity-enhancement versus cost/risk-reduction and the economics of timing/ordering of design decisions (e.g., time-to-market)
- The entirety of the engineering economy curriculum enables engineers to reason in economic terms. It also improves understanding between engineers and management in that it provides a common language through business-relevant concepts such as Return on Investment (ROI), Net Present Value (NPV), etc.

There are a few additional topics, such as depletion, which do not apply to software engineering, but a vast majority of the engineering economy curriculum is relevant to software engineering.

3: The Relevance of Engineering Economy to Engineering in General

Science is defined as [1]

> *a department of systematized knowledge as an object of study; knowledge or a system of knowledge covering general truths or the operation of general laws esp. as obtained and tested through scientific method*

The Accreditation Board of Engineering and Technology (ABET) is the recognized authority for accrediting engineering and technology degree programs at colleges and universities in the United States. ABET defines engineering as [3]

> *… the profession in which a knowledge of the mathematical and natural sciences gained by study, experience, and practice is applied with judgement to develop ways to utilize, economically, the materials and forces of nature for the benefit of mankind*

Comparing and contrasting these definitions shows that science is the pursuit of knowledge and engineering is the application of that knowledge for the benefit of people. For example, Chemistry as a science is concerned with expanding our knowledge of chemical processes in order to better understand and explain phenomena that can be observed in the universe. Chemical engineering, on the other hand, applies the knowledge derived from this "chemical science" to filling human needs. Core to chemical engineering is an understanding of the body of chemical theory. But, in addition, chemical engineering calls upon the practical aspects of chemical processes, such as the design of pressure vessels and waste-heat removal mechanisms, etc. together with an understanding of engineering economy.

Thus, the science branch and the engineering branch of a technical discipline are related but distinct. The science branch is concerned with the continued expansion of the body of theoretical knowledge about that discipline while the engineering branch is concerned with the practical and economical application of that same theoretical knowledge. The following equation is proposed to be a (possibly over-) simplified description of the general relationship between science and engineering

Engineering = Scientific theory + Practice + Engineering Economy

Equation 1

4: The Relevance of Engineering Economy to Software Engineering in Particular

Based on the definition of science, above, computer science can be defined as

> *a department of systematized knowledge about computing as an object of study; a system of knowledge covering general truths or the operation of general laws of computing esp. as obtained and tested through scientific method*

Based on the ABET definition of engineering, software engineering can be defined as

> *… the profession in which a knowledge of the mathematical and computing sciences gained by study, experience, and practice is applied with judgement to develop ways to utilize, economically, computing systems for the benefit of mankind*

Thus, from Equation 1 we can derive

Software Engineering = Computing theory + Practice + Engineering Economy

Equation 2

Both computer science and software engineering deal with computers, computing, and software. The science of computing, as a body of knowledge, is at the core of both. Computer science is concerned with computers, computing, and software as a system of knowledge, together with the expansion of that knowledge. Software engineering, on the other hand, should be concerned with the application of computers, computing, and software to practical purposes, specifically the design, construction, and operation of efficient and economical computing systems.

Leon Levy [4] makes some significant observations regarding engineering economy and its relevance to software engineering

> *… software economics has often been misconceived as the means of estimating the cost of programming projects. But economics is primarily a science of choice, and software economics should provide methods and models for analyzing the choices that software projects must make.*

and

> *In any software project there is always a balance between short term and long term concerns … economic methods can help us make enlightened choices.*

The relevance of engineering economy to software engineering is described in much more detail in [5].

5: Experience in Teaching Engineering Economy in a Software Engineering Degree Program

Several engineering economy textbooks are available, including [2], [6], and [7]. Unfortunately, these are oriented toward industrial and manufacturing applications and are quite foreign to us in the software industry. So, although the topic is definitely applicable to software engineering, appreciating its applicability under these circumstances can be difficult.

In the spring of 1996, while serving as an adjunct professor in the Master's of Software Engineering program at Seattle University, I taught a one-quarter course on engineering economy for software. The course used one of the standard engineering economy textbooks ([2]) with lecture examples and exercises recast for the software industry. Nineteen students enrolled in the session and all gave the course positive reviews.

Although my search was not exhaustive, I was unable to find any other academic institution where comprehensive engineering economy was offered as part of a software

engineering curriculum. Indeed, this was my prime motivation for developing the Seattle University course.

6: Conclusions

This paper presented

- An overview of engineering economy
- The relevance of engineering economy to engineering in general
- The relevance of engineering economy to software engineering in particular
- Experiences in teaching engineering economy in a software engineering degree program

These were presented in support of the position that engineering economy is relevant to software engineering and that it ought to be considered a required component of the software engineering curriculum.

References

1. ____, *The Merriam-Webster Dictionary*, New Edition, Merriam-Webster, Springfield, Massachusetts, 1994.
2. G. Thuesen; W. Fabrycky, Engineering Economy, Eighth Edition, Prentice-Hall, Englewood Cliffs, NJ, 1993.
3. ____, "Criteria for Accrediting Programs in Engineering in the United States", Accreditation Board of Engineering and Technology, Baltimore, Maryland, (http://www.abet.org/eac/98eac-criteria.htm).
4. Leon Levy, Taming the Tiger - Software Engineering and Software Economics, Springer-Verlag, New York, New York, 1987.
5. Steve Tockey, "A Missing Link in Software Engineering", IEEE Software, Vol 14, No 6, November/December, 1997.
6. E DeGarmo, W Sullivan, and J Bontadelli, *Engineering Economy*, 9th ed., Prentice-Hall, Upper Saddle River, NJ, 1993.
7. E Grant, W Ireson, and R Leavenworth, *Principles of Engineering Economy*, 8th ed., Wiley, New York, NY, 1990.