How project duration, upfront costs and uncertainty interact and impact on software development productivity? A simulation approach

Li Liu

Faculty of Engineering and Information Technologies,
The University of Sydney,
Australia
Email: li.liu@sydney.edu.au

Xiaoying Kong*

Faculty of Engineering and Information Technology,
University of Technology,
Sydney, Australia
Email: xiaoying.kong@uts.edu.au
*Corresponding author

Jing Chen

School of Business,
University of Northern British Columbia,
Canada
Email: Jing.Chen@unbc.ca

Abstract: Identifying impact factors on software development productivity and the static relations between the impact factors and performance has been the main focus in the literature. Insight into the dynamic relation between key factors and performance dimensions would expand and complement the conventional wisdom on software development productivity. This is the first study to present such dynamic relationship based on an Analytical Theory of Project Investment. Through simulation, we have demonstrated the dynamic relationship between project duration, the uncertainty level of the perceived project value, the fixed project upfront cost and software development productivity. The findings provide practitioners with insight into how these factors interact and impact on software development project productivity.

Keywords: software development methodology; software development productivity; modelling; simulation.

Reference to this paper should be made as follows: Liu, L., Kong, X. and Chen, J. (2015) 'How project duration, upfront costs and uncertainty interact and impact on software development productivity? A simulation approach', *Int. J. Agile Systems and Management*, Vol. 8, No. 1, pp.39–52.

Biographical notes: Li Liu is currently a Senior Lecturer in Project Management at the University of Sydney. He has a doctorate from UNSW Australia, and a BE in Electrical Engineering. He has one decade of work

experience in electrical/systems engineering, project management, research and consulting. His research interests include management of infrastructure projects, IT/IS project management and control theory. He has published in *IEEE Transactions on Engineering Management*, *Project Management Journal*, *Journal of Information Technology* and a number of other international journals and conferences.

Xiaoying Kong is currently a Senior Lecturer in the Faculty of Engineering and Information Technology at the University of Technology, Sydney. She received her BE and ME degrees in Control Engineering from Beijing University of Aeronautics and Astronautics. She received her PhD in Mechatronic Engineering from the University of Sydney. She has many years work experience in aeronautical engineering and commercial web development. Her research interests include control theory, sensor technologies, data fusion, robotics, software engineering, and web technologies.

Jing Chen is an Assistant Professor at the University of Northern British Columbia. He has a long-term interest in understanding life systems from physical laws. He developed a unified economic theory of life and human societies from the perspective of thermodynamics. He holds a PhD degree in Mathematics and is teaching finance at the University of Northern British Columbia.

This paper is a revised and expanded version of a paper entitled 'Modeling Impacts on Software Development Productivity' presented at 2013 Asian Conference on the Social Sciences (ACSS 2013), Hong Kong, 24–25 December 2013.

1 Introduction

Software development productivity is one of the focal topics in literature on software development. Typical approach concentrates on identifying factors influencing software development productivity and developing strategies for improving productivity. Nevertheless, the relationships between some of these factors and productivity change with software project development lifecycle and other factors such as project uncertainty level. The dynamic relationships between these factors and productivity limit the scope of applications of models on the static relations between impact factors and performance. Insight into the dynamic relation between key factors and performance dimensions would expand and complement the conventional wisdom on software development productivity. One of the widely used approaches to understand the dynamic relationship between factors and performance is through simulation based on theoretical predictions.

Drawing from an Analytical Theory of Project Investment, we simulate the relationships between main factors in software project development and demonstrated the dynamic relationship between project duration, the uncertainty level and the fixed project upfront cost. This is the first study to have presented such dynamic relationships based on the Analytical Theory of Project Investment.

Below relevant literature is reviewed and the setup for the simulation described. Then, the simulation results are demonstrated before conclusions are drawn and future research directions suggested.

2 Literature review

2.1 Conventional approach

Conventional approach to analysing software development productivity typically starts with defining productivity metrics, followed by data collection, metrics analysis and metric evaluation and finally productivity improvement initiatives based on the analysis outcome. Software development productivity is typically measured using software productivity ratio defined as "the relationship of an output primitive to its corresponding input primitive" (IEEE, 1992).

Extant literature mainly focuses on factors that influence the software development productivity and strategies for productivity improvement (Boehm, 1981; Maxwell et al., 1996; de Barros Sampaio et al., 2010). Examples of software development productivity metrics and models include constructive cost model (COCOMO) (Boehm, 1981), functional points (Albrecht, 1979).

Boehm uses the ratio of delivered source instruction (DSI) in the software product to the number of man-months (MM) as the measure of productivity (Boehm, 1981). DSI is equivalent to source line of code (SLOC). There are some problems with SLOC productivity measurement, such as lack of international standard for all software programming languages and irrelevant for tools generated SLOC (Jones, 1991; Maxwell et al., 1996). To avoid SLOC productivity measurement problem, function points and feature points are developed to replace SLOC in productivity measurement (Maxwell et al., 1996). Function points use the functionality count from software product users' viewpoint. Software algorithm complexity is not considered well in function point measurement. Feature point method expanded function point to consider the software algorithm complexity in feature point measurement (Maxwell et al., 1996). Despite the shortcomings of SLOC productivity measurement, many organisations still use DSI/MM or SLOC/MM as software productivity metric (Boehm, 1981; Maxwell et al., 1996).

Based on the metric of software productivity defined, data are collected from projects to measure aspects of the development efforts. Then following analysis of the data, impact factors to the productivity are identified, and strategies for productivity improvement are developed (Boehm, 1981; de Barros Sampaio et al., 2010). For example, using the DSI/MM metric, Boehm (1981) identified and classified software development productivity impact factors based on the cost driver attributes of COCOMO. Table 1 shows according to Boehm's classification, four categories of impact factors: product attributes, computer attributes, personnel attributes, and project attributes. Based on the analysis of these impact factors, Boehm recommended a software productivity improvement program (Boehm, 1981) which has been adopted by many organisations.

Recently, a similar classification scheme developed by de Barros Sampaio et al. (2010) is shown in Table 2.

The focus on software productivity impact factors of the conventional approach limits its application to model the static relations between impact factors and performance. Insight into the dynamic relation between key factors and performance dimensions would expand and complement the conventional approach. One of the widely used approaches to understand the dynamic relationship between factors and performance is through simulation based on theoretical predictions.

Table 1 Productivity impact factors

Classification	*Productivity impact factors*
Product attributes	Required reliability
	Database size
	Software product complexity
	Programming language
	Product size
Computer attributes	Execution time and storage constraints
	Resource control
	Relaxing performance requirements
	Virtual machine volatility
	Computer turnaround time
Personnel attributes	Staffing
	Motivation
	Management
Project attributes	Modern programming practices (MPPs)
	Use of software tools
	Schedule constraint
	Requirements volatility
	Work environment

Source: Boehm (1981)

Table 2 Productivity impact factors by de Barros Sampaio et al. (2010)

Classification	*Productivity impact factors*
Product factor	Reuse
	Software size
	Software complexity
People factor	Motivation
	Quality of management
	Team experience
	Knowledge and skills
Project factor	Team size
	Requirements stability
	Client participation and experience
	Process
	Tools
	Programming language

2.2 Analytical approach

The underpinning theory of analysis of software development productivity in this research is the Analytical Theory of Project Investment. The economic model in the Analytical Theory of Project Investment (Chen, 2005, 2012) is briefly reviewed in this section. The economic model will then be used in the later sections to model the software development productivity.

Below S represents economic value of a project in a unit time, r, the discount rate of return, and s, the rate of uncertainty. The process of S can be represented by a lognormal process:

$$\frac{dS}{S} = rdt + \sigma dz \tag{1}$$

where

$$dz = \varepsilon\sqrt{dt} \tag{2}$$

$\varepsilon \in N(0, 1)$ is a random variable with standard Gaussian distribution.

The process (1) is a stochastic process. Feynman (1948) developed a method of averaging stochastic processes under general conditions, which is called path integral. Kac (1951) extended Feynman's method into a mapping between stochastic processes and partial differential equations, which was later known as the Feynman-Kac formula (Øksendal, 1998). According to the Feynman-Kac formula [Øksendal, (1998), p.135], if

$$G(t,S) = e^{-rt}E^S\left(f\left(S_t\right)\right) \tag{3}$$

is the expected value of a function of S at time t discounted at the rate r, $f(S)$ is a function of S, then $G(t, S)$ satisfies the following equation

$$\frac{\partial G}{\partial t} = r\frac{\partial G}{\partial S} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 G}{\partial S^2} - rG \tag{4}$$

with the initial condition:

$$G(0,S) = f(S) \tag{5}$$

It should be noted that many functions of S satisfy equation (4). The specific property of a particular function is determined by the initial condition (5).

The total software development project costs include pre-committed fixed cost component K, and variable cost component. For a software project to be viable, the total cost of operation has to be less than the total revenue. In general, production factors that last for a long time, such as capital equipment, are considered fixed cost while production factors that last for a short time, such as raw materials, are considered variable costs. In software projects, fixed cost is the project cost incurred or committed before the start of the software development project. It may include costs for feasibility study, evaluation, bidding, pre-planning, and some equipment costs. Typically, a lower variable cost system requires a larger investment in fixed costs, though the converse is not necessarily true. It is usually possible to adjust the level of fixed and variable costs to achieve high level of return on our investment given particular environmental constraints.

From Chen (2005, 2012) and Chen and Galbraith (2011), the expected variable cost in generating S, value of a product in a unit time, can be represented by the following analytical formula:

$$C = SN(d_1) - Ke^{-rT}N(d_2) \tag{6}$$

where the function $N(x)$ is the cumulative probability distribution function for a standardised normal random variable. d_1 and d_2 are calculated using:

$$\begin{aligned} d_1 &= \frac{\ln(S/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \\ d_2 &= \frac{\ln(S/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T} \end{aligned} \tag{7}$$

This takes the same form as the Black-Scholes formula for European call options (Black and Scholes, 1973). But the meanings of the parameters in this theory differ from that in the option theory (Chen, 2005). Formula (6) provides an analytical formula of variable cost as a function of project value, fixed cost, uncertainty, duration of project, and discount rate of the project.

In the following sections, we will apply this theory to analyse the software development productivity using economic values.

The notations and definitions in this paper are listed in Table 3.

Table 3 Notations and definitions

S	Economic value of a project in a unit time
$S_{\_Total}$	Total economic value of a project
K	Total pre-committed cost of a project, or fixed cost
C	Expected variable cost in generating S
$C_{\_Total}$	Total variable cost of a project
T	Duration of a project
r	Discount rate of return of a project
σ	Rate of uncertainty of a project
Productivity_Eco	Metric for software development productivity
ε	Random variable with standard Gaussian distribution
$f(S)$	A function of S
$G(t, S)$	Expected value of a function of S at time t
$N(x)$	Cumulative probability distribution function for a standardised normal random variable x

3 Research design

Different from the conventional approach, we model the dynamic relationship between selected key project factors and software development productivity using the Analytical Theory of Project Investment.

We use the IEEE standard for productivity metric (IEEE, 1992), and consider the economic values of output primitive and corresponding input primitive. Name the software productivity metric in economical view point as *Productivity_Eco*. This metric is defined as the ratio of the economic value of project output primitive to the economic values of the corresponding input primitive.

Suppose the total software project value is $S_{\_Total}$. Then, the economic value of project output primitive is $S_{\_Total}$. The total input primitive is the combination of the total pre-committed project fixed cost and total variable cost of the project K and $C_{\_Total}$. *Productivity_Eco* is the ratio of total project value $S_{\_Total}$ to the sum of total fixed cost K and total variable cost $C_{\_Total}$:

$$Productivity\_Eco = \frac{S_{\_Total}}{K + C_{\_Total}} \tag{8}$$

Considering the unit value of the project S in a unit time, expected variable cost C in generating S, and project total duration T, this productivity metric can be calculated using the economic theory in formulas (6) and (7) as follows:

$$Productivity\_Eco = \frac{S \times T}{K + C \times T} \tag{9}$$

where C is calculated using the formulas (6) and (7) from the Analytical Theory of Project Investment:

$$C = SN(d_1) - Ke^{-rT}N(d_2)$$

and

$$\begin{aligned} d_2 &= \frac{\ln(S/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T} \\ d_2 &= \frac{\ln(S/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \end{aligned} \tag{10}$$

Based on the above model, the impact factors to the productivity of software project can be analysed. From equation (9), parameters affecting *Productivity_Eco* are: the software project value in unit time S, uncertainty level σ, project duration T, discount rate r, and pre-committed project cost K.

The analysis is conducted using Excel and the results are presented below.

4 Results and discussions

The economic model of software productivity provides an analytical tool to identify and analyse productivity and impact factors. It also provides a simulation approach for strategy implementation for productivity improvement. In this section, a simulation approach is demonstrated to analyse the impact factors to software productivity.

The simulation in this demonstration is set up using the following model parameters:

S software project value per month; (unit: dollar)

K total pre-committed project fixed cost; (unit: dollar)

C project variable cost per month; (unit: dollar)

T project duration; (unit: month)

r discount rate per month; (unit: %)

σ project uncertainty level; (unit: %).

4.1 Impact of project value uncertainty on project productivity

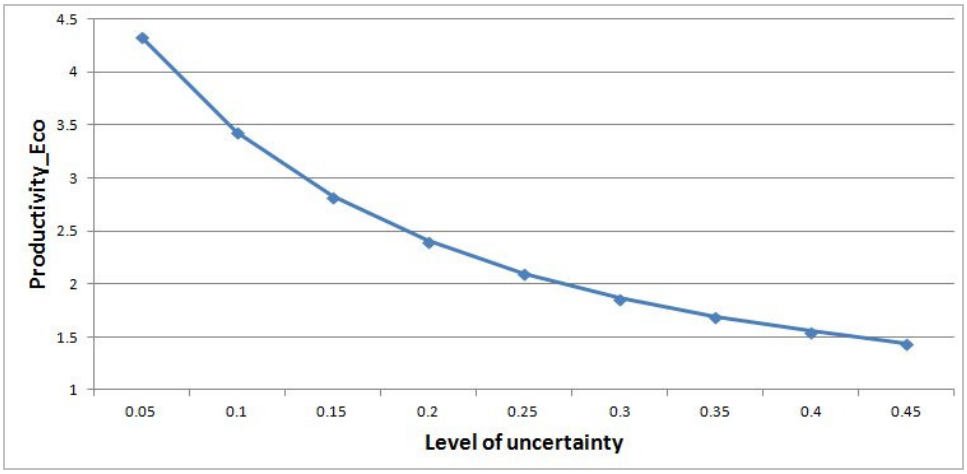
Realising expected benefits from software development projects is a major challenge (Hawking et al., 2004). Value of software development projects depends on a wide range of factors including fit for purpose, adoption by users and contribution to productivity improvement (Boehm, 1981; Albrecht, 1979; de Barros Sampaio et al., 2010; IEEE, 1992; Jones, 1991; Maxwell et al., 1996). Because of the nature of software products, it is difficult to predict the value of a software development effort before actually testing the product by users. Software size, complexity, project team experience, client participation and the maturity of software development processes all influence the extent of predictability of value of the development effort (Boehm, 1981; de Barros Sampaio et al., 2010). Project value uncertainty can be directly measured using the variance of project value.

We first simulate impact of uncertainty on software project productivity *Productivity_Eco* using equation (9) by changing the uncertainty level σ. Holding fixed cost K constant while setting project duration T to 12 months, discount rate r at 3% per month and uncertainty level range from 5% to 45%. Figure 1 shows the calculation data in simulation using Excel. The simulation results as illustrated in Figure 2 shows that holding constant fixed cost and project duration, productivity decreases as the level of uncertainty increases. The practical insight here is that to be able to develop software application with good level of productivity, it is important to minimise the level of uncertainty – i.e., clearer understanding of the benefits from the project, the better the development productivity. Thus, efforts should be directed to identifying project benefits and developing benefit realisation plan.

Figure 1 Productivity results when changing uncertainty (see online version for colours)

S ($)	10000	10000	10000	10000	10000	10000	10000	10000	10000
K ($)	13000	13000	13000	13000	13000	13000	13000	13000	13000
r	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
T (month)	12	12	12	12	12	12	12	12	12
σ	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45
d1	0.65	0.46	0.45	0.49	0.55	0.61	0.69	0.76	0.84
d2	0.48	0.11	-0.07	-0.21	-0.32	-0.43	-0.53	-0.62	-0.72
C ($)	1224.64	1827.42	2453.24	3073.31	3678.26	4262.51	4821.91	5353.22	5853.93
Productivity_Eco =ST/(K+CT)	4.33	3.44	2.83	2.41	2.10	1.87	1.69	1.55	1.44

Figure 2 Productivity declines with uncertainty increases (see online version for colours)



4.2 Impact of the pre-committed project cost to productivity

To analyse the impact of the project pre-committed cost, the simulation is set up using a constant uncertainty level 25%, fixed project duration of 12 months, and discount rate of 3% per month. Changing project pre-committed cost from \$5,000 to \$55,000, the simulation data and results of software productivity are illustrated in Figures 3 and 4.

Figure 3 Productivity simulation data and results when changing pre-committed project cost (see online version for colours)

S ($)	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000
K ($)	**5000**	**10000**	**15000**	**20000**	**25000**	**30000**	**35000**	**40000**	**45000**	**50000**	**55000**
r	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
T (month)	12	12	12	12	12	12	12	12	12	12	12
σ	**0.25**	**0.25**	**0.25**	**0.25**	**0.25**	**0.25**	**0.25**	**0.25**	**0.25**	**0.25**	**0.25**
d1	1.65	0.85	0.38	0.05	-0.21	-0.42	-0.60	-0.75	-0.89	-1.01	-1.12
d2	0.78	-0.02	-0.49	-0.82	-1.08	-1.29	-1.46	-1.62	-1.75	-1.88	-1.99
C ($)	6772.23	4579.60	3199.72	2307.62	1709.76	1295.78	1000.98	785.97	625.89	504.59	411.24
Productivity_Eco =ST/(K+CT)	1.39	1.85	2.25	2.52	2.64	2.63	2.55	2.43	2.29	2.14	2.00

Figure 4 shows that, initially, when pre-committed fixed cost increases from a low base, the productivity increases first and then plateaus before decreases with fixed costs. The optimal productivity level is achieved when the amount of fixed cost is at about \$25,000.

The simulation result shows that the investment committed pre-project could lead to productivity increase. However, too much overhead could result in poor productivity. The challenge is to find the level of fixed cost that corresponds to the optimal level of productivity.

Figure 4 Productivity vs. pre-committed project cost (see online version for colours)

In practice, pre-committed project cost typically includes investments in development of software processes, quality systems, staff training, company organisational infrastructure and administration (e.g., project management office, accounting department). To a certain level, investment in these organisational infrastructures is necessary for improving productivity. For example, to be productive, the project personnel need to be well trained, the organisation needs to have developed process for project management, quality assurance and software development. However, the return from these investments will diminish as fixed costs increase and eventually productivity will plummet with increasing fixed costs because over-investment in these infrastructures will increase project overhead and bureaucracy. The challenge is to identify the optimal level of fixed cost which leads to the highest level of software development productivity.

4.3 Impact of software project duration to productivity

The software development productivity metric using equation (9) shows that the duration of the project influences the level of productivity. Simulation data in Excel are illustrated in Figures 5, 6 and 7 using three different uncertainty levels. Figure 8 shows the curves for the changes in productivity as the project duration changes. This simulation is set up using a constant pre-committed fixed cost. Simulating cases for low, medium and high level of uncertainty, respectively, the results show that in all three cases productivity increases with project duration, then plateaus before plummet with increasing project duration. Further, the result shown in Figure 8 indicates that as project uncertainty level increases, the project with higher uncertainty level exhibits lower level of productivity and its optimal duration for optimal productivity level becomes shorter.

The results can be explained using findings on project time pressure on project productivity (Nan and Harter 2009). Very short project duration impose significant time pressure for project delivery which negatively affect project outcomes due to high work stress on project personnel, lack of time for necessary project planning, rushing, crashing and taking short cuts. As duration increases, the project team gets time to do things properly with reduced stress level and, as a result, productivity improves. However, as project duration increase above what is necessary, productivity deteriorates as project team wastes excess time on non-essential tasks such as excessive paper work and bureaucratic processes. The challenge is to identify the project duration with optimal level of software project productivity.

Figure 5 Productivity simulation data and results with low uncertainty level when changing duration (see online version for colours)

			σ = 0.05									
S ($)	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000
K ($)	13000	13000	13000	13000	13000	13000	13000	13000	13000	13000	13000	13000
r	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
T (month)	2	4	6	8	10	12	14	16	18	20	22	24
σ	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
d1	-2.83	-1.37	-0.61	-0.09	0.32	0.65	0.94	1.19	1.41	1.62	1.81	1.99
d2	-2.90	-1.47	-0.73	-0.23	0.16	0.48	0.75	0.99	1.20	1.40	1.58	1.75
C ($)	0.48	37.33	190.79	464.11	820.57	1224.64	1650.57	2081.40	2506.45	2919.31	3316.29	3695.47
Productivity_Eco =ST/(K+CT)	1.54	3.04	4.24	4.79	4.72	4.33	3.88	3.46	3.10	2.80	2.56	2.36

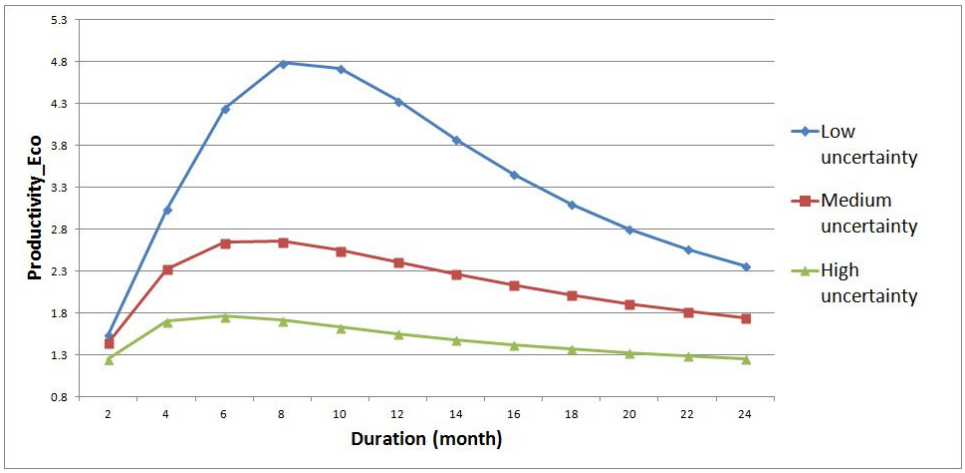
Figure 6 Productivity simulation data and results with medium uncertainty level when changing duration (see online version for colours)

			σ = 0.2									
S ($)	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000
K ($)	13000	13000	13000	13000	13000	13000	13000	13000	13000	13000	13000	13000
r	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
T (month)	2	4	6	8	10	12	14	16	18	20	22	24
σ	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
d1	-0.57	-0.16	0.08	0.24	0.38	0.49	0.58	0.67	0.75	0.82	0.89	0.96
d2	-0.86	-0.56	-0.41	-0.32	-0.26	-0.21	-0.16	-0.13	-0.10	-0.07	-0.05	-0.02
C ($)	433.09	1046.79	1616.72	2140.84	2624.72	3073.31	3490.56	3879.64	4243.16	4583.32	4901.99	5200.82
Productivity_Eco =ST/(K+CT)	1.44	2.33	2.64	2.66	2.55	2.41	2.26	2.13	2.01	1.91	1.82	1.74

Figure 7 Productivity simulation data and results with high uncertainty level when changing duration (see online version for colours)

			σ = 0.4									
S ($)	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000
K ($)	13000	13000	13000	13000	13000	13000	13000	13000	13000	13000	13000	13000
r	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
T (month)	2	4	6	8	10	12	14	16	18	20	22	24
σ	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
d1	-0.07	0.22	0.41	0.55	0.66	0.76	0.85	0.94	1.01	1.08	1.15	1.21
d2	-0.64	-0.58	-0.57	-0.59	-0.60	-0.62	-0.64	-0.66	-0.68	-0.71	-0.73	-0.75
C ($)	1507.34	2631.22	3502.75	4220.05	4828.25	5353.22	5811.77	6215.68	6573.72	6892.67	7177.90	7433.80
Productivity_Eco =ST/(K+CT)	1.25	1.70	1.76	1.71	1.63	1.55	1.48	1.42	1.37	1.33	1.29	1.25

Figure 8 Productivity vs. project duration (see online version for colours)



5 Conclusions and future research directions

Through simulation based on the Analytical Theory of Project Investment, we have demonstrated the dynamic relationship between project duration, the uncertainty level, the fixed project upfront cost and project productivity. Specifically, when holding other variables constant, productivity decreases as the level of uncertainty increases; the productivity increases with pre-committed fixed cost first and then plateaus before decreases; and, productivity increases with project duration initially, then plateaus before plummeting with increasing project duration. Further, the results show that as project uncertainty level increases, the project with higher uncertainty level exhibits lower level of productivity and its optimal duration for optimal productivity level becomes shorter.

Calculations from the theory also show that when project value is high, the corresponding optimal fixed cost for highest rate of return also increases. For example, the development of a safety controller for chemical, railway and other critical applications often requires very high fixed cost investment. New quality requirement may be difficult and expensive to implement for the first time but become easier and cheaper to meet over time in subsequent software development projects. The learning effect can be modelled as the reduction of uncertainty in subsequent developments.

The parameters of the model are based on observable quantities, such as fixed cost, duration of project, rate of return of project. This makes it easier to actually measure the factors involved and to make concrete improvements over time from past experiences. However, one variable, uncertainty, is difficult to measure in practice sometimes. This is a general problem facing all human activities and all living organisms. Because investment is made earlier than the expected future return, the estimation of level of uncertainty always contains an element of uncertainty. This is why there does not exist an optimal strategy under all circumstances. With an analytical theory of investment we can better evaluate what we can measure and try to reduce exposures from what are difficult to measure.

This is a general theory of investment. Since it was first developed, it has been applied to many different areas, such as cooperate finance (Chen, 2006), ecological economics (Chen, 2008), competition of firms of different sizes (Chen and Choi, 2009), social structures (Chen and Galbraith, 2011), monetary policies (Chen, 2012), and many other problems. This is the first study in software development to have presented such dynamic relationship based on economic theory. The findings provide practitioners with insight into how these factors interact and impact on the return of project. Although the results are intuitive, they need to be validated in empirical studies.

References

Albrecht, A. (1979) 'Measuring application development productivity', *Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium*, Vol. 10, SHARE Inc. and GUIDE International Corp., Monterey, CA.

Black, F. and Scholes, M. (1973) 'The pricing of options and corporate liabilities', *Journal of Political Economy*, Vol. 81, No. 3, pp.637–659.

Boehm, B. (1981) *Software Engineering Economics*, Prentice Hall, PTR, Upper Saddle River, NJ, USA.

Chen, J and Galbraith, J. (2011) 'Institutional structures and policies in an environment of increasingly scarce and expensive resources: a fixed cost perspective', *Journal of Economic Issues*, Vol. 45, No. 2, pp.301–308.

Chen, J. (2005) *The Physical Foundation of Economics: An Analytical Thermodynamic Theory*, World Scientific Publishing Co., Singapore.

Chen, J. (2006) 'Imperfect market or imperfect theory: a unified analytical theory of production and capital structure of firms', *Corporate Finance Review*, Vol. 11, No. 3, pp.19–30.

Chen, J. (2008) 'Ecological economics: an analytical thermodynamic theory', in Chapman, R. (Ed.): *Creating Sustainability Within Our Midst*, pp.99–116, Pace University Press.

Chen, J. (2012) 'The nature of discounting', *Structural Change and Economic Dynamics*, Vol. 23, No. 3, pp.313–324.

Chen, J. and Choi, S. (2009) 'Internal firm structure, external market condition and competitive dynamics', *Global Business and Economics Review*, Vol. 11, No. 1, pp.88–98.

de Barros Sampaio, S.C. et al. (2010) 'A review of productivity factors and strategies on software development', *Software Engineering Advances (ICSEA), 2010 Fifth International Conference on*, IEEE.

Feynman, R. (1948) 'The space-time formulation of nonrelativistic quantum mechanics', *Reviews of Modern Physics*, Vol. 20, No. 2, pp.367–387.

Hawking, P., Stein, A. and Foster, S. (2004) 'Revisiting ERP systems: benefit realization', *System Sciences, 2004: Proceedings of the 37th Annual Hawaii International Conference on*, p.8, IEEE.

IEEE (1992) *IEEE Standard for Software Productivity Metrics*, p.1045.

Jones, C. (1991) *Applied Software Measurement: Assuring Productivity and Quality*, McGraw-Hill, New York.

Kac, M. (1951) 'On some connections between probability theory and differential and integral equations', in Neyman, J. (Ed.): *Proceedings of the Second Berkeley Symposium on Probability and Statistics*, pp.189–215, University of California, Berkeley.

Maxwell, K., Luk, W. and Soumitra, D. (1996) 'Software development productivity of European space, military, and industrial applications', *Software Engineering, IEEE Transactions on*, Vol. 22, No. 10, pp.706–718.

Nan, N. and Harter, D.E. (2009) 'Impact of budget and schedule pressure on software development cycle time and effort', *Software Engineering, IEEE Transactions on*, Vol. 35, No. 5, pp.624–637.

Øksendal, B. (1998) *Stochastic Differential Equations: An Introduction with Applications*, 5th ed., Springer, Berlin, New York.