Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 44 (2015) 609 – 618

Procedia Computer Science

2015 Conference on Systems Engineering Research

Shaping the Effort of Developing Secure Software

Ye Yang[a,*], Jing Du[b], Qing Wang[b,c]

[a]*Stevens Institute of Technology, Hoboken 07030, USA*
[b]*Laboratory for Internet Software Technologies, Institute of Software Chinese Academy of Sciences, Beijing 100190, China*
[c]*State Key Lab of Computer Science, Institute of Software Chinese Academy of Sciences, Beijing 100190, China*

Abstract

Effort estimation is extremely challenging for developing secure software systems. Two major challenges are: (1) lack of validated methods or models, (2) large variation in existing security standards that limits applicability of existing methods. This paper reports an exploratory study in establishing effort estimation model for secure operating system software development in China. More specifically, we investigate the existing cost estimation relationships in the domain of secure software systems, then conduct a comparative analysis of existing Chinese IT security standards and the corresponding international standards, and build a customized estimation model to leverage cost estimation relationships with the most similar security requirements, with appropriate adjustment to reflect the differences in standards. The resultant model is evaluated through an example project and results show encouraging improvement in estimation accuracy.

Peer-review under responsibility of the Stevens Institute of Technology.

Keywords: effort estimation; information security; operating system; Common Criteria; COCOMO; COSECMO

1. Background and Motivations

Effort estimation methods and models have been enthusiastically discussed in research community over the past several decades. However, the enthusiasm surprisingly cools down when the methods are put in practice in industry community [1, 2], especially for secure software system. Two major challenges are:

* Corresponding author. Tel.: +1-201-216-8560; fax: +1-201-216-554.
E-mail address: ye.yang@stevens.edu

Peer-review under responsibility of the Stevens Institute of Technology.
doi:10.1016/j.procs.2015.03.041

1) There is a lack of research in estimation methods and models for developing secure software systems, and among these, very few of reported studies offer validated methods or models to address this problem;
2) Existing security standards have been evolving rapidly over the past two decades. Despite the complexity of IT security itself, such standards from different application domains or countries often demonstrates large variations in both contents and structures. The complexity of variations in IT standards place great limitations on the applicability of existing estimation methods.

There are a number of existing cost estimation models and methods that address security-related issues, including COSECMO [5], COCOTS Security Extension [14], SECOMO [16], Function-Point Extension Analysis [15], SQUARE-Based Analysis [17], SAEM [18], and Lawrence Gordon's model [19]. All of existing methods are largely based on international standards such as Common Criteria (CC) [4], and very few of them have been validated. Among those, COSECMO is reported to have only one calibration data point.

The security standard used in developing secure operating system (OS) software products in China is a national standard, GB20272-2006. While it resembles with widely adopted international standards such as CC and Trusted Computer Systems Evaluation Criteria (TCSEC, also known as Orange Book) [8], there are some differences as well. The Chinese standards are presented from the perspective of building a Security Subsystem of an Information System which is the Trusted Computing Base (TCB) of an information system, while the CC are viewed as the rule set for product evaluation [27]. Information security certification in China is based on the Chinese national standards. There have been 132 valid certificates issued from 2011 to 2014 [28].

Although most local companies have adopted and tailored formal estimation models such as COCOMO or FPA to their own environments, those models are generally inapplicable in estimating the development effort for secure software systems. When estimators face difficulties in finding a suitable method, they will not be convinced by the estimates produced by the method. Consequently the method cannot be validated or improved due to lack of use and historical data. The vicious spiral goes on and on. Eventually these methods are abandoned and estimators are driven to guess work or thumb-up decisions. The industry is in desperate need of better approaches and guidance in providing more confidence in their estimates.

2. Research Questions

Three questions were formulated to direct the search and review of existing in estimation method for secure software system development:

(1) How does the national standard GB20272 relate to existing international standards?
(2) What estimation methods or models can be leveraged in our modeling?
(3) What are the existing cost estimating relationships between security level and associated development effort?

The first question intends to capture the similarities and differences between the national standard GB20272 and international ones on information security. Based on the relationship, we can map GB20272 to some widely-used standards, which will provide a feasible way to incorporate the national standard as the benchmark to evaluate the security level referred in the effort estimation method.

According to [4, 5], it is believed that the software system with different security levels will have different costing patterns. The second question is to review on the existing quantitative relationship between security level and associated effort, which will provide a basis for building effort estimation method.

The third question aims at identifying relevant estimation methods or models with security considerations and/or extension. Based on the industry setting, some estimation methods or models may be more suitable to act as the customization basis for our secure software effort estimation.

The rest of the paper describes the research effort in deriving answers to the above questions and developing predictive models to address the sizing and costing issues of security-sensitive software systems.

3. Understanding the Differences in Standards

GB20272-2006 is the basis for the development as well as evaluation of secure operating systems (OS) software. We reviewed some related, widely adopted international security criteria or standards for software development,

including Common Criteria [4], Orange Book [8], ISO27000 Series [9], ISO21827 (SSE- CMM) [10], and some related Chinese national standards such as GB17859-1999 [11], GB20271-2006 [12].

The relationships between GB20272 and other standards are illustrated in Figure 1.

GB 17859 classifies the security protection capability of Computer Information Systems into five levels including discretionary protection, system audit protection, security flag protection, structure protection, and access verification protection. GB/T 20271 details the security functional requirements for each of the five security protection levels defined in GB 17859, for example, it elaborates on security audit and system recovery portion of GB 17859-1999 in more details. GB20272 is the enhancement and refinement of GB17859 and GB20271, with an emphasis on the operating system security requirement during the whole lifecycle. It remains the same five rating scales from level 1 to level 5 in GB 17859, but with much detailed security technical requirements.

The Common Criteria is a security requirement based evaluation criteria. The software product is evaluated to EAL1 to EAL7 for their security attribute. GB/T 20271 shares a high degree of similarities with CC v2.3 in terms of definitions for security functional requirements, security assurance requirements, and technical requirements for security level classification.

The Orange Book was published by Department of Defense in 1985. The security of software system is evaluated from level D to level A based on the security mechanism implemented in the system. It focuses on operating systems, and does not separate security functionalities with security assurance. Now in many countries it is replaced by CC.

ISO27000 series standards mainly focus on the development process for secure IT system. The standard acts more as the guideline than an evaluation standard. SSE-CMM is also a process-oriented standard. The process for secure software development can be evaluated from CL1 to CL5 based on SSE-CMM.

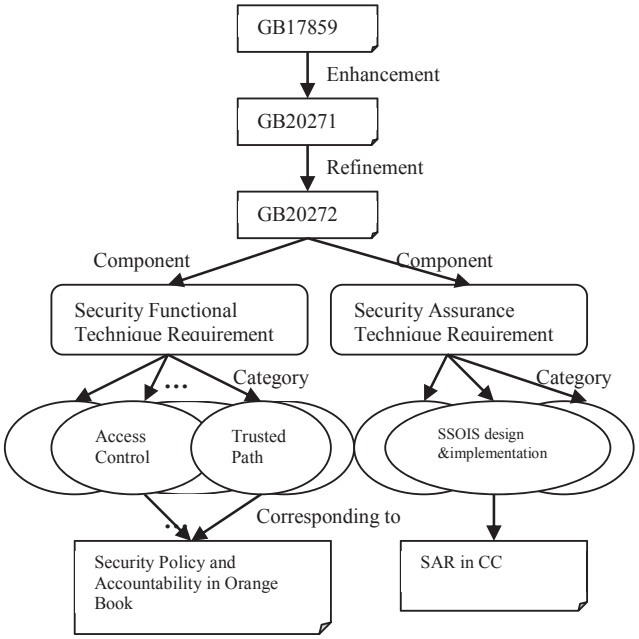

Figure 1. Corresponding Relation among the Standards

We compare the reviewed standards with GB 20272 and score them in a qualitative manner according to their evaluation basis, evaluation target, year and overall feasibility, as summarized in Table 1. Each column of the first three aspects is scored from one to five '*' according to the similarities with GB20272, or potential applicability in the case study setting. Overall applicability is the summary of the former four columns. Since GB20272 is the baseline for analysis, its four aspects are given by detail information instead of scores. Our literature review does not aim to

serve as the basis for systematic analysis, so it is by no means to cover all related literatures. Only widely-used and highly relevant ones are included, which we believe are sufficient for the modeling work in the scope of our study.

Table 1. Comparison of security criteria and standards.

Name	Evaluation Basis	Evaluation Object	Yeah	Overall feasibility
GB20272 [7]	Security Requirements	Operating systems	2006	
Common Criteria [4]	*****	*****	2009	*****
Orange Book [8]	*****	*****	1985	***
ISO27000 Series [9]	*	**	2005	**
ISO21827 (SSE-CMM)[10]	***	**	2008	**
GB17859 [11]	**	*****	1999	**
GB20271 [12]	*****	*****	2006	***

Based on the comparison, CC and Orange Book are the most relevant standard to be mapped to GB20272. To estimate the overall development effort of a secure OS product in compliance with GB20272, it can be mapped to a similar set of security functional requirements with the corresponding security level in the Orange Book, and a tailored set of security assurance requirements in CC.

4. Identifying Reusable Cost Estimation Components

There are a number of existing cost estimation models and methods that address security-related issues, including COSECMO [5], COCOTS Security Extension [14], SECOMO [16], Function-Point Extension Analysis [15], SQUARE-Based Analysis [17], SAEM [18], and Lawrence Gordon's model [19]. The first three are COCOMO-base algorithm model. COSECMO address the security concern by introducing a SECU cost driver based on common criteria. COSECMO attempts to address not only development effort, but also the external evaluation effort, however it has not been validated on any actual project yet. COCOTS security extension model mainly focuses on analyzing security-related risks in COTS-based development and there is no security cost estimation associated. SECOMO is also a COCOMO-like model with an addition of two cost drivers emphasizing on attack and audit to address network security issues. Function point extension analysis treats the security attribute as a size multiplier and provides corresponding effort estimates based on function point analysis. SQUARE analysis is a cost-benefit analysis framework for small companies. Effort estimation for secure software development is not explicitly discussed in this framework. Like SQUARE analysis, SAEM is also a model to predict the cost-benefit for developing the secure software. Lawrence Gordon's model provides the analysis on security investment for general information systems.

Since it is our purpose to develop a suitable effort estimation model for GB20272 based secure software development, we compare the above models and evaluate the similarity between their home grounds and our particular project circumstance. The results are summarized in Table 2.

Table 2. Evaluation of existing security estimation methods.

Name	Output	Referred Standard	Validation	Year	Overall Feasibility
COSECMO [5]	*****	*****	*	2008	****
COCOTS-Security Extension [14]	***	*	**	2003	*
Function-Point Extension Analysis [15]	*****	***	*	2010	***
SECOMO [16]	***	*	**	2008	*

SQUARE-Based Analysis [17]	***	*	****	2004	**
SAEM [18]	***	*	****	2002	**
Lawrence Gordon's model [19]	****	*	*	2002	*

It is noted that COCOMO model is the most feasible basis for security extension and customization. COSECMO and FPEA model the most similar security objectives and concerns as those we need, however, they are developed for international standards, i.e. CC. In our case, we will need to develop an extension of cost driver for security issues with GB20272 as the reference standard.

Meantime, we investigate a few relevant industry reports to understand the unique characteristics during secure software development and figure out the cost estimating relationships. These include the FAA guidelines [20] and DHS process models [21] for developing and sustaining secure software, CSI Computer Crime & Security Survey [23], GAO report [24] and the DoD report on Software Security Assurance [22]. Table 3 summarizes the percentage of added effort due to security enhancement provided by GAO report and COSECMO model. These cost relatives suggests similar trends. Only EAL2 to EAL 4 are reported in GAO report, we assume the effort for EAL2 are "1", the efforts for other EAL are expressed as multiplier of EAL2. COSECMO provides a more complete range corresponding to all EALs from 1 to 7 with respect to difference project sizes [26], but it is for more generic information security rather than OS products. These are good examples and helpful in our method establishment.

Table 3. Cost relative in GAO report.

EAL	1	2	3	4	5	6	7
Effort relative	n/a	1	1.2-3.0	1.7-4.0	n/a	n/a	n/a
COSECMO	1	1	1.2-1.8	1.5-3.0	2.25-6.0	4.13-13.5	8.8-32.25

Although for some EALs, there are no effort relative information in GAO report, still it can be referred as an industry benchmark to determine how much more effort is needed for higher security level. It should be pointed out that the effort indicated in Table 3 contains the effort spent on external EAL evaluation, which is not included in our case. We will only be modeling the internal development effort.

5. Modeling the Cost of Security in GB20272

In our previous studies, significantly different productivity patterns in Chinese software industry have been reported [2, 29]. One major difference is that software productivity in Chinese software industry is significantly higher than those suggested by the COCOMO II model. In this study, we start with a customized COCOMO II model representing typical productivity level in Chinese software industry (e.g. A=1.32, B=0.94) as the baseline model, add the cost of developing security requirements as emphasized in GB20272 to address additional development cost, and then evaluate the model. To address the security influence on the effort required to accomplish the secure system, a cost driver SECU is introduced into our model to reflect the extra effort needed to achieve a certain security level. The model is indicated in Equation (1).

$$Effort = 1.32 * Size^{E} * \prod_{i=1}^{17} EM_i * EM_{SECU}$$
$$where, E = 0.94 + 0.01 * \sum_{j=1}^{5} SF_j \quad (1)$$

The cost driver SECU needs to be further elaborated with scale and corresponding multiplier. To balance the accuracy and generalizability of our model, we did the following special handling: (1) categorize the security cost driver into five scales from Nominal to Super High in order to correspond to five security levels in GB20272, which also need to be mapped to those from CC and Orange Book, to come up with a unified cost driver scale, (2) set the

industry data shown in Table 3 as the baseline and derive best a-prior for SECU multipliers. Table 4 shows the five rating levels of the proposed SECU driver, and the mapping relationship with those in CC and orange book.

Table 4. Security cost driver rating and numeric values.

SECU	Nominal	High	Very High	Extra High	Super High
CC	EAL1-2	EAL 3	EAL4	EAL5-6	EAL7
Orange Book	D-C1	C2	B1	B2	B3-A1 and above
GB20272	Level 1	Level 2	Level 3	Level 4	Level 5

A mini-delphi was conducted among the authors and an industry expert to derive the corresponding numeric value for each rating level. The security related multipliers in COSECO model, COCOTS security extension model and other quantitative methods have been taken into consideration for reference. The resultant numeric values of the SECU multiplier are shown in Table 5.

Table 5. SECU Multiplier.

SECU Scale	Nominal	High	Very High	Extra High	Super High
Multiplier	1	1.25-1.5	1.75-2.0	2.0-2.75	3.0-4.0

Each rating level corresponds to a range of numeric effort multipliers. There are a couple of advantages doing so. One is that, similar to what COSECMO currently offer, it is a way to accommodate projects of different sizes, which tend to have different influence on additional security development effort even if they are at the same required security level. The other advantage is that, it will provide a range estimate to address certain risks associated with single point estimates. The final multiplier values are generally smaller than the GAO one. This is because the effort estimated in our model does not include the effort spent on external security evaluation, which is a part of the total effort in GAO report.

6. Application

The application of our proposed model is done through a case study project. It was the development of secure operating system software for computer server, enhancing a level 3 product. The project was completed in March 2011 and the external certification was passed at security level 4 of GB20272. The evaluation is organized in three parts: data collection, size modeling, and evaluation result. For data confidentiality reason, we can only publish the final evaluation results of our proposed model.

6.1. Data Collection

The actual effort and size data is extracted from the organizational asset repository.

The effort collected includes the total effort devoted to develop the secure operating system. Independent external evaluation effort is not considered in our model. The effort data is initially extracted from the closure report, and compared with that extracted from the project's process management data repository, tracked on a weekly basis. We summarized the weekly effort to obtain the total effort and confirmed it with the data from the closure report. The effort data derived from the tool is consistent with the data from report. So the actual size and effort data is obtained. Due to the confidential issue, the actual size and effort data cannot be published.

For effort estimation using the proposed model, the 22 COCOMO II cost drivers and the additional SECU cost driver are needed. With the help of the manager and the key developer, the driver scales are derived. The SECU driver is rated as very high (VH) to reflect the GB20272 Level 4 requirement.

6.2. Sizing

The actual size is the total LOC in the final product. Further interviews were conducted with the project manager and key developers, in order to derive equivalent size of the project. The following proxy factors were derived based on the interview feedbacks:

- The total product size contains 66% adopted code, 2% modified code, and 32% new-added code. The adopted size means the intact part of the security level 3 operating system while the modified part means the size that is modified from level 3 system to achieve level 4.
- To convert the adopted part to its equivalent size, a 5% effort is needed for understanding of the adopted code.
- A 40% effort is needed to understand and modify for converting the modified size into its equivalent one.

We then propose Equation (2) to calculate the equivalent size as:

$$\begin{aligned} TotalSize_{equ} &= AdoptedSize_{equ} + ModSize_{equ} + NewSize_{equ} \\ &= 5\% * AdoptedSize + 40\% * ModSize + NewSize \\ &= 5\% * 66\% * TotalSize + 40\% * 2\% TotalSize + 32\% TotalSize \\ &= 36.1\% TotalSize \end{aligned} \quad (2)$$

6.3. Evaluation Results

To evaluate the performance of the model, we choose four existing methods as comparison baselines, including the original COCOMO II, a local COCOMO II representing general productivity in Chinese software industry, the COSECMO, and a local COSECMO which is COSECMO with locally calibrated COCOMO II parameters (i.e. model constants A and B). It should be pointed out that COSECMO uses Common Criteria in security cost driver scale. Since we have integrated those standards into a unified framework, the corresponding scale can be easily mapped.

Based on the size and cost driver input, we apply the chosen models to effort estimation for the security level 4 OS software. Since COSECMO estimates both internal development effort and external evaluation effort, when applying COSECMO, we only calculate the internal development effort with independent evaluation effort excluded. This consideration is consistent with the specific data situation in our setting. The metric Relative Error is used to measure the accuracy of the models. The validation results are shown in Table 6.

Table 6. Evaluation results.

Models	Relative Error (RE)
COCOMO II	-0.73
Local COCOMO II	0.70
COSECMO	-7.93
Local COSECMO	-0.53
Customized Model	0.41

According to the evaluation results, our estimation model has produced the best result compared with the other four models. This indeed proves the positive effect of model local customization based on specific industry setting. Generally speaking, the model performance has been remarkably improved after calibration with local data. Especially the accuracy of calibrated COSECMO model is remarkably increased comparing with the ones before calibration.

7. Discussion

Performance Improvement: There are a few observations about improved estimation performance from results in Table 1.

The first one is that the proposed model provides the best estimation accuracy. It is interesting to note that the general COCOMO II model overestimates the effort by 73%, while the local COCOMO II model underestimate the effort by 70%. These two results confirm that the absence of security considerations in the general COCOMO II model, as well as our previous conclusion on high software productivity in Chinese software industry. However, the 41% of relative error from our customized model indicates that it also underestimates the associated effort. A possible reason is that we reuse the productivity parameters calibrated from general non-security-sensitive software projects. If more data points become available from similar security domains, the model performance can be expected to increase through further calibration.

The second observation is that among these models compared, COSECMO provides an estimate with the largest relative error, which is about 8 times overestimated (i.e. a RE of -7.93). COSECMO is developed largely based on international standard CC, and its calibration is based on 1 data points, which is from security-critical software-intensive systems in the US. It is not surprising considering the compound effects of dramatic differences in terms of software productivities and security standards in the two countries. More specifically, there are several factors causing such differences: (1) The secure OS product in our case is actually a secure OS kernel, and might be in a much smaller size compared with the project that COSECMO was calibrated on. This can be confirmed by the 50% over-estimation of the Local COSECMO estimate. (2) Software industry in China typically has higher productivity as reported in our previous study [2, 29]. (3) Most secure OS products are part of government acquisition contract, and government acquisition systems in the two countries are totally different which greatly affect the development processes and categories of costing items.

The third one is that local calibration helps to improve the estimation accuracy. Both Local COCOMO II and Local COSECMO models outperform the generic ones. This is mainly because the local historical projects have the similar development profile with the targeted one including productivity, project characteristics, development platform etc. These similarities can provide more accurate effort estimating relationships than the general model. The assumption is proved by the increased accuracy after the model is locally calibrated, for both COCOMO II and COSECMO model. This result proves the necessity for a specific model dealing with secure software effort estimation instead of a general one.

Despite of the improved performance, our model is by no means perfect. The impacts brought by security requirements on the other cost drivers have not been explicitly analyzed in our model. For example, a system with a Very High SECU level is required to achieve at least a Very High RELY scale according to COSECMO. However in our case, the RELY scale is merely Low. To analyze this impact, we set the RELY scale as VH and keep the other driver the same, the accuracy is improved to a 0.21 RE after the RELY is changed to VH. Thus, in the future, we intend to analyze the SECU driver's impact on the other drivers' scale setting. Explicit guidelines for driver scale setting in our model will be provide to the estimator to further improve the model performance.

Secure Software Sizing: In our case, the targeted project is a security level 4 operating system upgraded on a security level 3 system. From the size analysis, it indicates that when the software security is enhanced, there will be corresponding increase of software size. We assume that unlike IT systems with complicated human-computer interfaces, the secure operating system focuses more on the security kernel. For a secure OS, the primary security objectives include identification & authentication, confidentiality, security management, accountability [25]. An example list of the most critical tasks consists of handling user accounts and the privileges, such as file access and OS configurations (identification & authentication), generating error message and providing the administrator system log to track the error (Accountability), and providing security schemes of automatic detection of security attack, etc. The increased security of operating system requires more sophisticated security functionality instead of more complicated interface control. The enhanced functionality is always reflected on the increase of system size.

In our model, the size input is decomposed into different categories in GB20272. Although the effort spent on security assurance categories is not reflected in the corresponding size, it gets reflected in the cost driver to derive total effort. We believe that the assurance effort including testing, enhanced requirement and design, other

management tasks and so on can be measured by the size of test cases and other documentation. The separation for sizes of different parts means the potential separation of effort on security function and assurance. This may provide the possible future refinement for our estimation model.

Furthermore, we break down the total size into several parts according to components in GB20272. The security assurance effort is not reflected on the product size, our analysis result shows that about 72% of the total size are related to the six categories of security functional requirements in GB20272. However, the effort referred in our report indicates the total effort including the part contributing to the security assurance.

Threats to Validity: Nonetheless, there are some limitations in our work. First, the literature review is not a systematic one. Although our method is chosen based on the behavior analysis, there is still a chance that some better methods may exist out of our literature review scope. Second, the multiplier for SECU cost driver is based on the industry report and expert judgment. This may affect the accuracy of the estimation result. Third, the calibration of parameters A and B are based on seven historical projects. However, these projects have no particular concern on system security. Thus, the local calibrated parameters A and B in the model may not exactly reflect the secure system characteristics. Last, only data from one project is used to evaluate our model. This may make the conclusion biased by the single data point. Also according to the detail analysis, some effort reported in higher system level has contributed to the secure system development without being counted into this project. This may jeopardize the final result.

8. Conclusions

Effort estimation for secure software is really difficult but vitally important for software companies and organizations. In this paper we report an approach in developing a customized model for developing a specific line of secure software products, secure operating system, in China. The research effort is driven by the actual problems and challenges that chinese software companies are facing when planning for security sensitive software projects due to limited applicability of available methods and mismatch in existing underlying standards. We investigate the existing cost estimation relationships in the domain of secure software systems, then conduct a comparative analysis of existing Chinese IT security standards and the corresponding international standards, and build a customized estimation model to leverage cost estimation relationships with the most similar security requirements, with appropriate adjustment to reflect the differences in standards. The resultant model is evaluated through an example project and results show encouraging improvement in estimation accuracy.

Future work include: (1) to expand our literature review scope so that some other methods may be found to be taken into consideration; (2) to collect more data on secure software projects, and further calibrate and validate our model; (3) to develop practical guidelines for continuously and consistently collecting metric data on sizing, and driver multiplier to support estimation and comparison across companies and standards.

Acknowledgements

This paper is partly supported by the Chinese National Science Foundation (Project #s: 91318301, 91218302, 61432001).

References

1. Moloekken-OEstvold K., Joergensen M, Tanilkan S. S., Gallis H. Lien A. C. and Hove S. W., “A survey on software cost estimation in the Norwegian software industry,” Proc. 10th International Symposium on Software Metrics, Sept. 2004, pp. 208-219, doi: 10.1109/METRIC.2004.1357904.
2. Da Yang, Qing Wang, Mingshu Li, Ye Yang, Kai Ye and Jing Du, “A survey on software cost estimation in the Chinese software industry,” Proc. Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM’08), ACM Press, 2008, pp. 253-262, doi:10.1145/1414004.1414045.
3. Barry W. Boehm, Chris Abts, A. Winsor Brown, et al., “Software Cost Estimation with COCOMO Ⅱ,” Prentice Hall, August 2000.

4. "Common Criteria for Information Technology Security Evaluation," Version 3.1, Revision1, CCMB-2006-09-001, URL: http://www.commoncriteriaportal.org/.
5. Ed Colbert, Barry W. Boehm, "Cost Estimation for Secure Software & Systems," ISPA/SCEA 2008 Joint International Conference.
6. Jing Du, Ye Yang, Zhongpeng Lin, Qing Wang, Mingshu Li and Feng Yuan, "A Case Study on Usage of a Software Process Management Tool in China," Proc. 2010 Asia Pacific Software Engineering Conference (APSEC 2010), pp. 443-452. doi: 10.1109/APSEC.2010.57.
7. Chinese National Standard GB/T 20272-2006, "Inforamtion security technology- Security techniques requirement for operating system," May 2006.
8. Department of Defense Standard DoD 5200.28-STD, "Trusted Computer System Evaluation Criteria," Dec. 1985..
9. ISO/IEC 27002:2005, "Information technlogy-security techniques-Code of practice for information security management", 2005.
10. ISO/IEC 21827:2008, "Information technology-Security techniques-System Security Engineering- Capability Maturity Model (SSE-CMM),"2008.
11. Chinese National Standard GB17859-1999, "Classified criteria for seucrity proteciton of computer information system," 1999.
12. Chinese National Standard GB/T 20271-2006, "Information security technology- Common security techniques requirement for information system," 2006.
13. Jing Du, Ye Yang, Qing Wang, "An analysis for understanding software security requirement methodogies," Proc. Third IEEE International Conference on Secure Software Integration and Reliability Improvement,IEEE Press, 2009, pp. 141-149, doi: 10.1109/SSIRI.2009.14.
14. Donald J. Reifer, Barry W. Boehm, and Murali Gangadharan, "Estimating the Cost for Security for COTS Software," Proc. International Conference on Composition-Based Software Systems, Springer-Verlag Berlin Heidelberg, 2003, pp. 178-186, doi: 10.1007/3-540-36465-X_17.
15. Nur Atiqah Sia Abdullah, Rusli Adbullah, Mohd Hasan Selamat, and Amzi Jaafar, "Extended Function Point Analysis Prototype with Security Costing Estimation," Proc. 2010 International Symposium in Information Technology (ITSim), June 2010, pp. 1297-1301, doi: 10.1109/ITSIM.2010.5561460.
16. Jihene Krichene, Noureddine Boudriga, and Sihem Guemara EI Fatmi, "SECOMO: An Estimation Cost Model for Risk Management Projects," CNAS Report, CNAS-2008-008, Jan. 2008.
17. Nick Xie, Nancy R. Mead, "SQUARE Project: Cost/Benefit Analysis Framework for Information Security Improvement Projects in Small Companies," Technical Report CMU/SEI-2004-TN-045, Nov. 2004.
18. Shawn A. Butler, "Security Attribute Evaluation Method: A Cost-Benefit Approach," Proc. 24th International Conference on Software Engineering (ICSE '02), May 2002, pp. 232-240.
19. Lawrence A. Gordon, and Martin P. Loeb, "The Economics of Information Security Investment," ACM Transaction on Information and System Security, Vol. 5, No. 4, Nov. 2002, pp. 438-457.
20. Samuel T. Redwine, Jr.(ed), "Secure Software Assurance-A guide to the common body of knowledge to produce, acquire, and sustain secure software," Draft Version 0.9, Jan. 2006.
21. Linda Ibrahim, Joe Jarzombek, Matt Ashford, et al., "Safety and Security Extensions for Integrated Capabiilty Maturity Models," Sep. 2004.
22. Information Assurance Technology Analysis Center (IATAC), and Data and Analysis Center for Software (DACS), "Software Security Assurance", State-of-the-Art Report, July, 2007.
23. Robert Richardson, "2008 CSI Computer Crime & Security Survey," 2008.
24. United States Government Accountability Office, "Infromation Assrance- National Partnership Offers Benefits, but Faces Considerable Challenges," GAO-06-392, March, 2006.
25. Dan Wu. Security Functional Requirements Analysis for Developing Secure Software. Dissertation, USC. May 2007.
26. http://www.abssw.com/papers/Cost_Estimation_for_Secure_Software_and_Systems.pdf
27. Yi Mao, Xiaohua Chen, and Yan Liu, "Comparative Study Between the Chinese Standards and the Common Criteria", in the proceedings of the 12th ICCC, Kuala Lumpur, Malaysia, 2011.
28. http://www.isccc.gov.cn/zsgg/07/743245.shtml
29. Mei He, Mingshu Li, Qing Wang, Ye Yang, Kai Ye: An Investigation of Software Development Productivity in China. ICSP 2008: 381-394