Journal of Software Engineering and Applications, **2012, 5, 545-559**
http://dx.doi.org/10.4236/jsea.2012.58063 Published Online August 2012 (http://www.SciRP.org/journal/jsea)

Simulation-Based Evaluation for the Impact of Personnel Capability on Software Testing Performance

Jithinan Sirathienchai[1], Peraphon Sophatsathit[2], Decha Dechawatanapaisal[3]

[1]Technopreneurship and Innovation Management, Graduate School, Chulalongkorn University, Bangkok, Thailand; [2]Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok, Thailand; [3]Department of Commerce, Faculty of Commerce and Accountancy, Chulalongkorn University, Bangkok, Thailand.
Email: jithinan.s@student.chula.ac.th, peraphon.s@chula.ac.th, decha@acc.chula.ac.th

Received May 3rd, 2012; revised June 6th, 2012; accepted June 15th, 2012

ABSTRACT

This study presents a decision making process in three steps of knowledge management for test organization using process simulation and financial analysis. First, project effort cost assessment of test knowledge management process subjects to different project duration and number of staffs is established. Two knowledge management simulation models representing experienced personnel with knowledge sharing and inexperienced personnel with internal training respectively are employed to contrast test personnel capability. Second, performance evaluation of software testing process by different personnel capability is conducted to simulate system test using three project metrics, namely, duration, effort cost, and quality. Third, a comparative financial analysis is prepared to determine the best solution by return on investment, payback period, and benefit cost ratio. The results from three stages of finding are discussed to arrive at the final scenario. We provide a case study evaluating how software testing industry needs to build effective test organization with high quality personnel for sustainable development and improvement.

Keywords: Software Testing Process; Knowledge Management; Test Organization; Process Simulation; Personnel Capability

1. Introduction

This paper focuses on software testing process improvement by knowledge management and human development. The study stems from the needs to build quality test people and strong test organization. A case study is taken from Small and Medium-sizes Enterprise (SME) software testing industry whose founders are test experts. From long experience, they found problems about personnel capability that affects software testing performance, such as missing of high severity defects detection, lacking test technique skills, and unclear understanding of test related business processes. These problems constitute the cause of low performance in software testing projects. Lengthy test duration, high cost, undetected defects passed on to user acceptance test and production phases.

The research objective establishes directions as to where the company should start resource investment, select target tester group, and improve test personnel capability. Current dilemma unveils that most testers come from developers. Worse yet some projects employ the same development and testing personnel due to limited resources. From our preliminary study with academic software testing activities, we found that the software testing subject in most universities is part of software engineering courses. A few institutions provide extended testing courses for direct and continuous studies. Moreover, new graduates cannot do well when they start testing work in company because the classroom environment is different from industrial practices.

The case study was carefully organized emphasizing on two knowledge management options with new staffs. The first option is knowledge sharing of experienced personnel who have worked in software development, such as programmer, system analyst, and database administrator. The second option is internal training of inexperienced personnel or new graduates. For this research, we define the work duration for experienced personnel to be 2 - 3 years and 0 - 1 year for inexperienced personnel. The research objective will be carried out as follows:

1) Estimate project costs of knowledge management for both options using different number of participants and project duration.

2) Evaluate performance factors (duration, effort cost, and quality) of software testing project obtained from

different personnel capability.

3) Select the best solution by comparing outputs from first and second items in terms of financial comparative analysis.

We apply simulation to solve research issues by implementing two models for knowledge sharing and internal training processes, and one model for the System Test (ST) which is used in the case study. The rationale for software process simulation is to enable forecasting or predicting future changed situation under various conditions such as factors, variables, activities, and assumptions [1,2]. Typical benefits are to understand and answer problems, to make decision on multiple alternatives, to estimate project performance, and to reduce the project risks [3-5]. Software process simulation is becoming increasingly widespread acceptance in academia and software industry, for example, Process Simulation Modeling (PSIM) of well-known framework, in particular, CMMI can be applied to support several process areas such as Causal Analysis and Resolution of level 5, Organizational Process Performance (OPP) of level 4, Decision Analysis and Resolution (DAR) of level 3, Project Planning (CAR) of level 2. Examples of PSIM in process improvement are cost estimation, qualitative process monitoring and control, and process optimization [6-9].

We are interested in some output parameters of referenced key performance factors, namely, duration, effort cost, and quality [10-13]. Total test duration in hours of duration factor and total test effort costs of effort human factors are investigated. The output parameters of quality factor are total number of defects, number of corrected defects, number of outstanding defects; number of old defects from User Acceptance Test (UAT), and number of new defects from UAT. The last two parameters of quality factor may occur from failure of development team due to version control and the defects in view of business. However, testers need to help support both development team and user team for quality assurance. At present, most defects should be detected in ST or previous phases are frequently found in UAT instead.

The organization of this paper is as follows. Section 2 describes the concepts of test personnel capability and software testing performance. Section 3 presents process simulation modeling. Section 4 contains the simulation experiment describing input and output parameters, scenarios, and assumptions. Section 5 summarizes simulation results of the case study in three stages. Section 6 concludes with some final thoughts.

2. The Personnel Capability and Software Testing Performance

Software defects can occur in every software development phase from several causes, for example, miscommunication, software complexity, poor documentation, and less skillful resources [14]. The last cause is apparently people cause, including overloading of resources and unprofessional attitude. Low quality developers can create big defects while low quality testers cannot detect whole defects. Increasing defects from development team and remaining defects from test team affect the project cost and duration in the same direction, while affecting the product and project quality in the inverse direction [15]. Despite the fact that testing and correcting can work in every software development process, the cost and duration of finding and correcting these failures increase dramatically [16].

A test team consists of several roles which encompass different responsibilities and skills. For example, a test manager needs developing test strategy and planning the overall test project skills, while a tester should develop proficient software test method and process skills, including design of test cases, executing test with proper test techniques, etc. [17,18]. Furthermore, their responsibilities are different in each organization due to different organizational structures. In general, test people should possess three main knowledge areas, namely, application or business domain, technology, and basic and advance test techniques [19].

This personnel capability in the test team can exceedingly impact on software testing performance and leads to success or failure of the project [17]. A quality tester with high capability can help project save a lot of cost and time. Moreover, they exert high impact on the quality of product and process delivered. To improve testing personnel quality, the executives have to continuously support them for knowledge management by knowledge sharing, internal training, and external training [20,21].

Some researchers have proposed solutions to improving test and knowledge management such as new test framework, customized test tools, test center, and test performance measurement [22,23]. One researcher incorporates knowledge management strategies in Software Process Improvement (SPI) project at first step [24]. The build of an effective software testing organization need to recruit or develop the right people with right skills and provide continuing education for them [21].

Unfortunately, in a real situation, many test people lack direct testing and related skills because of inadequate knowledge management. Exploitation of the accrued experience by way of knowledge management became an important issue [25,26]. The shortfalls are typically resulted from inefficient knowledge transfer, managing process, people, and organization conflicts [27-29]. Most organizations are well aware of inadequate sharing and training that affect knowledge improvement. Moreover, the cost incurred plays a significant role to reduce or reject the needed knowledge management. They are not confident in the value of knowledge management investment that could return in software testing performance terms. To strive for high performance of software development and testing, competent developers and testers are indispensably needed. We can conclude that knowledge management can improve people and qualified staffs can improve project performance.

3. The Process Simulation Modeling Approach

Two main purposes are deployed in three simulation models developed that focus on small-medium scale software testing projects. The first two models are knowledge sharing and internal training processes, hereafter referred to as option A and B respectively, that support the first purpose to assist the director in making decision on human resource development. The last model is software testing process to investigate the effect of different test personnel capability on various performance factors. The three models were implemented using the discrete-event simulation (DES). The DES is the general and powerful paradigm that is widely used and easy to represent process activities, queues, schedules, specific discrete items, and attributes [30-33]. We selected Arena, a discrete-based commercial simulation modeling package consisting of several applications such as business process reengineering and workflows, and supply chain management [30,34].

3.1. Option A: Experienced Personnel with Knowledge Sharing Model

This knowledge sharing model is based on preceding related researches on the aspects of knowledge management that are related to software engineering and software testing [35-38]. Four main processes or activities will start after knowledge needs are identified as illustrated in **Figure 1**.

The first process is “Identify Knowledge Need” that the top management who plans organizational strategy will identify areas of knowledge which in accord with testing policies and goals. Next process is “Create Knowledge” that gathers a new knowledge or keeps the existing knowledge. The third process is “Storage Knowledge” for systematic analysis and extraction of knowledge, including allocation of storage methods and tools for subsequent used. To ensure that completely stored and maintained knowledge is in standard format representing accurate and reliable information, “Organize Knowledge” is conducted at the next process. The last is “Share Knowledge” that precedes any application of knowledge for a particular purpose that needs to focus on knowledge distribution and transfer in the assigned groups.

3.2. Option B: Inexperienced Personnel with Internal Training Model

For inexperienced personnel or new college graduates, internal training is important to keep them up to speed with the qualified staff. A typical area of study is software engineering training [39-41]. The internal training model consists of seven main processes as illustrated in **Figure 2**.

For training analysis phase, the “Analyze Instructional Content” is the first process after finished need assessment. This process defines the content of training. Both “Design Training Organization” and “Design Curriculum” are also a part of training design phase which in essence addresses the method of test organization structure and aligns its people and resources. The next process

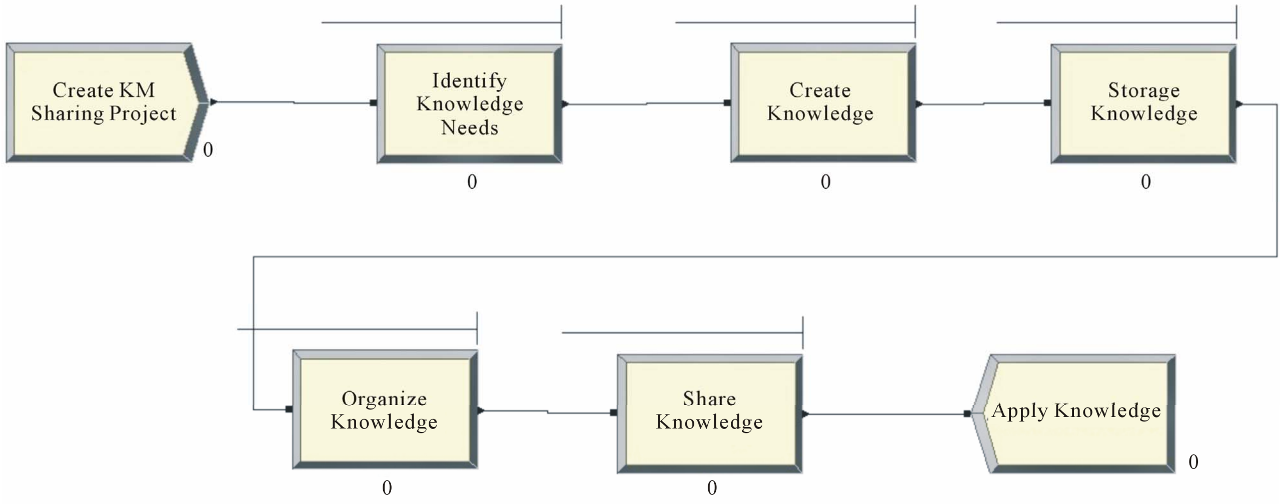

Figure 1. The knowledge sharing model.

is "Develop Materials" of training development phase for trainers, trainees, and participants, for example, checklist, presentation, assignment sheets, and questionnaires. The implementation phase of training model is "Conduct Training" which transfers specific knowledge from prepared materials. After finished training, "Evaluate Training Course" measures the effectiveness of the training program. The last process is "Feedback to Improve Training" that is a part of training evaluation phase to report success or failure of each training activity for continual training enhancement.

3.3. Software Testing Process Model

The software testing process model is based on the traditional V-Model or V diagram of software development and testing process [42]. However, this case study limits the scope to system test (ST) process because many projects will exclude integration test (IT) and system integration test (SIT) due to time constraint. The proposed model comprises of four major phases, namely, test planning and test preparation as shown in **Figure 3**, test execution as shown in **Figure 4**, and test report as shown in **Figure 5**.

Test planning is created at the start of the simulation while the test team assesses test requirements with the development team. After test assessment, the process will parallelly create a master test plan and test preparation phase which consists of create test scenario, create test case, prepare test data, and prepare test environment. For test execution phase, the test team will determine the number of test cycles. Usually many projects run two test cycles imposing by time and cost-limiting conditions. Execution of test cases and related activities are performed in accordance with the above test plan and strategy. Moreover, defect tracking, recording, and reporting are also included in this execution phase. All processes will stop when the test cases are completely performed or test deadline from scheduled plan release is reached. The test summary report will be created during test report phase before sending the report along with related documents to the customer for UAT.

For this case study, some resources of the test team are set aside to support UAT, for example, detailed explanations of some defects from ST, support of an acceptance test plan in developing and executing activities. Both AS-IS and TO-BE adopt the same model because the main process does not change. Minor adjustments are

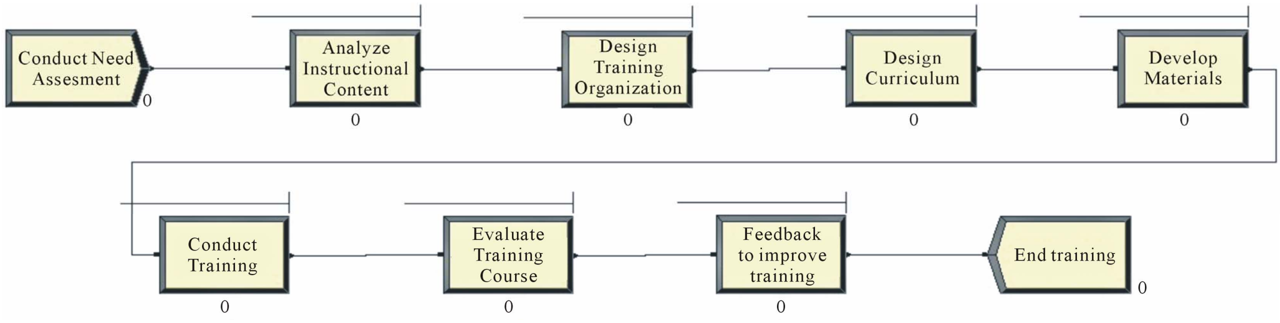

Figure 2. The internal training model.

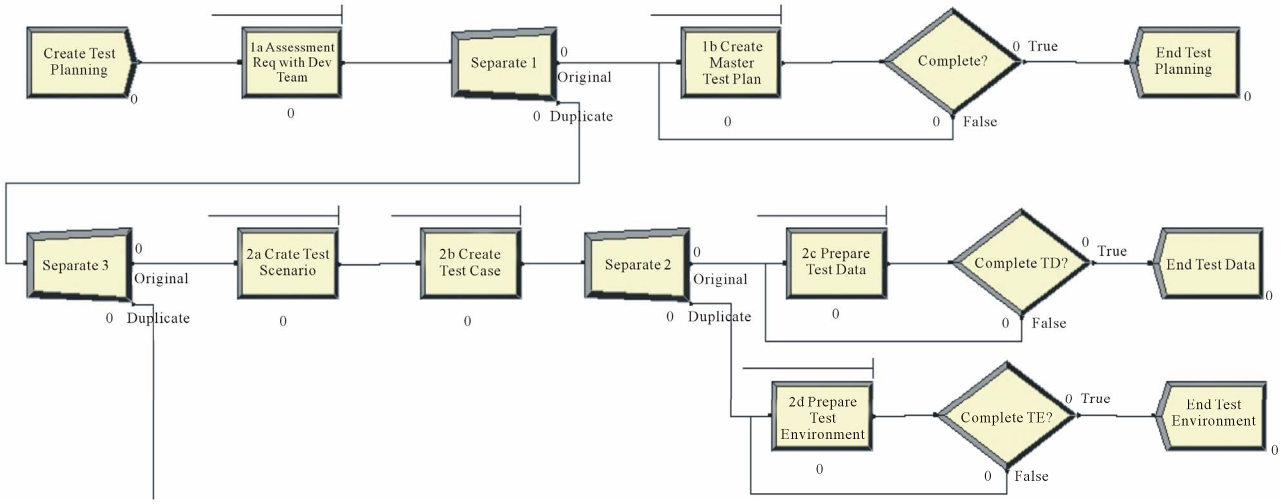

Figure 3. Software testing process model: test planning and preparation phases.

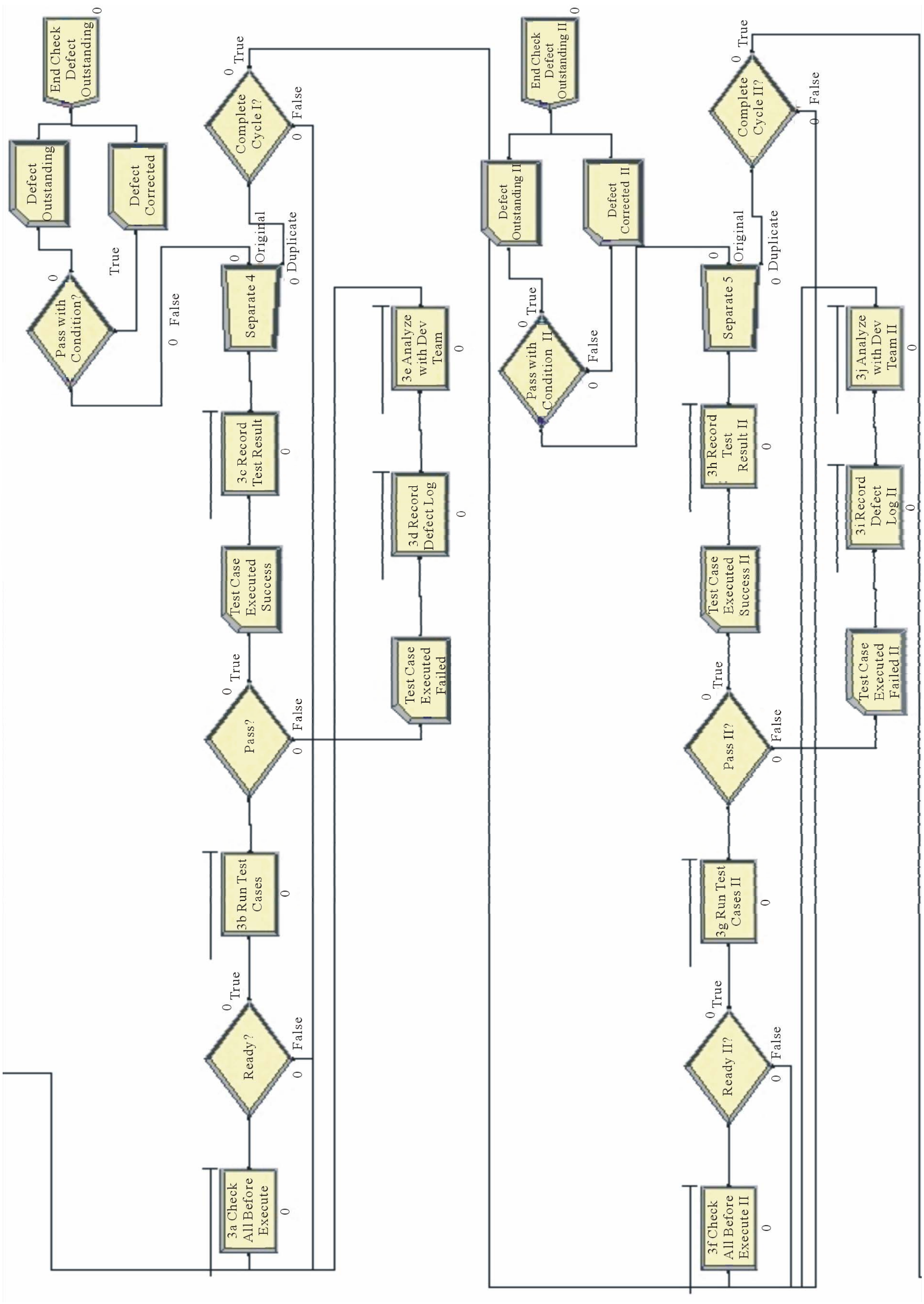

Figure 4. Software testing process model: test execution phase.

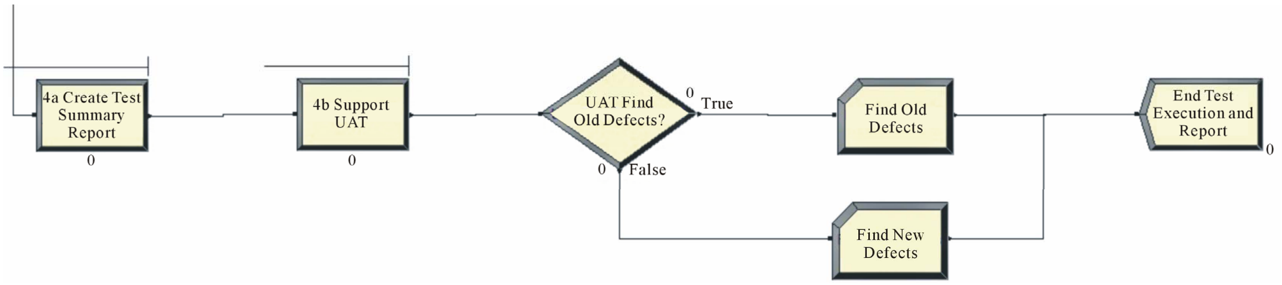

Figure 5. The software testing process model: test report phase.

required to accommodate different personnel capability, in particular, input parameters used for defect injection and process duration. Details will be elaborated in the next section.

4. Simulation Experiments

This section describes the simulation experiment of each model in three procedural groups, namely, input and output parameters, simulation scenarios, and model assumptions.

4.1. Input and Output Parameters

The input parameters are assigned at the invocation of simulation, denoting specific information assigned in each process. Input data are collected from historical records, experienced estimation, or pilot test. The output parameters are values obtained from the simulation that represent key performance variables.

4.1.1. Options A and B

The value of major input and output parameters in both models are shown in **Table 1**.

Knowledge management projects are randomly generated where the time between consecutive arrivals is exponentially distributed. Five process times of option A and seven process times of option B are triangular probability distributed over the minimum, mode, and maximum value. Two variables representing staff and other resource consumed are established to be the input parameters. The staff parameter refers to capacity and cost per hour of all related staffs such as test manager, test lead, and tester. The resource parameters represents essential materials needed in each process and cost per hour of staff parameter to acquire those materials, thereby total efforts can be determined. For output parameters, a key measure of project costs are compared between option A and option B, in conjunction with the key performance outputs of software testing process model, to determine the optimal ROI.

4.1.2. Software Testing Process Model

The process performance measures for this model are effort cost, schedule, and quality can be derived from the parameters as shown in **Table 2**.

Table 1. Major input and output parameters of options A and B.

#	Type	Parameter name	Unit
1	Input	Arrival rate	days
2		Process time	days
3		Process staff	persons
4		Process resource	machines, documents,…
5	Output	Total project cost	baht

Software testing projects are generated in the same manner as knowledge management project. The process time offers several distributions to best fit a given scenario with the smallest estimation error. Available distributions are lognormal, triangular, uniform, gamma, beta, and weibull. Variables used in the simulation will increase personnel capability. The measure on defect detection capability of the team focuses on ST and UAT, while defect correction capability considers only ST. For output parameters, major seven results constrained by three key performance factors, namely, effort cost, schedule, and quality, will serve as indicative comparisons between simulation scenarios to be elaborated in the next sub-section.

4.2. Simulation Scenarios

This section describes simulation scenarios to select the optimal solution for option A and B models obtained by comparison of each scenario with the baseline performance with the help of the software test process model.

4.2.1. Options A and B

Each option has nine scenarios. The project duration and number of tester are used as the principal factors to affect the output parameters of each scenario. The simulation does not adopt AS-IS baseline model but compares scenarios in a pairwise fashion. They are listed in **Table 3**.

The case study focuses on three sets of project duration, *i.e.*, eight, sixteen, and twenty four months, and

Table 2. Major input and output parameters of software testing process model.

#	Type	Parameter name	Unit
1	Input	Arrival rate	days
2		Process time	hours
3		Process staff	persons
4		ST Defect detection	defects
5		ST Defect correction	defects
6		UAT Defect detection	defects
7	Output	Total test duration	hours
8		Total test effort cost	baht
9		Total number of defects	defects
10		Number of corrected defects	defects
11		Number of outstanding defects	defects
12		Number of old defects from UAT	defects
13		Number of new defects from UAT	defects

Table 3. Simulation scenarios of options A and B.

#	Scenarios	
	Project duration (months)	Number of tester (persons)
1(a)	8	5
1(b)	8	10
1(c)	8	15
1(d)	16	5
1(e)	16	10
1(f)	16	15
1(g)	24	5
1(h)	24	10
1(i)	24	15

three sets of number of tester, *i.e.*, five, ten, and fifteen persons. This information is estimated from expert experiences in industry to initialize the test scenario for small-medium enterprises.

4.2.2. Software Testing Process Model

The software testing process model has six scenarios of TO-BE model and one AS-IS baseline model. The AS-IS baseline model represents current situation with testers who have test experience around 1 - 3 years but do not receive continuous skill improvement from self-study or company training support. The value-change scenario of options A and B based on different project duration will serve as key performance factors. **Table 4** presents the prerequisite scenarios and effectiveness percentage of personnel capability and process time in comparison with the baseline. These starting values are estimated by test experts of the company who in turn may extract them from pilot tests or previous projects. The results show that experienced personnel with knowledge sharing are more capable of short-term implementation, whereas inexperienced personnel with internal training are more capable of long-term implementation. Positive capability values indicate that the testers of each scenario can detect more defects than the incumbent. Negative values, on the contrary, indicate otherwise. The positive process time specifies that the scenarios use more time than the baseline while the negative value denotes shorter process time.

The baseline and six alternative scenarios were compared to determine the optimal user's acceptance. In conclusion, we exercise the model's scenarios both AS-IS and TO-BE. The project case incorporates personnel capability and project duration associated with each scenario for simulation execution.

4.3. Model Assumptions

In this section, some assumptions and scope imposed on the simulation model are discussed below.

4.3.1. Options A and B

Both models assume the followings:

1) Only necessary resources are considered in project cost calculation.

2) All process run continuously without failure or intermission, for example, all staffs join from start to end of project.

3) Project cost computations focus on resources and process duration without process improvement.

4) All participants in the project will be working 2 hours per day and 10 hours per week.

Table 4. Simulation scenarios of software testing process model.

#	Scenarios		Capability (%)	Process time (%)
	Options	Project duration (months)		
2(a)	A	8	0	+15
2(b)	A	16	+30	+5
2(c)	A	24	+60	−10
2(d)	B	8	−30	+20
2(e)	B	16	0	+10
2(f)	B	24	+100	−15

5) Simulation evaluation is based on project cost only; no other impacts are taken into account.

6) The simulation run length is set at 6, 18, and 24 months as identified scenarios.

7) 150 simulation replications are conducted for all nine scenarios of both models.

8) The 95% confidence interval is employed throughout the result values.

4.3.2. Software Testing Process Model, the Model Is Deployed Based on the Following Assumptions

1) The selected project size is approximately 50,000 lines of code.

2) Project of staff consists of as one test manager, one test leader, and five testers.

3) The experiments are designed emphasizing on human resource only, no other resource is considered.

4) All participants in the project will be working 8 hours per day and 40 hours per week.

5) All process run continuously without failure or intermission, for example, all staffs work from start to end of project.

6) Key performance computations focus on personnel capability under no process improvement.

7) Simulation evaluation is based on effort cost, duration, and quality only; no other impacts are taken into account.

8) The simulation run length is chosen to be 12 months.

9) 150 simulation replications are conducted for all baseline and six scenario combinations.

10) The 95% confidence interval is employed throughout the result values.

5. Dichotomy of Simulation and Its Results

The breakdown of simulation encompasses three stages as shown in **Figure 6**. The first stage determines the project costs of option A and B models in accordance with the prescribed nine scenarios. The second stage applies three key performance factors to the selected baseline and six scenarios of the software testing process model. In so doing, simulation output mean, standard deviation (SD), and standard error (SE) are obtained.

The results precipitated from the preceding two stages are used in the decision making process. The second step computes project ROI, payback period, and benefit cost ratio to arrive at the most viable financial decision. The following subsections elucidate the analyses of simulation results in all three stages.

5.1. Stage I: Simulation Results from Options A and B

Table 5 shows the simulation results from options A and B in nine scenarios under the assumption of both models.

The results show that project cost spent on option A will continuously increase. In contrast, option B slowly increases when project duration is cumulative. In two years of experiment, option B exhibits the lower costs than option A (1.62% for scenario 1(g), 24.77% for scenario 1(h), 34.30% for scenario 1(i)), while the number of testers has small impact on the project cost for option B but high impact for option A. These results indicate that labor costs have higher effect than other resources. Some contrasting statistics of both options show the lowest cost was resulted from five testers on eight months scenario (240181.24 and 418590.19), whilst the

Table 5. Simulation results from options A and B.

Scenario	Option					
	Option A			Option B		
	Mean	SD	SE	Mean	SD	SE
1(a)	240181.24	29570.07	2414.38	418590.19	63375.24	5174.56
1(b)	386613.99	51664.00	4218.34	543457.53	82181.69	6710.10
1(c)	538948.16	74648.87	6095.05	672396.16	102405.56	7045.04
1(d)	490162.84	34069.82	2781.78	674202.32	46831.42	3823.76
1(e)	788969.57	61439.28	5016.49	842918.62	59272.27	4839.56
1(f)	1100101.87	89672.29	7321.71	1000314.28	71269.99	5819.17
1(g)	734966.70	38467.71	3140.87	723247.40	38831.58	3170.58
1(h)	1185375.45	75263.47	6145.23	950076.42	52986.75	4326.35
1(i)	1652599.32	110227.82	9000.06	1230562.71	69916.22	5708.63

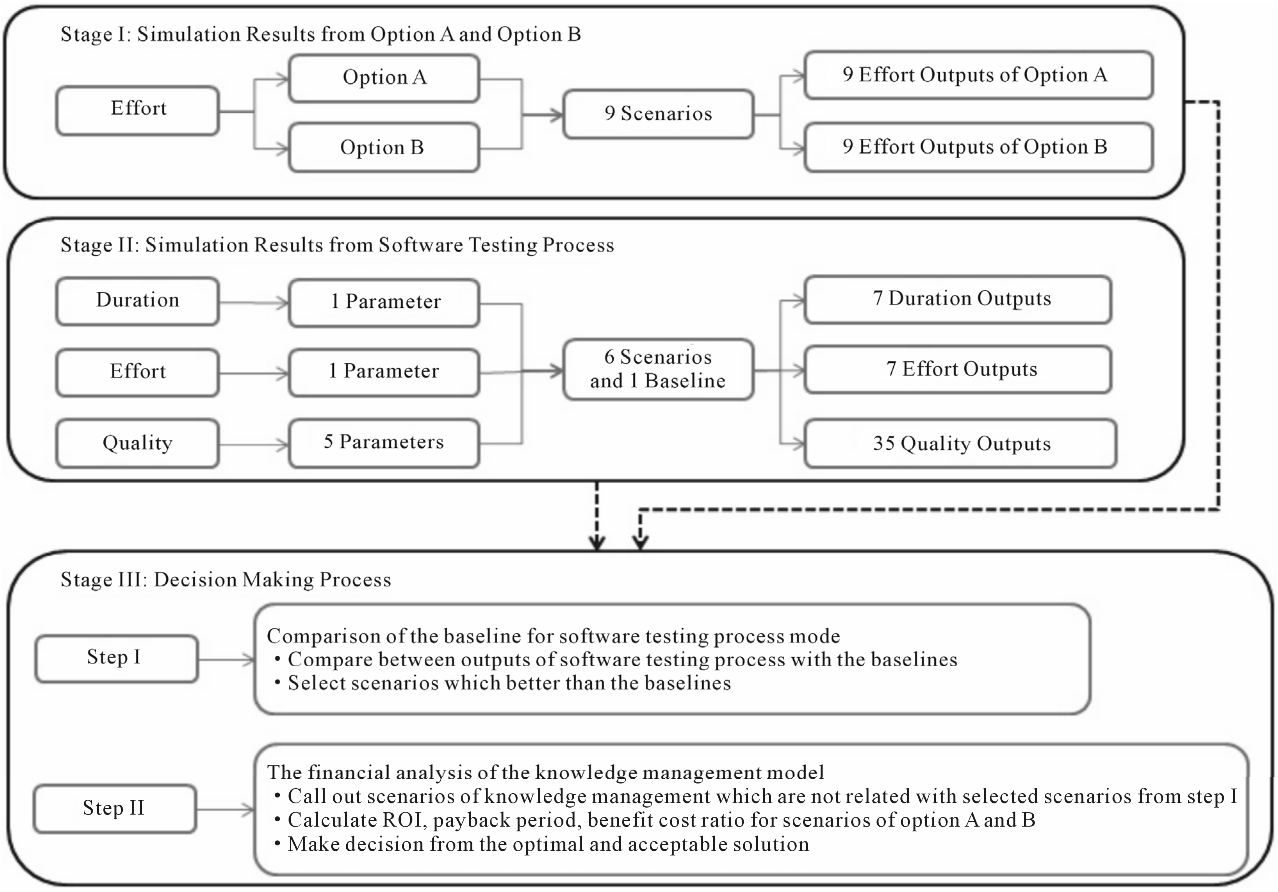
Stage I: Simulation Results from Option A and Option B
Effort
Option A
Option B
9 Scenarios
9 Effort Outputs of Option A
9 Effort Outputs of Option B
Stage II: Simulation Results from Software Testing Process
Duration
Effort
Quality
1 Parameter
1 Parameter
5 Parameters
6 Scenarios and 1 Baseline
7 Duration Outputs
7 Effort Outputs
35 Quality Outputs
Stage III: Decision Making Process
Step I
Comparison of the baseline for software testing process mode
• Compare between outputs of software testing process with the baselines
• Select scenarios which better than the baselines
Step II
The financial analysis of the knowledge management model
• Call out scenarios of knowledge management which are not related with selected scenarios from step I
• Calculate ROI, payback period, benefit cost ratio for scenarios of option A and B
• Make decision from the optimal and acceptable solution

Figure 6. The stages of simulation.

highest cost was fifteen testers on twenty four months scenario (1652599.32 and 1230562.71).

5.2. Stage II: Simulation Results from Software Testing Process

Three key performance factors made up the set of simulation results under one baseline and six scenarios of software testing process model.

5.2.1. Duration

Four project duration statistics measured in hours are shown in **Table 6**.

The baseline duration of software testing process is 1728.94 hours. The differences in knowledge management approach, same in project duration on two years of scenario 2(c) and 2(f) can help to reduce the time to 131.37 and 252.30 hours. The remaining four scenarios 2(a)/2(d) and 2(b)/2(e) of option A and B take more work time than the baseline measure. The best solution is scenario 2(f), *i.e.*, inexperienced personnel with internal training for two years. Hence, the duration of software testing project will depend on how long of knowledge management project training and sharing last. As a consequence, the better project time of software testing will be affected by the amount of time dedicated for knowledge management project.

5.2.2. Effort Cost

The effort cost is a measure of effectiveness factor for investment decision in any software projects across the software industry. The results are shown in **Table 7**.

Only one the highest scenario effort cost of 1805448.98, experienced personnel with knowledge management on eight months is higher than the baseline effort cost of 70736.52. The impact of personnel capability from two knowledge management ways on software testing process effort costs shows that three scenarios can reduce test effort cost as 2(e) of 15048.18, 2(c) of 161014.21, and 2(f) of 336392.89. Inexperienced personnel with internal training is more cost effective than other option for both short and long term projects because of lower human costs.

5.2.3. Quality

Five output parameters of ST and UAT from the baseline and six scenarios are considered for software testing process performance as shown in **Table 8**.

Table 6. Duration: simulation results from software testing process.

Scenario	Mean	SD	SE
Baseline	1728.94	195.30	15.94
2(a)	1789.99	141.72	11.57
2(b)	1774.26	162.71	13.28
2(c)	1597.57	223.26	18.22
2(d)	1797.24	133.07	10.86
2(e)	1768.46	168.42	13.75
2(f)	1476.64	229.25	18.71

Table 7. Effort cost: simulation results from software testing process.

Scenario	Mean	SD	SE
Baseline	1734712.46	232670.71	18997.48
2(a)	1805448.98	172581.73	14091.23
2(b)	1783511.71	194977.92	15919.88
2(c)	1573698.25	252749.78	20636.93
2(d)	1753694.51	152633.36	12462.46
2(e)	1719664.28	194237.17	15859.39
2(f)	1398319.57	247262.74	20188.91

Four output parameters, namely, total number of defects and number of corrected defects are the sum value of two test cycles that show the results in same direction that mean scenario 2(b), 2(c), and 2(f) higher quality than the baseline.

The total number of detected defects increases when testers are in knowledge management project for a longer period time. The difference in defect detection of option A and B for eight, sixteen, and twenty four months are 377 (1007 of option A and 630 of option B), 564 (2036 of option A and 1472 of option B), respectively. The twenty four month scenario, in particular, unveils performance improvement of defect detection on experienced testers participating in internal training. Similar results of corrected defects are obtained. The number of outstanding defects of the baseline and six scenarios are insignificantly different because both output parameters depend on the proficiency of the development team.

The number of old defects in UAT shows that long-term scenarios, especially the twenty four month of knowledge management project, 16, and 18 of option A and B, respectively, has the most probable potential to reduce defects in UAT. Perhaps it takes some time for learning curve to sink in. As a consequence, the number of new defects from UAT becomes significant because

Table 8. Quality: simulation results from software testing process.

Scenario	Defect Parameters	Mean	SD	SE
Baseline	Total number	1550	913.38	74.57
	Corrected defects	1525	933.50	76.22
	Outstanding defects	25	42.86	3.49
	UAT-Old defects	66	170.53	13.92
	UAT-New defects	413	494.08	40.34
2(a)	Total number	1007	888.66	72.55
	Corrected defects	985	899.39	73.43
	Outstanding defects	23	40.30	3.29
	UAT-Old defects	73	261.55	21.35
	UAT-New defects	306	723.03	59.03
2(b)	Total number	2036	1135.49	92.71
	Corrected defects	2007	1161.44	94.83
	Outstanding defects	29	56.63	4.62
	UAT-Old defects	26	112.72	9.20
	UAT-New defects	393	490.12	40.01
2(c)	Total number	3381	1070.32	87.39
	Corrected defects	3357	1114.66	91.01
	Outstanding defects	24	58.20	4.75
	UAT-Old defects	16	62.56	5.10
	UAT-New defects	390	207.81	16.96
2(d)	Total number	630	592.08	48.34
	Corrected defects	613	599.62	48.95
	Outstanding defects	17	31.86	2.60
	UAT-Old defects	60	238.28	19.45
	UAT-New defects	266	682.14	55.69
2(e)	Total number	1472	893.00	72.91
	Corrected defects	1455	908.14	74.14
	Outstanding defects	17	37.46	3.05
	UAT-Old defects	46	145.93	11.91
	UAT-New defects	306	462.65	37.77
2(f)	Total number	4190	1269.02	103.61
	Corrected defects	4160	1325.18	108.20
	Outstanding defects	30	70.23	5.73
	UAT-Old defects	18	65.38	5.33
	UAT-New defects	430	174.07	14.21

other factors exert direct effect on the results such as capability of testers or users in UAT phase.

Analysis of the simulation results reaches three possible decisions, namely, 2(b), 2(c), and 2(f) as they yield higher quality than those from the baseline.

5.3. Stage III: Decision Process

The decision process is performed in two analysis steps. The first step is the comparison of output parameters from six scenarios with baseline based on software testing process model. The second step conducts financial analysis of the knowledge management model (options A and B) to determine the financial value of all selected scenarios obtained from the first step.

5.3.1. Step I: Baseline Comparison of Software Testing Process Model

In first step, four output parameters in duration, effort cost, and quality factors are used to compare with the baseline outputs as illustrated in **Figure 7**.

The selected solutions for each output parameter are summarized below.

Duration: total test duration of scenario 2(f) and 2(c) are the better solutions than the baseline since they can reduce the process time from 1728.94 to 1476.64 (14.59%) and 1597.57 (7.60%), respectively.

Effort Cost: Three scenarios of the total test effort cost save 2(f), 2(c), and 2(e), ranging from 1734712.46 to 1398319.57 (19.39%), 1573698.25 (9.28%), and 1719664.28 (0.87%), respectively.

Quality: For this factor, we only consider two output parameters that have directly effects on the outcome. The three highest effective defect detection values in ST and UAT are obtained from scenario 2(f), 2(c), 2(b). The total number of defect detection in ST increases from 1550 to 4190 (170.32%), 3381 (118.13%), and 2036 (31.35%), while the number of old defects in UAT reduces from 66 to 18 (72.72%), 16 (75.76%), and 26 (60.61%), for 2(f), 2(c), and 2(b), respectively.

The results show that scenario 2(c) and 2(f) are the preferable solutions because they are better than those of the baseline in all performance factors.

5.3.2. Step II: Financial Analysis of the Knowledge Management Model

The candidate scenarios obtained from the first step are

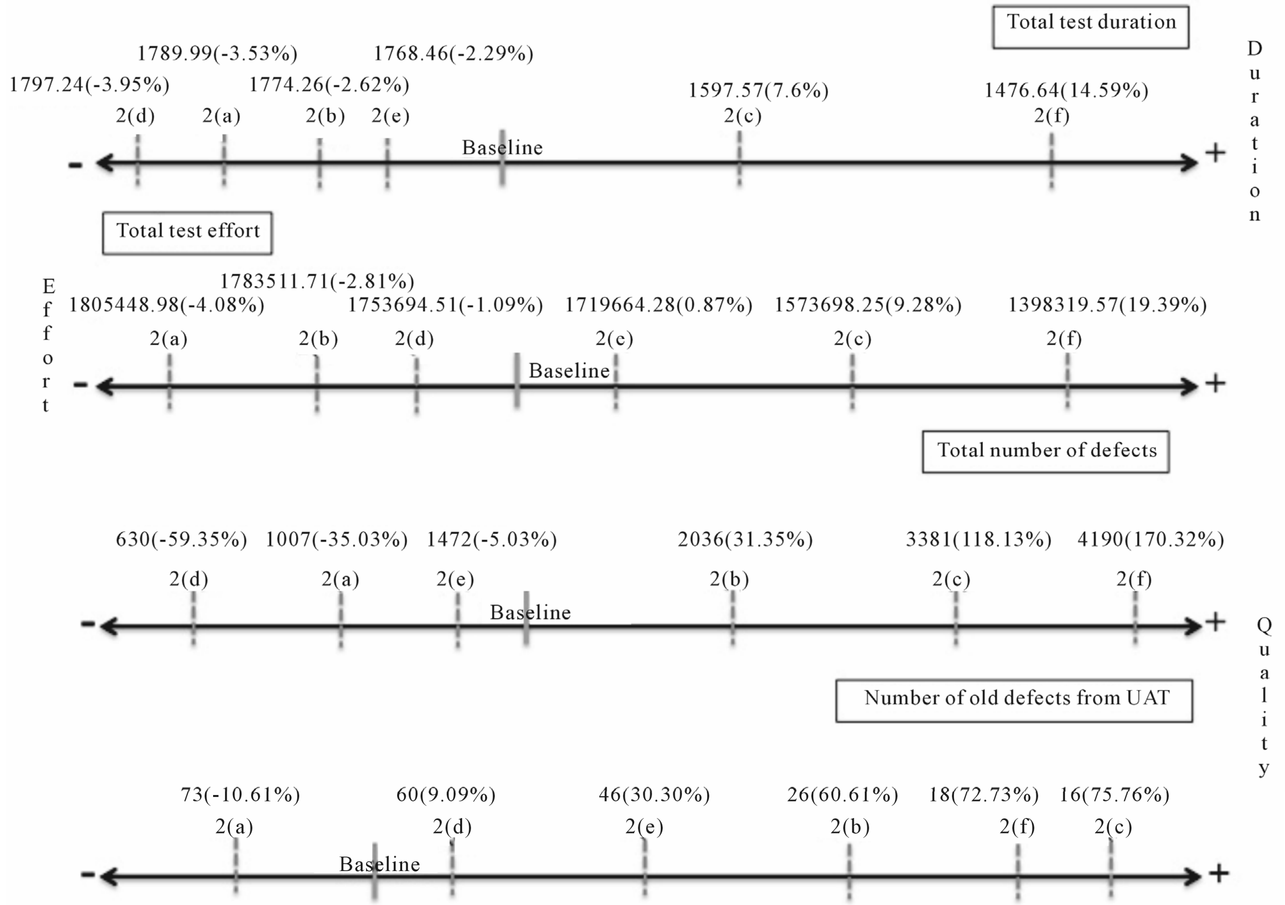

Figure 7. Comparison of the baseline for software testing process model.

taken into financial analysis consideration. They are the scenario 2(c) denoting experienced personnel with knowledge sharing that includes 1(g), 1(h), and 1(i), and the scenario 2(f) denoting inexperienced personnel with internal training that includes 1(g), 1(h), and 1(i).

The participating six knowledge management scenarios will be analysed using ROI, payback period, and benefit cost ratio as shown in **Table 9**.

ROI is calculated from Net Knowledge Management Project Benefit/Project Cost × 100. Payback period is calculated from Costs/Monthly Benefit, where Monthly Benefit value is 12 months for evaluation duration of this study. Benefit Cost Ratio is estimated from Project Benefit/Cost. The outcomes are labor cost reduction and productivity increase.

The highest ROI is 33% of 2(f)-1(i) scenario. In one year of evaluation duration, the case study organization gains on investment from all scenarios of knowledge sharing with inexperienced personnel while loss on the investment from all scenarios of internal training with experienced personnel. There is no significant difference when the number of staffs in internal training project increases as they have little effect on the project cost. The lower human cost of inexperience has higher value of project benefits. Furthermore, the results from software testing process model show that internal training project for the 24 month scenario has twice the productivity increase in the knowledge sharing project. For payback period, all scenarios of internal training project can return benefit within one year. The selection by the decision makers in lieu of financial results yields the investment in internal project with 15 inexperienced persons.

The final solution is 2(f)-1(i) since the difference in performance of AS-IS and TO-BE based on the three main parameters is optimal as shown in **Table 10** and **Figure 8**. Note that the dispersion of total number of defects is uniformly distributed; thereby no congregation of outputs can be formed.

The wide span of result box indicates the high risk of volatility, potential for unpredictable results, as well as considerable impacts on project outcomes. One can see the potential impact of requirement volatility on key project management parameters such as effort cost, schedule, and quality. The differences between baseline and requirement volatility case can be attributed to the addition (or reduction) in project scope caused by requirement volatility, cause and effect relationships between factors, and the stochastic distribution of a number of the factors.

6. Conclusions

An important basis of software testing process improvement is test personnel development. There are many human development approaches to administer knowledge management. This industrial simulation case study considers two ways of improvement, *i.e.*, knowledge sharing for experienced personnel and internal training for inexperienced personnel.

To evaluate the impact of personnel capability on software testing process performance, we employed simulation models of knowledge management based on nine scenarios to determine the project cost. The resulting knowledge improvement was further exploited by simulation to investigate personnel capability using duration,

Table 9. The financial analysis of the knowledge management model.

Scenario	Total Cost (Baht)	Total Benefit (Baht)	ROI (%)	Payback Period (Months)	Benefit Cost Ratio
2(c)-1(g)	734966.70	651014.21	−11	13.55	0.89
2(c)-1(h)	1185375.45	902028.42	−24	15.77	0.76
2(c)-1(i)	1652599.32	1153042.63	−30	17.20	0.70
2(f)-1(g)	723247.40	746392.89	3	11.63	1.25
2(f)-1(h)	950076.42	1192785.78	25	9.56	1.25
2(f)-1(i)	1230562.71	1639178.67	33	9.02	1.33

Table 10. Performance difference between AS-IS and TO-BE.

Performance Measure	AS-IS Process	T0-BE Process	Difference
Total test duration	1728.94 hours	1476.64 hours	252.30 hours
Total test effort	1734712.46 baht	1398319.57 baht	336392.89 baht
Total number of defects	1550 defects	4190 defects	2640.33 defects

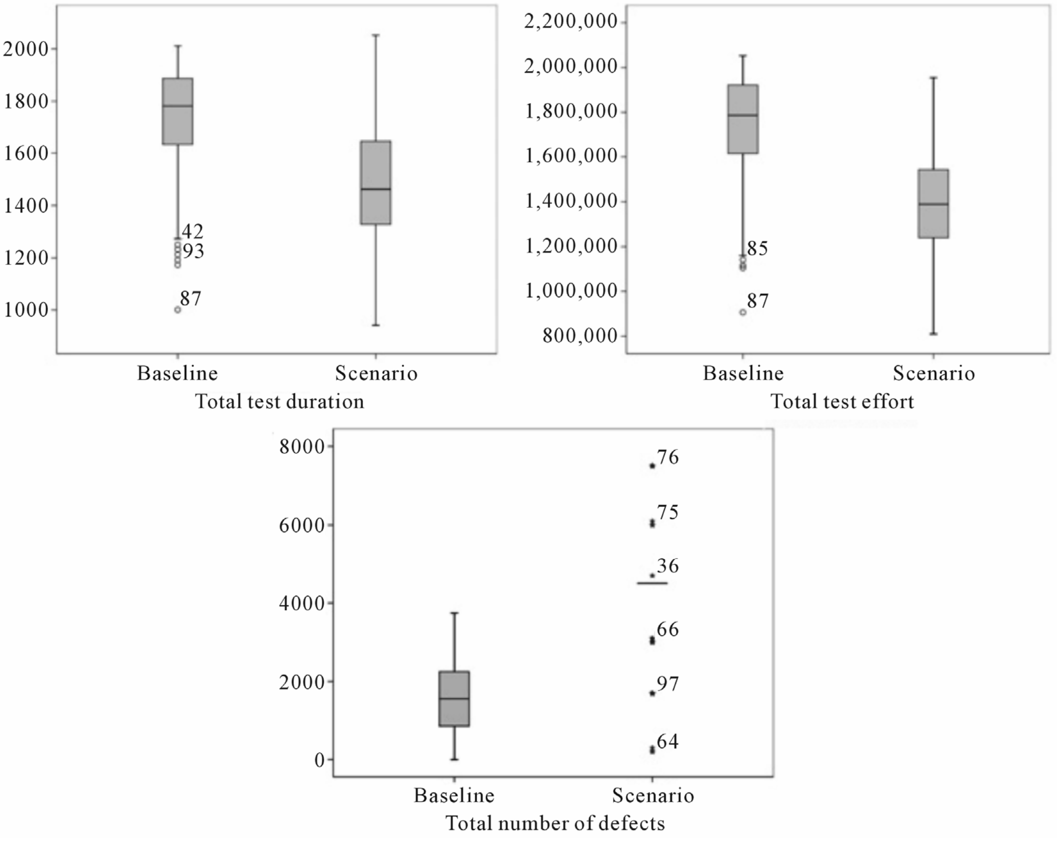

Figure 8. The output parameters box plots.

effort cost, and quality as the measurement yardsticks. The final decision is decided by standard financial indicators to arrive at the optimal business investment choice.

The findings from the simulation evaluation reveal that long-term continuous investment on knowledge management can improve the key process performance more effective than the short-term counterpart. Despite the high costs of internal training of inexperience staffs in the first year, they steadily decrease in subsequent years, as well as effort cost and duration of software testing process. In contrast to cost reduction in knowledge sharing of experience staffs, the quality of software testing process gradually increases due to accumulated experience. The decision maker can decide the best solution to suit organization strategy for the benefits of business goal.

Simple and straightforward as they are, these simulation models help support test teams to easily understanding and further their personal development. Future work will investigate addition factors in the simulation, for example, human factor aspects in behavior, characteristic, and organization factors as in turnover rate and performance measurement.

7. Acknowledgements

This study was financially supported by The 90th Anniversary of Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund), Chulalongkorn University. The authors wish to acknowledge all interviewees of the case studies for their invaluable inputs and comments.

REFERENCES

[1] M. I. Kellner, R. J. Madachy and D. M. Raffo, "Software Process Simulation Modelling: Why? What? How?" *Journal of Systems and Software*, Vol. 46, No. 2-3, 1999, pp. 91-105. doi:10.1016/S0164-1212(99)00003-5

[2] W. Scacchi, "Understanding Software Process Redesign using Modeling, Analysis and Simulation," *Software Process Improvement and Practice*, Vol. 5, No. 2-3, 2000, pp. 183-195. doi:10.1002/1099-1670(200006/09)5:2/3<183::AID-SPIP115>3.0.CO;2-D

[3] W. Wakeland, S. Shervais and D. Raffo, "Heuristic Optimization as a V&V Tool for Software Process Simulation Models," *Software Process Improvement and Practice*, Vol. 10, No. 3, 2005, pp. 301-309. doi:10.1002/spip.231

[4] M. Melis, I. Turnu, A. Cau and G. Concas, "Evaluating the Impact of Test-First Programming and Pair Programming through Software Process Simulation," *Software Process Improvement and Practice*, Vol. 11, No. 4, 2006,

pp. 345-360. doi:10.1002/spip.286

[5] I. Rus, J. Collofello and P. Lakey, "Software Process Simulation for Reliability Management," *International Journal of Simulation*, Vol. 46, No. 2-3, 1999, pp. 173-182.

[6] D. M. Raffo and W. Wakeland, "Moving Up the CMMI Capability and Maturity Levels Using Simulation," *Technical Report CMU/SEI*-2008-*TR*-002, *ESC-TR*-2008-002, Carnegie Mellon University, Pittsburgh, 2008.

[7] D. M. Raffo, R. Ferguson, S. Setamanit and B. Sethanandha, "Evaluating the Impact of Requirements Analysis Tools Using Simulation," *Software Process Improvement and Practice*, Vol. 13, No. 1, 2008, pp. 63-73. doi:10.1002/spip.364

[8] D. M. Raffo, J. V. Vandeville and R. H. Martin, "Software Process Simulation to Achieve Higher CMM Levels," *Journal of Systems and Software*, Vol. 46, No. 2-3, 1999, pp. 163-172. doi:10.1016/S0164-1212(99)00009-6

[9] S. Setamanit, W. Wakeland and D.M. Raffo, "Using Simulation to Evaluate Global Software Development Task Allocation Strategies," *Software Process Improvement and Practice*, Vol. 12, No. 5, 2007, pp. 491-503. doi:10.1002/spip.335

[10] R. Martin and D. Raffo, "Application of a Hybrid Process Simulation Model to a Software Development Project," *Journal of Systems and Software*, Vol. 59, No. 3, 2001, pp. 237-246. doi:10.1016/S0164-1212(01)00065-6

[11] D. X. Houston, S. Ferreira, J. S. Collofello, D. C. Montgomery, G. T. Mackulak and D. L. Shunk, "Behavioral Characterization: Finding and Using the Influential Factors in Software Process Simulation Models," *Journal of Systems and Software*, Vol. 59, No. 3, 2001, pp. 259-270. doi:10.1016/S0164-1212(01)00067-X

[12] F. Stallinger, "Software Process Simulation to Support ISO/IEC 15504 Based Software Process Improvement," *Software Process Improvement and Practice*, Vol. 5, No. 2-3, 2000, pp. 197-209. doi:10.1002/1099-1670(200006/09)5:2/3<197::AID-SPIP120>3.0.CO;2-K

[13] S. Ferreira, J. Collofello, D. Shunk and G. Mackulak, "Understanding the Effects of Requirements Volatility in Software Engineering by Using Analytical Modeling and Software Process Simulation," *Journal of Systems and Software*, Vol. 82, No. 10, 2009, pp. 1568-1577. doi:10.1016/j.jss.2009.03.014

[14] A. Ahmed, "Software Testing as a Service," CRC Press, Boca Raton, 2010.

[15] G. Tassey, "The Economic Impacts of Inadequate Infrastructure for Software Testing," *Technical Report*, NIST, 2002.

[16] C. Kaner, J. L. Falk and H. Q. Nguyen, "Testing Computer Software," 2nd Edition, John Wiley & Sons, New York, 1999.

[17] E. Dustin, "Effective Software Testing: 50 Specific Ways to Improve Your Testing," John Addison-Wesley Longman Publishing, Boston, 2002.

[18] A. M. J. Hass, "Guide to Advanced Software Testing," Artech House, Norwood, 2008.

[19] D. Graham, E. V. Veenendaal, I. Evans and R. Black, "Foundations of Software Testing: ISTQB Certification," Thomson Learning, 2007.

[20] E. J. Weyuker, T. J. Ostrand, J. Brophy and R. Prasad, "Clearing a Career Path for Software Testers," *IEEE Software*, Vol. 17, No. 2, 2000, pp. 76-82. doi:10.1109/52.841696

[21] I. Burnstein, "Practical Software Testing: A Process-Oriented Approach," Springer Inc., New York, 2003.

[22] T. A. Majchrzak, "Best Practices for the Organizational Implementation of Software Testing," *Proceedings of the 43th Annual Hawaii International Conference on System Sciences*, Washington, 5-8 January 2010, pp. 1-10.

[23] T. A. Majchrzak, "Improving the Technical Aspects of Software Testing in Enterprises," *International Journal of Advanced Computer Science and Applications*, Vol. 1, No. 4, 2010, pp. 1-10.

[24] L. Mathiassen and P. Pourkomeylian, "Managing Knowledge in a Software Organisation," *Journal of Knowledge Management*, Vol. 7, No. 2, 2003, pp. 63-80. doi:10.1108/13673270310477298

[25] O. Taipale, K. Karhu and K. Smolander, "Observing Software Testing Practice from the Viewpoint of Organizations and Knowledge Management," *Proceedings of the 1st International Symposium on Empirical Software Engineering and Measurement*, IEEE Computer Society, Washington, 2007, pp. 21-30. doi:10.1109/ESEM.2007.18

[26] K. Karhu, O. Taipale and K. Smolander, "Investigating the Relationship between Schedules and Knowledge Transfer in Software Testing," *Information and Software Technology*, Vol. 51, No. 3, 2009, pp. 663-677. doi:10.1016/j.infsof.2008.09.001

[27] C. F. Cohen, S. J. Birkin, M. J. Garfield and H. W. Webb, "Managing Conflict in Software Testing," *Communications of the ACM*, Vol. 47, No. 1, 2004, pp. 76-81. doi:10.1145/962081.962083

[28] C. Lovin and T. Yaptangco, "Best Practices: Enterprise Test Management," Dell Power Solutions, 2006, pp. 37-39.

[29] T. Parveen, S. Tilley and G. Gonzalez, "A Case Study in Test Management," *Proceedings of the 45th Annual Southeast Regional Conference*, New York, 23-24 March 2007, pp. 82-87. doi:10.1145/1233341.1233357

[30] T. Altiok and B. Melamed, "Simulation Modeling and Analysis with Arena," Academic Press, San Diego, 2007.

[31] W. Scacchi, "Experiences with Software Process Simulation and Modeling," *Journal of Systems and Software*, Vol. 46, No. 2, 1999, pp. 183-192. doi:10.1016/S0164-1212(99)00011-4

[32] F. Padberg, "A Discrete Simulation Model for Assessing Software Project Scheduling Policies," *Software Process Improvement and Practice*, Vol. 7, No. 3-4, 2002, pp. 127-139. doi:10.1002/spip.160

[33] K. G. Kouskouras and A. C. Georgiou, "A Discrete Event Simulation Model in the Case of Managing a Software Project," *European Journal of Operational Research*, Vol. 181, No. 1, 2007, pp. 374-389.

doi:10.1016/j.ejor.2006.05.031

[34] E. M. O. Abu-Taieh and A. A. R. El Sheikh, "Commercial Simulation Packaged: A Comparative Study," *International Journal of Simulation*, Vol. 8, No. 2, 2007, pp. 66-76.

[35] A. Aurum, F. Daneshgar and J. Ward, "Investigating Knowledge Management Practices in Software Development Organizations—An Australian Experience," *Information and Software Technology*, Vol. 50, No. 6, 2008, pp. 511-533. doi:10.1016/j.infsof.2007.05.005

[36] K. Nogeste and D. Walker, "Using Knowledge Management to Revise Software-Testing Processes," *Journal of Workplace Learning*, Vol. 18, No. 1, 2006, pp. 56-27. doi:10.1108/13665620610641283

[37] C. Kerkhof, J. Ende and I. Bogenrieder, "Knowledge Management in the Professional Organization: A Model with Application to CMG Software Testing," *Knowledge and Process Management*, Vol. 10, No. 2, 2003, pp. 77-84. doi:10.1002/kpm.167

[38] T. Dingsøyr and R. Conradi, "A Survey of Case Studies of the Use of Knowledge Management in Software Engineering," *International Journal of Software Engineering and Knowledge Engineering*, Vol. 12, No. 4, 2002, pp. 391-414. doi:10.1142/S0218194002000962

[39] T. Lee, D. Baik and H. P. In, "Cost Benefit Analysis of Personal Software Process Training Program," *Proceedings of IEEE* 8*th International Conference on Computer and Information Technology Workshops*, Sydney, 8-11 July 2008, pp. 631-636.

[40] M. B. Carpenter, "Process Improvement for Software Engineering Training," *Proceedings of the* 9*th Conference on Software Engineering Education*, Daytona Beach, 21-24 April 1996, pp. 172-183. doi:10.1109/CSEE.1996.491371

[41] M. Carpenter and H. Hallman, "Training Guidelines: Creating a Training Plan for a Software Organization," *Technical Report CMU/SEI*-95-*TR*-007, *ESC*-*TR*-95-007, Carnegie Mellon University, Pittsburgh, 1991.

[42] W. E. Lewis, "Software Testing and Continuous Quality Improvement," 3rd Edition, Auerbach Publications, Boca Raton, 2009.