

Impact of Personnel Factors on the Recovery of Delayed Software Projects: A System Dynamics Approach

Mostafa Farshchi, Yusmadi Yah Jusoh, and Masrah Azrifah Azmi Murad

Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
43400 Serdang, Selangor, Malaysia
farshchi; yusmadi; masrah@fsktm.upm.edu.my

Abstract. Delay in a software project may result in the loss of a market opportunity or the postponement of a dependent project. Therefore, software project managers take various steps to ensure that their project is completed on time, such as adding new members to the project team. However, adding new manpower to a delayed project may cause a negative impact on the team's productivity due to assimilation time, training overhead and communication overhead. Consequently, project managers have difficulty in making the decision on whether or not to add new members to the team. Thus, this research aims to examine whether a significant schedule improvement can be achieved with consideration of the new manpower's capabilities, skills and experience. A System Dynamics Model is proposed to simulate the behaviour of a project's progress when new members are added. The proposed model was evaluated through experiments using two types of case studies. The results of the experiments indicate that a significant schedule improvement of a late project can be achieved if people with certain levels of personnel factors are added to the project.

Keywords: software project management, personnel factors, system dynamics, schedule delay.

1. Introduction

Despite recent advances in software project management technology and methods, project failure rates are still considerably high. Many software project failures are reportedly due to problems related to delayed delivery and deferred deployment [1; 2]. Late completion of a software development project cannot be tolerated because it may cause delay to a dependent project or result in the potential loss in profits due to a missed business opportunity. Time delay in such situations could incur a much higher cost than the cost of the project itself. Consequently, the management would undertake any possible measure to ensure that their project is delivered on time. One of the

common practices in dealing with schedule delays is adding new manpower [3; 4; 5; 6; 7].

Although adding new members to an ongoing software project team is anticipated to increase the available effort to the team, it may result in the loss of productivity due to assimilation delay, inter-personnel communication overhead, and training overhead. The above phenomenon was first explained by Frederick Brooks [8] and became known as Brooks' Law: "adding manpower to a late software project makes it later".

Accordingly, several studies have been reported to address the problem: the issue was investigated formally by Abdel-Hamid and Madnick [3]; in another study, an attempt to address the problem through mathematical expression was carried out by Stutzke [9]; later, an investigation of the impact of task constraints was performed by Hsia *et al.* [10]; and a study on the impact of pair programming was conducted by Williams *et al.* [11]. In addition, an exploration of the issue through a newly-structured System Dynamics approach was presented by Madachy [12], and some assessments of the impact of the issue on open-source projects were recently conducted [13; 14; 15; 16; 17].

As a result of the noted effects, a project manager faces difficulty in making the decision on whether or not to add new members to his team. Software development is a human-intensive process [18]. Several studies have reported the high contribution of personnel attributes on software development productivity, such as those carried out by Finnie *et al.* [19], Boehm *et al.* [20], Trendowicz and Munch [21], and Hannay *et al.* [22]. As highlighted above, scholars have studied the issue from various aspects. However, the effect of new manpower capability, skills and experience in addressing the problem has yet to be studied further. Therefore, this study tries to understand:

- How software project managers can minimise the negative effects of adding new manpower to delayed software projects using personnel factors trade-off analysis; and
- Whether a significant schedule improvement of a late project can be achieved by taking into consideration the new manpower capabilities, skills and experiences.

If a project manager wishes to add new members, then to succeed, the manager must look for ways to minimise the negative impact of communication overhead, training overhead, and assimilation delay. Hence, in this research, we attempt to assess how effectively the variety of personnel factors and capabilities can influence the progress of a late software project. Obviously, it's likely that a more skilful person would add more productivity to a project team than a less skilful one, but whether the difference would be significant enough to prevent schedule slippage or would just be a waste of resources is the concern of this study. This research aims to improve software project management by reducing the failure rate in meeting the schedule. Therefore, this study extends the work of previous research in both software process simulation and software project management in the following ways: Firstly, it proposes a System Dynamics simulation model that is capable of

effectively simulating the impacts of adding new manpower with distinct levels of skills, experience and capabilities to the progress of ongoing software projects. Secondly, it assesses the effects of the personnel factors of new manpower on selected factors through an examination of industry experiments, and provides an analysis of the benefits and drawbacks of considering the capabilities of new manpower added to ongoing projects. Thirdly, it provides a tool that empowers software project managers to become proactive in their decision making, enabling them to perform trade-off analysis and forecast the likely consequences of their decisions in altering manpower in a project.

The rest of this paper is organised as follows: In Section 2, related work in the study is presented and reviewed. Section 3 explains the approach taken in conducting the research and constructing the System Dynamics simulation model. Section 4 gives the results and discusses the findings of executing the proposed model in two case studies by comparing them with a reference model, and then validating the model against a real case study. Next, in Section 5, the model evaluation is discussed. In last section, we summarise our insights into the study and propose directions for future research.

2. Related Work

This section discusses the related literature concerning the impact of adding new manpower to delayed software projects.

2.1. The Impact of Adding New Manpower to Delayed Software Projects

As mentioned in the previous section, adding more manpower to a delayed software project would likely increase its completion time. This was explained as a consequence of the following three effects: (i) new staff need training and they need to be trained by experienced staff; (ii) there is an assimilation delay for new staff to become productive in a project; (iii) adding new people increases the communication overhead among the members of a project team. The above phenomenon is known as Brooks' Law, credited to years of industry experience [3; 23; 24; 25; 26]. However, the non-Brooks' Law phenomenon has also been reported by some researchers and organisations by utilising certain techniques [11; 25; 27; 28].

The study by Abdel-Hamid *et al.* revealed that adding more people to a late project did not always cause it to complete later. He assumed in his study that there is, in general, a desire among project managers to change the composition of their workforce. However, if the perceived remaining time is shorter than the anticipated hiring and assimilation delay, then the project manager would not add new manpower. Although the comprehensive software project model he proposed has been examined through a case

study, the effect of changing manpower has not been examined due to the limitations of the case study.

Stutzke [9] investigated the possible situations in which a project may benefit by adding new members to the workforce. Results derived from his mathematical model indicate that adding new staff would not necessarily have a negative impact on a project if certain constraints are taken into account. These constraints are: $r > (1 + m) * a$ and $f \leq 1/m$ where f = fractional increase in the staff; r = remaining time to complete project, a = assimilation time, and m = mentoring cost (fraction of staff member's time spent mentoring one new hire). Stutzke's model has one weakness, which is that it does not represent the communication overhead. His model also didn't take into account how the levels of capability and experience of the new hires can affect assimilation time and training needed.

In recent years, the effects of pair programming on software development productivity have gained more attention [11; 22; 29; 30; 31]. Williams *et al.* utilised Stutzke's mathematical equations to explore the relationship between pair programming and Brooks' Law. Their study implies that practising pair programming helps reduce assimilation delay and mentoring time. Although their result shows improvement in assimilation time and training overhead, the effect of pair programming hasn't been investigated on a project's overall communication overhead.

Through literature review, it is noted that human factors, such as lower programmer capability and insufficient experience, are the main causes of software project delays [32; 33]. For instance, the result obtained from analysing thirty-one causes of project delays by Ziya Ma [32] shows that "insufficient experience of developers" and "inappropriate access to development and test tools" are the most frequently reported causes of delay. Therefore, attempts have been made to further explore the effects of personnel factors, particularly their impact on the productivity levels of software projects.

In the field of software engineering, researchers have long acknowledged the importance of Brooks' Law. Results obtained from recent literature review indicate the growing trend of issues that were initially pointed out by Brooks [24; 25]. In recent years, the reinvestigation of Brooks' Law for projects of different types and characteristics has been suggested. For example, Callegari and Bastos [4] declared that "Brooks' Law needs more investigation when considering a project's characteristics, such as size and available time". Meanwhile, recent studies show, in contrast to Brooks' Law, that more programmers contributing to open-source software projects led to the resolution of difficult problems and the success of the software [13; 14; 15; 16; 17]. Schweik and English [13] are among those who argue that a higher number of people on free and open-source projects would contribute to the success of the project. The summary of approaches, factors and project types that have been studied before are presented in Tables 1 and 2.

Table 1. Summary of previous studies on adding new manpower to late software projects

Author	Approach	Main Parameters	Project Type
Abdel-Hamid and Madnick [3]	System Dynamics	Communication Overhead, Hiring Delay, Assimilation Time, Training Overhead	Commercial
Stutzke [9]	Mathematical Expression	Mentoring Time, Effective Assimilation Time	Commercial
Hsia <i>et al.</i> [10]	System Dynamics	Task Constraints, Overhead, Training Time, Assimilation Time,	Commercial
Williams <i>et al.</i> [11]	Mathematical Expression	Mentoring Time, Effective Assimilation Time, Pair Programming	Commercial
Madachy [12]	System Dynamics	Communication Overhead, Assimilation Time, Training Overhead	Commercial
Schweik and English [13]	Statistical	Inter-Personnel Communication Rate	Open-Source
Capiluppi and Adams [15]	Statistical	Inter-Personnel Communication Rate	Open-Source

Table 2. Main parameters of previous models on adding new manpower to late software projects

Author	New Personnel (NP) vs. Experienced Personnel (EP) Productivities	Assimilation time (workdays)	Training Time	Communication Overhead
Abdel-Hamid and Madnick [3]	EP = 1 task/ person-day NP = 0.5 task/ person-day	80	20.2 % of the time of an experienced person during assimilation time	$0.06 \times n^2$ Where n is the total manpower
Stutzke [9]	Not differentiated	33.4	25.0 % of the time of an experienced person during mentoring time	Not considered
Hsia <i>et al.</i> [10]	EP = 1 task/ person-day NP = 0.5 task/ person-day	80	20.2 % of the time of an experienced person during assimilation time	$0.06 \times n^2$ Where n is the total manpower
Williams <i>et al.</i> [11]	Not differentiated	27(with pair programming: 12)	37.0 % (with pair programming: 26.0 %) of the time of an experienced person during mentoring time	Not considered
Madachy [12]	EP = 1.2 function points/ person-day NP = 0.8 function points/ person-day	20	25.0 % of the time of a fulltime, experienced person during assimilation time	$0.06 \times n^2$ Where n is the total manpower

2.2. Personnel Factors

It was found through literature review that personnel factors, such as inadequate programmer capability and insufficient experience, are among the main causes of delays in software projects [32; 33]. Practitioners identify people as the main contributor of a project's success but project managers argue that although people are indeed key, their actions sometimes contradict their words [18]. Product complexity and personnel/team capability have been reported to have the highest influence on the development rate of a software project [20; 21; 22]. Trendowicz and Munch [21] identified 249 different factors that influence software development productivity through literature review. Their study shows that team capability and experience (personnel factors) were the most commonly cited factors. Their results are shown in Figure 1.

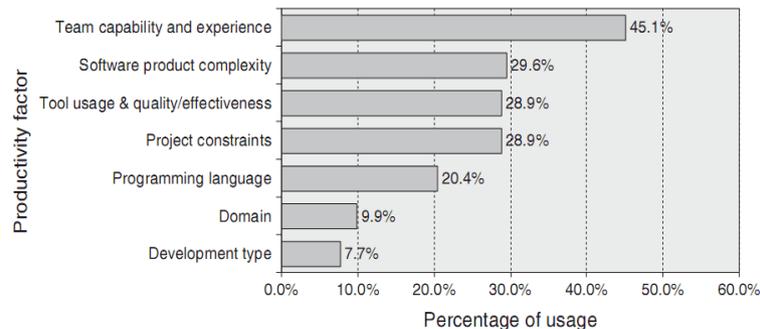


Fig. 1. Most common productivity factors reported in literature, adopted from [21]

Carpetz and Ahmed [34] emphasised the importance of skill diversity in the software engineering field. They argue that “the diversity of skills contributes to problem solving as different people see a problem from different perspectives”. Various studies have reported the influence of a variety of these factors. For instance, the effect of the experience and ability of software developers on the comprehension of web applications has been studied by Ricca *et al.* [35], and a study on the impact of developer personality on pair programming has been conducted by Hannay *et al.* [22].

The importance of people in the productivity of software development projects, as emphasised in literature, is the motivation for this study to focus on personnel attributes. Many studies have focused on and ranked the importance of different productivity factors, however, very few studies have provided numeric values for the productivity ranges of these factors. Among others, Boehm *et al.* proposed the productivity ranges of different effort multipliers for estimating the cost and schedule of software projects, using the Constructive Cost Model (COCOMO).

COCOMO is a very well-known cost/schedule estimation model for software projects. The COCOMO II.2000 is the latest version available for this model [20]. The complete Bayesian analysis on COCOMO II yields the

Productivity Ranges that provide an insight into identifying the high payoff areas to focus on in improving software productivity [36]. Therefore, personnel factor productivity multipliers of the COCOMO II model are employed for the purpose of this study.

3. The Proposed Model

The development of a System Dynamics model involves the identification of model parameters and variables, equations and relationships. These include the modelling of a software development process that comprises factors representing communication overhead, training overhead, and assimilation delay. Personnel capability factors should then be added to the model.

In order to construct and execute the model, a simulation environment with the capability to execute the model is needed. A simulation environment allows the experimentation of the developed model, which provides an opportunity to observe dynamic interactions and results. In this study, the IThink simulation environment was chosen to implement the proposed System Dynamics model and execute the simulation this was due to: (a) it is a robust and user-friendly model that covers all the needs of the current study; (b) its license was available; and (c) it has been used by others and reported to be a reliable tool [12; 37].

3.1. The System Dynamics Model

The main components (associated parameters that represent process flows, and the different factors that affect these process flows) of the base model along with extended sections are explained in this section. The main components of the model include: - Software Production Flow; - Personnel Allocation Flow; - Assimilation Time; - Training Overhead; -Communication Overhead; - Personnel Factors (all six factors).

Software Production Flow: The System Dynamics model of a software project simulates the high level of a software development process. It uses production rate and predicts the completion date of a project. The relationship between the amount of original (or remaining) work and the current development rate determines a project's scheduled completion date. The main contributors to the software production rate are effort and productivity.

Personnel Allocation Flow: A gap between the planned schedule and the actual progress of a software project would trigger the need to add new manpower. As shown in Figure 2, this trigger is represented in a model, with an auxiliary element called the personnel allocation trigger. We have also allocated a separate level for new personnel to differentiate their influence on software production from that of the initial personnel of a project.

Assimilation Time: New personnel need time to train in order to become a productive member of a project team. This training duration is called

assimilation time. On average, once new personnel pass the assimilation time, they become as productive as experienced personnel. Thus, we added an assimilation rate to the personnel allocation flow chain. The values of assimilation delays reported in literature are listed in Table 2.

Training Overhead: The existing, experienced personnel need to spend portions of their effort for mentoring new personnel. This leads to an effort loss of experienced personnel as the result of mentoring. The reported values for training (mentoring) time in literature are listed in Table 2.

Communication Overhead: Communication overhead directly affects the software production rate. It is widely held that communication overhead increases in portion to n^2 , where n is the size of the team [3] [10; 12]. Therefore, the communication overhead is expressed using an auxiliary element in the model, as depicted in Figure 2, with a non-linear function as:

$$CO = 0.06 \times NP^2, \tag{1}$$

where, *CO* is *Communication Overhead*,

and *NP* is the total *Number of Personnel*.

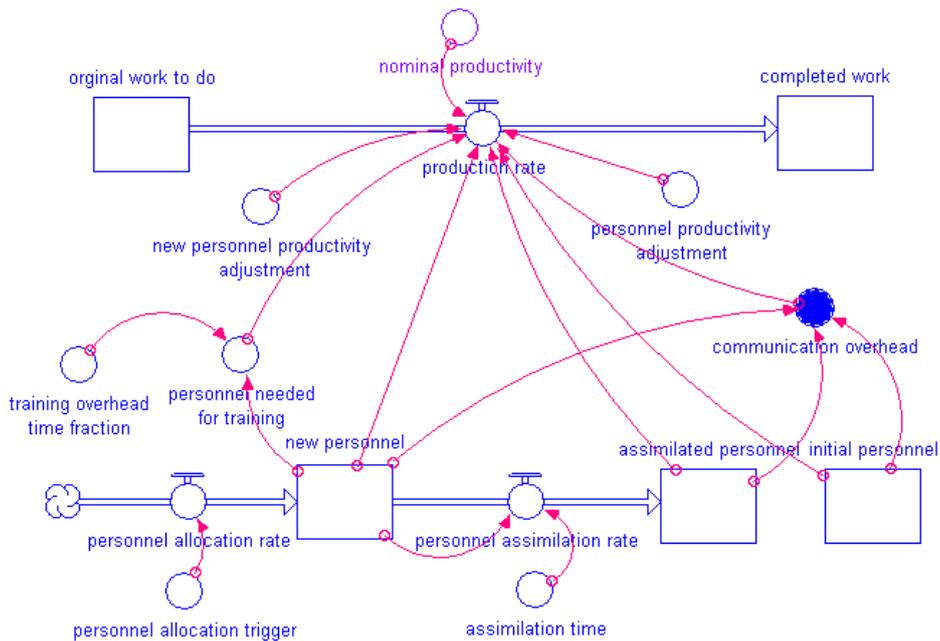


Fig. 2. Model representing training overhead, communication overhead and assimilation time

3.2. Representation of Personnel Factors in the Model

It has been stated in literature review that among related publications, the productivity drivers of COCOMO II.2000 [20] present the most appropriate values that can be employed as input variables of the personnel capabilities of our System Dynamics simulation model. The data range of productivity factors in COCOMO II shows that the combination of human factors provides the highest productivity range compared to all other types of factors, such as process, product, project and organisation. Six personnel factors from COCOMO.II(2000) [20] were employed for our study, namely, Analyst Capability (ACAP), Programmer Capability (PCAP), Personnel Continuity (PCON), Applications Experience (APEX), Platform Experience (PLEX), and Language and Tools Experience (LTEX). The productivity ranges of these human factors are shown in Figure 3.

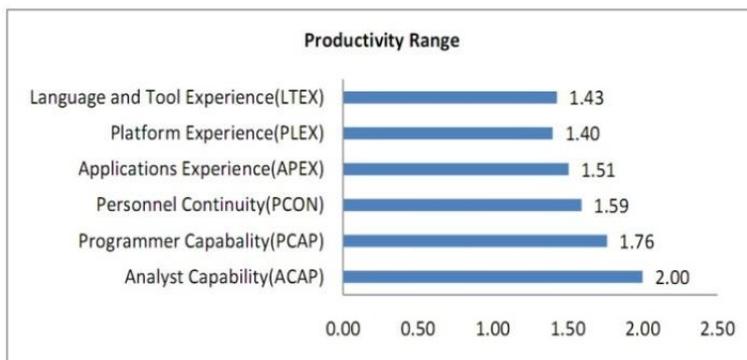


Fig. 3. Productivity ranges of COCOMO II personnel factors

Table 3. Productivity influence of COCOMO II personnel factors extracted for different levels

Personnel Factors	Very Low	Low	Nominal	High	Very High
Analyst Capability (ACAP)	0.7042	0.8403	1.00	1.1765	1.4085
Programmer Capability (PCAP)	0.7463	0.8696	1.00	1.1364	1.3158
Personnel Continuity (PCON)	0.7752	0.8929	1.00	1.1364	1.2346
Application Experience (APEX)	0.8197	0.9091	1.00	1.1364	1.2346
Platform Experience (PLEX)	0.8403	0.9174	1.00	1.0989	1.1765
Language and Tool Experience (LTEX)	0.8333	0.9174	1.00	1.0989	1.1905

We needed to make the model capable of simulating the productivity of new personnel with different levels of productivity ranges as personnel factors. The ratios between very low, low, high and very high productive parameter ratings, and the nominal productive parameter rating were extracted based on the effort multiplier values of personnel factors given in the COCOMO II cost/schedule estimation model. In other words, the

productivity rate of a person with different levels of capability in comparison to a person with the nominal level is extracted as values are reported in Table 3.

In order to simulate the influence of new manpower with different capabilities, one auxiliary element was allocated in the model for each of the above personnel capabilities in the model, namely, LTEX, PLEX, APEX, PCON, PCAP, and ACAP, as shown in Figure 4.

For the next step, we had to identify the parameters that might be influenced by personnel factor productivity multipliers. Therefore, the relevant relationships were applied and necessary equations were updated in the model, as depicted in Figure 4.

So far, it has been assumed that nominal values for assimilation time and training overhead will be applied for all new personnel regardless of their personnel experience and capabilities. However, it is rational to assume that a more capable and more experienced person would need less training and would become assimilated sooner than a person who is less capable and less experienced. Therefore, we have attempted to represent the impact of personnel factors on training overhead and assimilation time.

Based on the above explanation, two cause-effect relationships were added to the model, one between 'personnel factor multipliers' and 'assimilation time', and another between 'personnel factor multipliers' and 'training overhead time fraction', as illustrated in Figure 4. The affected factors were then proportionally adjusted according to the values given for the personnel productivity multipliers.

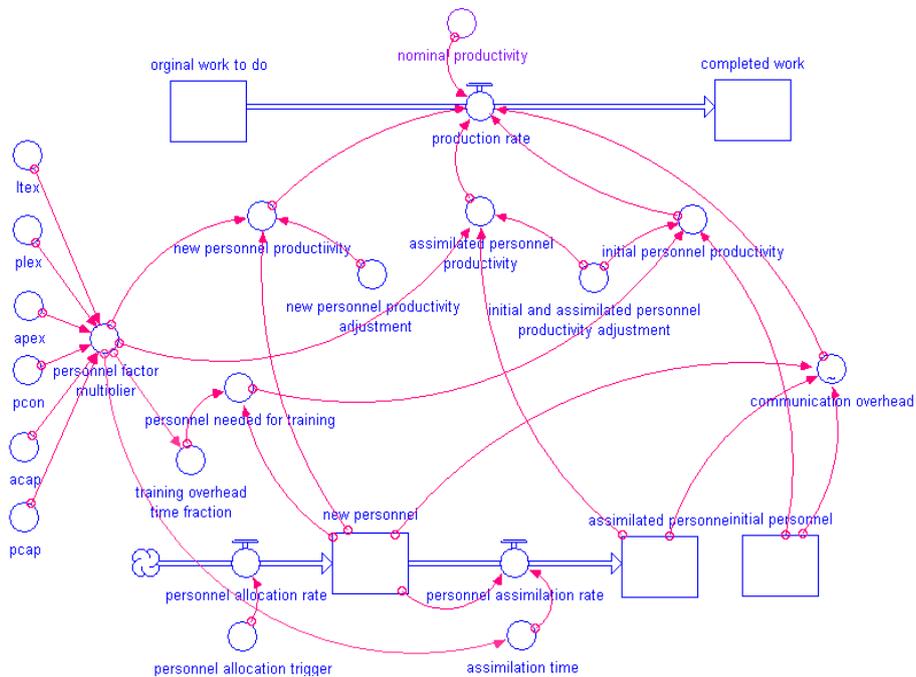


Fig. 4. The System Dynamics Model

After including all necessary factors into the model, it was now ready for simulation experiments.

4. Model Experiment - Case Study Taken from Literature

For the current research, we applied two case studies for model experiment and simulation analysis of the proposed model: (1) A case study taken from literature, and (2) A case study from an industry software project.

As the proposed model extends Madachy's model structure, and to have a basis for comparison, his reported case study was used to execute the simulation model. The implementation of the same case study from literature provided us with the advantages of evaluating the validity of our model and observing whether it produces the same results as that of the base model.

4.1. Simulation Model Settings

Variables and parameters of a model need to be initialised before running a simulation. Table 5 shows the list of parameters that were initialised.

Table 5. Simulation model specifications from literature

Parameter Name	Value
Original work	500
Initial (Experienced) Personnel	20
Estimated completion duration	274
Assimilation time	20
Communication overhead	$0.06 * n^2$
Training overhead	0.25
New personnel	0, 5, 10, 15
Productivity	Nominal productivity = 0.1 FP
New vs. Initial (experienced) personnel productivity	Productivity adjustment for experienced personnel = 1.2 Productivity adjustment for new personnel = 0.8

4.2. Experiment with Base Model Setting

Once its parameters are initialised, the model is ready to execute the simulation and analyse simulation behaviours and results. To emulate the base model setting, all personnel factor productivity multipliers in the model were set to their nominal values. Therefore, we expect the constructed model to reproduce the same simulation behaviours as reported for the base model.

For the first experiment, the model simulates the "as-is" (or default) situation, with no new manpower addition to the project. Curve #1 in Figure 5

shows the software production rate and scheduled completion day without adding new manpower. Based on the specifications of the case study, the overall software production rate of the 20-member project team is consistently equal to “1.82” function points. Based on this production rate, the project was forecast to be completed in 274 days. In the next simulation, the personnel allocation trigger was set to add 5 new personnel at about Day 110. As can be observed in curve #2, when 5 new people were added, the software team’s production rate dropped suddenly. However, it recovered during the assimilation time and eventually exceeded the default production rate. This interesting effect resulted in the project finishing three days sooner, on Day 271. Although a 3-day reduction in project duration is hardly significant, it demonstrates that Brooks’ Law doesn’t hold in all situations. Curve #3 shows the simulated process behaviour with the addition of 10 new people to the software team. The drop in development rate caused by the negative effects of adding 10 people never did correct itself to surpass or even catch up to meet the default level and the project was only completed on Day 296.

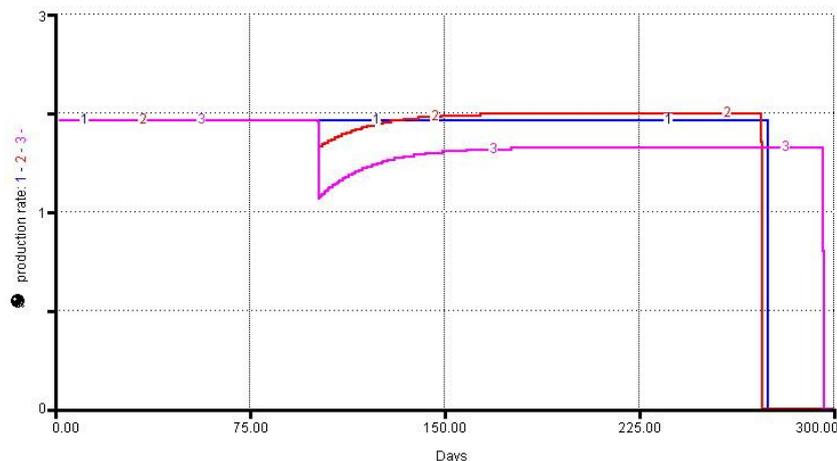


Fig. 5. Effects of adding manpower to a project at Day 110. Curve 1 - no new manpower added, curve 2 - five manpower added, curve 3 - ten manpower added

Above results demonstrates the proposed model reproduces the exact behaviour of that reported for the base model. By verifying our simulation result against a reference model, it shows that our model is reliable to use for further experiments.

4.3. Experiment of Adding New Manpower with High Levels of Capability and Experience

In this stage, we investigate how personnel attributes may impact the project completion schedule. To observe the behaviour of new changes to the model,

the same case was used for this part as well. The hiring of new manpower with very high productivity rates is then simulated. Based on the actual conditions of the case study, it is very unlikely that a project manager would add analysts to a software team at about Day 110. Therefore, we did not include the effect of analyst capability in our simulation. In addition, we also set the value of personnel continuity at the nominal level in order to focus only on the impact of the new manpower's skills, experience and capabilities.

As demonstrated in Figure 6, curve #1 indicates the simulated behaviour of the default situation in which no one is added to the project. Next is the simulation of adding 5 new members with the highest level of capability to the project. Interestingly, the curve doesn't show any drop in production rate (relative to the base model), which leads to a significant schedule reduction (of around 36 work-days) and the project finishes on Day 238 (curve #2). The addition of 10 people to the project (Figure 6, curve #3) shows some schedule improvement, although slightly less effective than adding 5 people. However, adding 15 people (Figure 6, curve #4) causes a significant drop in production rate and the project eventually ends on Day 292.

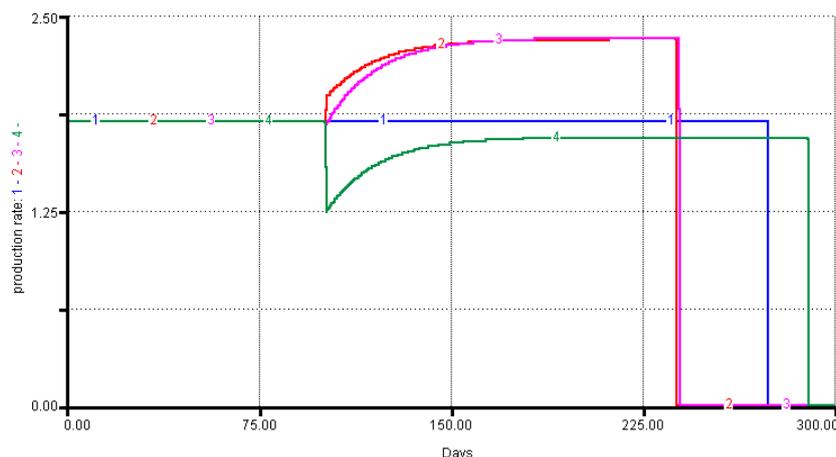


Fig. 6. Effect of adding new manpower with the best possible personnel factors to the project at Day 110. Curve 1 - no new manpower added, curve 2 - five new personnel added, curve 3 - ten new personnel added, curve 4 - fifteen new personnel added

The above result indicates that a significant schedule improvement can be achieved if people with a certain level of capability are added to ongoing software projects. The result shows that the schedule reduction was around 36 work-days. That is approximately two calendar months and can be significant enough to save a project from failure. Nevertheless, it was also found that adding a large number of people (Figure 6, curve #4), even those with high levels of capability and experience, can have a catastrophic result. Therefore, it is concluded that adding people with high levels of capability and experience can significantly minimise the negative effects of adding new

Mostafa Farshchi, Yusmadi Yah Jusoh, and Masrah Azrifah Azmi Murad

manpower, and would accelerate the progress of a project only if the number of people added is reasonable.

5. Model Experiment - A Case Study from an Industry Software Project

A real-life case study was also adopted to validate our simulation model. We compared the results and calibrated the model using data obtained from a real-life case study. For the purpose of current research and according to the defined scope, we aimed to study the effects of adding new manpower to medium-sized, in-house, new development projects.

The above constraints are applied because the assumptions and model parameter values that have previously been identified are reported in project environments with such specifications. The information of case study is obtained from a well-known company in the production of on-demand enterprise software applications. The organisation is rated at Level 3 of the Capability Maturity Model-Integrated (CMMI) model for the Software Production division. The company's Project Office provided the information of a project that matched the needs and the constraints of our study. The Project Office is responsible for all the project planning and monitoring activities.

5.1. Simulation Model Settings

The Project Office and the project manager were asked through interviews to provide information on the project size, the number of initial personnel, the number of new people added to the project and the time they were added, the capability and experience of these new personnel, etc. Upon receiving these project specifications, we calibrated and customised the model according to the following characteristics of the previous case study.

Inconstancy of initial personnel: Based on the information obtained from the case study, we noticed a pattern in the Project Office's policy for personnel allocation. A project would usually begin with the minimum number of personnel required to perform the fundamental analysis and design activities. More personnel would progressively be added to the project for detailed design, development and testing activities. Gradually, after deploying and finalising individual modules, a few people will be released from the project. This is because sequential task constraints exist in different phases of a software project. For instance, in the construction phase, breaking tasks into smaller ones are more practicable than in the inception phase. Therefore, changes in the number of personnel were implemented into the model and the constant value of the initial personnel was converted into a variable that represents the number of personnel during the project lifecycle. Changes in the number of personnel wouldn't add overheads to the project as its effects had already been taken into account prior to project commencement.

Adding people with distinct capabilities: In the previous case study, the capabilities of new manpower were within the equivalent range. However, in this case people have different capabilities and experience. To manage such an issue, we designed a spreadsheet into which the personnel factors of individuals could be entered. The spreadsheet is then bound to its related representative factor of the model. This enabled the representation of new manpower with distinct attributes. The detailed list of people added, along with their capability and experience levels, is given Table 6.

Table 6. Capability and experience levels of new personnel added to the project (N: Nominal, VL: Very Low, L: Low, H: High, VH: Very High)

Personnel	PCAP	PCON	APEX	PLEX	LTEX
Person One	VH	N	H	VH	VH
Person Two	H	N	H	H	H
Person Three	H	N	N	H	H
Person Four	H	N	H	N	H
Person Five	N	N	H	N	N
Person Six	N	N	L	N	N

Based on the explanations given and the project specifications, the parameters of the model were initialised (as presented in Table 7), after which the simulation was ready to be executed and the simulated behaviour analysed using the results obtained.

Table 7. Simulation model specifications of the second case study

Parameter Name	Value	Description
Original work	760	Project Size in Function Points (FPs), equivalent to 152 main tasks, where each task, on average, is equal to 5 FPs.
Initial personnel	10 – 16	Equivalent number of fulltime personnel that varies during the project lifecycle.
Planned completion days	375	Estimated completion time in work-days
Assimilation time	40	Assimilation duration in work-days
Communication overhead	$0.06 * n^2$	n is the number of personnel who will be communicating with one another.
Training overhead	0.25	The portion (in this case, 25%) of a fulltime, experienced person's time needed to train new personnel during the assimilation period.
New personnel	6	Number of personnel added from about Day 200.
Planned Nominal productivity	0.17	Expressed in function points/person-day.
Actual Nominal productivity	0.14	Expressed in function points/person-day.

Parameter Name	Value	Description
New vs. Initial(experienced) personnel productivity	New Personnel= 1 Exp Personnel= 0.5	New personnel productivity level is set to half that of experienced personnel

5.2. Simulation with Base Model Settings

The effects on the project’s progress of adding people with the given capabilities are investigated in the following simulation run.

Curve #1 in Figure 7 plots the expected project progress (completed work in FPs) based on the project plan. This shows that the project was planned to be completed in 375 days. However, the project’s actual progress did not match its planned progress. Curve #2 shows that the project progressed behind schedule. Based on the team’s average production rate after 200 days, only 307 FPs were completed, which is 35 work-days (or 20%) behind the scheduled 387 FPs. The curves continue to diverge, widening the gap between the planned and actual schedules toward the end of the project, prompting the project management team to assign 6 more people.

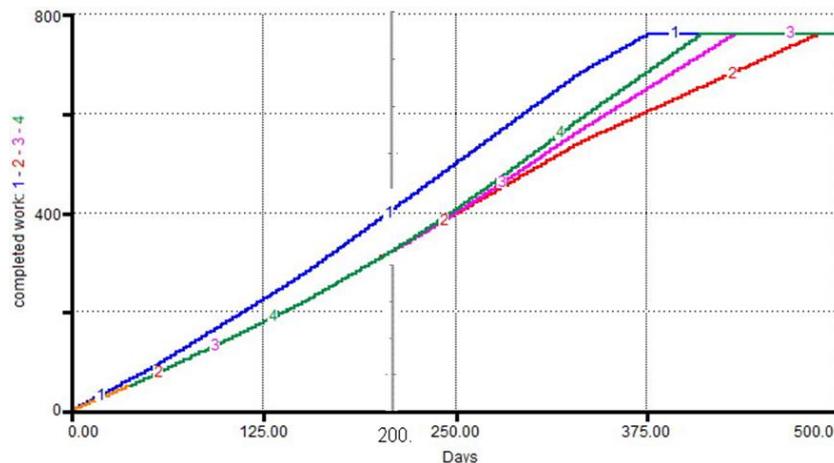


Fig. 7. Effect of adding new personnel to the project (without considering the impact of personnel factors). Curve 1 - planned, curve 2 - no new manpower added, curve 3 - six new personnel added on Day 200

To investigate the effect of adding six new people to the project, simulation was performed, first without consideration of their personnel attributes, as illustrated by curve #3. This shows that adding new people has a positive effect on the overall progress, but the project could still not sufficiently catch up and make up for the delay. Curve #4 simulates the same situation with consideration of personnel factors. Although the progress did not achieve the planned situation, the work was completed considerably faster (relative to

curve #3) after adding people at around Day 200. This shows that adding new people with more than the nominal capability level has a positive impact on a project's progress.

Simulations of the team's production rates are shown in Figure 8, Curves #1 and #2 show the changing production rates of the software team throughout the project lifecycle, caused mainly by the changing number of personnel. In the project plan, team members were expected to achieve an average productivity rate of 3.48 FPs per month. However, as at Day 200, the actual average production rate was only 2.76 FPs per month. Curve #2 forecasts the project completion based on an average production rate of 2.76 FPs per month, giving an estimate that the project would be completed in 486 days. Unfortunately, this is more than 100 days later than planned.

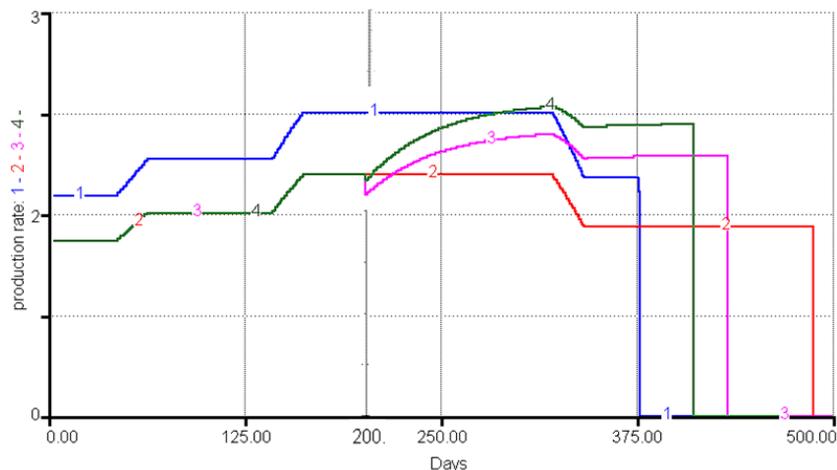


Fig. 8. Effect of adding new personnel to the project (with and without considering the impact of personnel factors). Curve 1 - planned, curve 2 - no new manpower added, curve 3 - six manpower added on Day 200, curve 4- six new personnel added on Day 200 with including the personnel capability and experience

Curve #3 in Figure 8 illustrates the effect of adding six people at about Day 200, all of whom with productivities equal to the average rate recorded during the first 200 days of the project. It shows a small drop in productivity once new personnel is added, picking up during the assimilation time, and finally achieving overall higher team productivity, completing the project on Day 434. In the next simulation run (curve #4), the effects of personnel capability and experience were included, showing that when new manpower was added to the project, the production rate almost did not drop, and instead, increased progressively over the assimilation period, whilst training overhead was gradually eliminated as new people become trained and assimilated. The project completes after approximately 406 days, which is considerably shorter than the 439 days it would take without consideration of the personnel factors of new manpower. Although it still failed to bring the project back to schedule,

it completed much sooner than if the project had been continued without adding new manpower. But the question is whether or not the forecast progress and completion day would approximately represent the actual situation. This is discussed in the next section.

5.3. Analysis of Project Progress: Simulation against Actual

In this section we attempt to evaluate how accurately the proposed simulation model would represent an actual situation. Based on the values given by the Project Office, the actual progress of the project is plotted in Figures 9 and 10.

Figure 9 shows the cumulative work progress during the project lifecycle. Curve #1 shows the progress as predicted by the proposed simulation model, while curve #2 is the reported actual progress of the project. Although the simulated progress doesn't exactly mimic the real situation, it is a close approximation. Interestingly, as the figure shows, the actual project had completed only slightly sooner than predicted by the simulation.

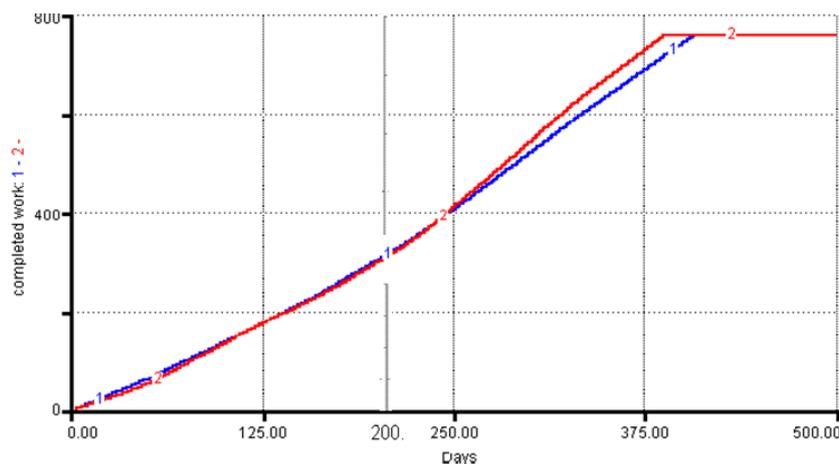


Fig. 9. Project Progress: Curve 1 - simulation experiment, curve 2 - actual situation

Figure 10 shows the difference in production rate between the simulation results and the actual situation. Curve #1 shows the production rate and completion date of the project as predicted by the simulation, while Curve #2 shows the actual production rate as recorded monthly. The simulation forecast that the project would be completed at around Day 406. This was based on the average productivity rate from project commencement to Day 200, and took into account the addition of six people with predicted personnel capabilities. In the actual situation (curve #2), the software team's productivity rate follows almost the same pattern, but with some fluctuation, as that predicted by the simulation. From around Day 220, curve #2 gradually surpasses curve #1 and continues to maintain its productivity difference from

curve #1, eventually leading to the project completing sooner at around Day 387.

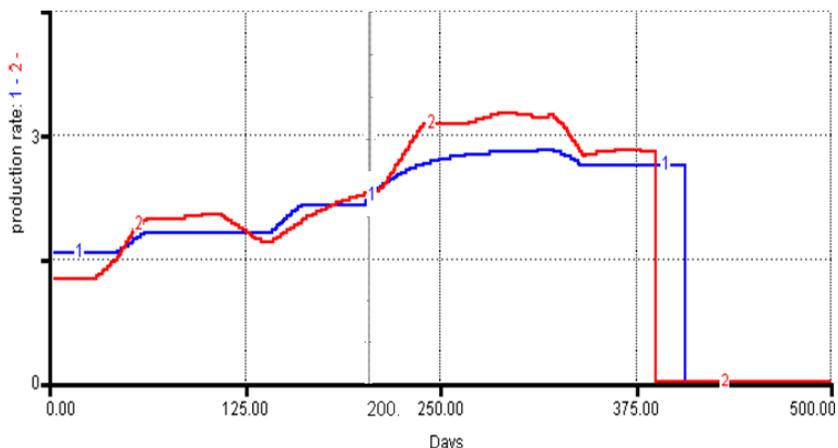


Fig. 10. Software team productivity: Simulation experiment against real situation. Curve 1 - simulation experiment, curve 2 - actual situation

To investigate how accurately the simulation result predicts an actual situation, we measured the percentage error using the absolute differences between the simulated and experimented values. As the project completion day was forecast at Day 200, the baseline for calculation was set at Day 200. Therefore, we have:

$$\text{Error \%} = \frac{|\text{Experimental} - \text{Theoretical}|}{\text{Theoretical}} \times 100 \quad (2)$$

$$= \frac{|(387 - 200) - (406 - 200)|}{406 - 200} \times 100 = \frac{19}{206} \times 100 = 9.23 \%$$

The proposed System Dynamics model with consideration of personnel factors demonstrated 90.77% accuracy in the investigated case study. The result gives us confidence that the proposed model is capable of approximately simulating an actual situation, and can be employed for decision trade-off analysis by software project managers.

5.4. Comparison of Experiment Results

In earlier sections, the simulation results of different scenarios were presented and discussed. Here, we summarise the results gained from the simulation experiments and compare them with a case study. Table 8 shows the project's scheduled completion days in each scenario, and compares the

predicted completion days in the different scenarios with the actual completion days.

Table 8. Comparison of scheduled completion days in a real-life case study

Scenario	Project completion days	Differences in schedule compared with actual situation	Prediction accuracy in comparison with actual situation
(1) Plan based on expected productivity	375	-12 days	Not applicable
(2) Estimation based on actual productivity	486	+99 days	Not applicable
(3) Estimation based on actual productivity and new manpower	434	+47 days	79.91%
(4) Estimation based on actual productivity, new manpower, personnel factors, relative assimilation time and training overhead	406	+19 days	90.23%
(5) Actual situation	387	0	100%

The above comparisons show that when personnel factors are taken into account, the prediction achieves better accuracy. Furthermore, it should be noted that although adding people did not help bring the project back on schedule, it was effective in preventing further delays. The above findings affirm our confidence that the proposed simulation model can approximately represent an actual outcome. The results indicate the importance of personnel capability and experience to productivity and, consequently, to an improvement in project schedule.

6. Trade-off Analysis

As previously mentioned, an aim of this study is to enable project managers foresee the possible consequences of their decisions and to perform various trade-off analyses. In experiments for the first case study (section 4.1), it was observed (Figure 6) that too many people, despite having very high levels of capability and experience, would not be beneficial to a project and would be a waste of time and resources. In this section, we will explore how the proposed simulation model can help project managers find the optimal number of personnel to add to a project at a given time, and examine at which point on the time scale would adding more manpower be a clearly wrong decision.

In an actual situation, project managers have to trade-off the number of personnel that can be added to a project with the additional cost it would incur. A project manager can take advantage of the proposed simulation model to perform various sensitivity analyses using simulation experiments to find the best possible solution for his project. These analyses can factor in a

range of different variables, such as the number of team members, the levels of capability and experience of the manpower, and the time at which new manpower could be added to the project. Due to space constraints, it is not possible to show the result of each sensitivity analysis on paper of this size. However, a summary of the simulation results using three different numbers of additional manpower (3, 6 and 9 new personnel) at four different points on the time scale (Days 200, 250, 300 and 350) in an industry case, as explained in section 4.2, is provided in Table 9.

Based on the above settings, the results (Table 9) show that adding more new people leads to a bigger reduction in schedule up to a certain point. However, after this point, as the number of additional manpower increases, the effect on reducing the schedule decreases. For instance, the difference between adding 3 and 6 new personnel on Day 200 is 26 days, but the difference is less than 6 days between adding 6 and 9 new team members.

Table 9. Trade-off analysis of the effect of adding different numbers of new personnel at four different points on the time scale.

Day	Item	3 New Personnel	6 New Personnel	9 New Personnel
200	Completion date (day)	432	406	400
	Cumulative added effort (day)	858	1716	2574
	Cost of adding manpower	128700	257400	386100
	Added value	216000	320000	344000
	Trade-off Cost	87300	62600	-42100
250	Completion date (day)	441	420	415
	Cumulative added effort (day)	708	1416	2124
	Cost of adding manpower	106200	212400	318600
	Added value	180000	264000	284000
	Trade-off Cost	73800	51600	-34600
300	Completion date (day)	450	433	429
	Cumulative added effort (day)	558	1116	1674
	Cost of adding manpower	83700	167400	251100
	Added value	144000	212000	228000
	Trade-off Cost	60300	44600	-23100
350	Completion date (day)	460	448	444
	Cumulative added effort (day)	408	816	1224
	Cost of adding manpower	61200	122400	183600
	Added value	104000	152000	168000
	Trade-off Cost	42800	29600	-15600

In a real-life situation, a project manager needs to work out the best point on a time scale at which to add new people and the ideal number of new personnel to add for his project. The trade-off between the cost of adding new manpower and the added value they bring to the project (as the result of a

reduction in project duration) would answer the above questions. For example, if we determine that a delay in project completion would cost \$4,000 per day, and the cost of adding new manpower is \$150 per day per person, as shown in Table 9, then the trade-off of adding 9 new personnel is negative in all cases. The figures show that the earlier the new staff is added, the higher the trade-off value obtained. In this case, adding 3 new team members would be the best fit for the project in terms of trade-off cost at each of the four points on the time scale.

7. Model Evaluation

It is important to note that a model is seldom intended to be a complete and accurate representation of a real system. Usually, the aim of evaluating a System Dynamics model is not to validate its accuracy; instead, it's about making sure that the model provides greater insight into and better understanding of a phenomenon. Given the above, the verification and validation of this study are focused on building confidence in the model as a reasonable representation of the system and in its usefulness in producing results. Hence, the assessment of the proposed system dynamics simulation model has been performed mainly in the following stages: model verification, and model validation.

During model verification, we examined whether its implementation is error-free and properly represents the intended logical behaviour. For this purpose, the model components were first reviewed and analysed during its incremental development stage. Second, the model was verified by comparing its simulated behaviour with that of the base model.

In model validation, we examined whether the model addresses the problem defined in the current study. Two experiments, one with values reported from the case study, and the other one with data taken from a real-life situation, were chosen to perform the examination.

Comparing the simulation experiments with the results obtained from a case study reported in literature assures that the model has behavioural consistencies with the previously reported results. Furthermore, validating the results of the simulation against those obtained from the actual reported progress of a project in a real-life environment strengthened our confidence that the model can be effectively employed by project managers.

8. Conclusion

In order to fulfil the objectives of the study, a System Dynamics model was designed and constructed. Experiments were performed to verify and validate the model, and to explore the effects of adding new manpower with different personnel capabilities, skills and experiences to delayed software projects.

Experiments with two case studies demonstrated that the negative effects of adding new manpower to the progress of a project can be considerably minimised when new manpower with certain levels of personnel factors is added. The results obtained show a significant improvement in project schedule for both the case studies.

In the first case study, which was taken from literature, the objectives of doing the experiments were, firstly, to verify the accuracy of the model and, secondly, to have a basis of comparison for the different experiment's results. The outcome of performing the experiments with base work simulation settings assures that the model correctly replicates the previously reported results.

In the second case study, which was taken from a real-life project, the aims of performing the experiments were, firstly, to identify the limitations of the model when applied to a real-life environment and, secondly, to assess the accuracy of the forecast results against the actual situation. The results of the experiment were most promising, demonstrating that the model is capable of approximately forecasting actual progress and the actual completion date of a project.

The proposed System Dynamics model can be employed for trade-off analysis of a software project and to forecast the impact of different decisions on changing manpower in a project. It has significant benefits for the software project managers, namely providing better clarification on the status of a project, forecasting the optimum staff level required to meet a deadline, and reducing the risk of decision making. The above advantages help project managers become less doubtful and more confident about their decisions as the simulation analysis reveals the likely consequences of their decisions.

The research and experiment in this study focused on the impact of personnel skills, capability and experience, thus one of the good future extensions of this work could be the study of how psychological and social characteristics of personnel could contribute to faster fitting to a software production working team.

References

- [1] It-cortex. Statistics on IT Projects Failure Rates. October 2009, from http://www.it-cortex.com/Stat_Failure_Rate.htm .2002
- [2] Standish-Group. New Standish Group Report Shows More Projects Failing and Less Successful Projects. Retrieved October 2009, from http://www1.standishgroup.com/newsroom/chaos_2009.php. 2009
- [3] Abdel-Hamid, T. and S. E. Madnick. Software Project Dynamics: An Integrated Approach. Upper Saddle River, Prentice-Hall.(1991)
- [4] Callegari, D. and R. Bastos. A Systematic Review of Dynamic Reconfiguration of Software Projects. Information technology and management 1(3): 103-113.(2008)
- [5] Eden, C., T. Williams and F. Ackermann. Dismantling the Learning Curve: The Role of Disruptions on the Planning of Development Projects. International Journal of Project Management 16(3): 131-138.(1998)

Mostafa Farshchi, Yusmadi Yah Jusoh, and Masrah Azrifah Azmi Murad

- [6] Ford, D. and J. Sterman. Dynamic Modeling of Product Development Processes. *System Dynamics Review* 14(1): 31-68.(1998)
- [7] Lyneis, J. and D. Ford. System Dynamics Applied to Project Management: A Survey, Assessment, and Directions for Future Research. *System Dynamics Review* 23(2-3): 157-189.(2007)
- [8] Brooks, F. P. *The Mythical Man-Month (Anniversary Ed.)*, Addison-Wesley Longman Publishing Co., Inc.(1995)
- [9] Stutzke, R. A Mathematical Expression of Brooks' Law. 9th International Forum on COCOMO and Cost Modeling, Los Angeles, CA, 1-24.(1994)
- [10] Hsia, P., C. Hsu and D. Kung. Brooks' Law Revisited: A System Dynamics Approach. *Proceedings of 23rd Annual IEEE Computer Software and Applications Conference, COMPSAC, Phoenix, USA, IEEE Computer Society: 370-376.*(1999)
- [11] Williams, L., Shukla A., and Anton A.I. An Initial Exploration of the Relationship between Pair Programming and Brooks' Law. *Proceedings of the Agile Development Conference, IEEE Computer Society.*(2004)
- [12] Madachy, R. *Software Process Dynamics*. Chichester, Wiley-IEEE Press.(2008)
- [13] Schweik, C. M., R. C. English, M. Kitsing and S. Haire. Brooks' Versus Linus' Law: An Empirical Test of Open Source Projects. *Proceedings of the international conference on Digital government research, Montreal, Canada, Digital Government Society of North America.*(2008)
- [14] Adams, P., A. Capiluppi and C. Boldyreff. Coordination and Productivity Issues in Free Software: The Role of Brooks' Law. *Proceedings of ICSM 2009, Edmonton, Canada: 319-328.*(2009)
- [15] Capiluppi, A. and P. J. Adams Reassessing Brooks' Law for the Free Software Community. In. *Open Source Ecosystems: Diverse Communities Interacting*, Springer Boston. 299: 274-283.(2009)
- [16] Neus, A. and P. Scherf. Opening Minds: Cultural Change with the Introduction of Open-Source Collaboration Methods. *IBM Systems Journal* 44(2): 215-225.(2010)
- [17] Samoladas, I., L. Angelis and I. Stamelos. Survival Analysis on the Duration of Open Source Projects. *Information and Software Technology* 52(9): 902-922.(2010)
- [18] Pressman, R. *Software Engineering: A Practitioner's Approach Sixth Edition*. New York, McGraw-Hill.(2006)
- [19] Finnie, G., G. Wittig and D. Petkov. Prioritizing Software Development Productivity Factors Using the Analytic Hierarchy Process. *Journal of Systems and Software* 22(2): 129-139.(1993)
- [20] Boehm, B. W., Clark, Horowitz, Brown, Reifer, Chulani, R. Madachy and B. Steece. *Software Cost Estimation with Cocomo li with Cdrom*. Upper Saddle River, Prentice Hall.(2000)
- [21] Trendowicz, A. and J. Münch. Factors Influencing Software Development Productivity-State-of-the-Art and Industrial Experiences. *Advances in Computers* 77: 185-241.(2009)
- [22] Hannay, J. E., E. Arisholm, H. Engvik and D. I. K. Sjoberg. Effects of Personality on Pair Programming. *IEEE Transactions on Software Engineering* 36(1): 61-80.(2010)
- [23] Putnam, L. A General Empirical Solution to the Macro Software Sizing and Estimating Problem. *IEEE Trans. Software Eng.* 4(4): 345-361.(1978)
- [24] McCain, K. and L. Salvucci. How Influential Is Brooks' Law? A Longitudinal Citation Context Analysis of Frederick Brooks' the Mythical Man-Month. *Journal of Information Science* 32(3): 277.(2006)

Impact of Personnel Factors on the Recovery of Delayed Software Projects: A System Dynamics Approach

- [25] Verner, J., S. Overmyer and K. McCain. In the 25 Years since the Mythical Man-Month What Have We Learned About Project Management? *Information and Software Technology* 41(14): 1021-1026.(1999)
- [26] Abdel-Hamid, T. The Dynamics of Software Project Staffing: A System Dynamics Based Simulation Approach. *IEEE Transactions on Software Engineering* 15(2): 109-119.(1989)
- [27] Opelt, K. Overcoming Brooks' Law. In. *Agile Processes in Software Engineering and Extreme Programming*. Berlin Heidelberg, Springer-Verlag: 175-178.(2007)
- [28] Wu, M. and H. Yan. Simulation in Software Engineering with System Dynamics: A Case Study. *Journal of Software* 4(10): 1127.(2009)
- [29] Arisholm, E., H. Gallis, T. Dyba and D. I. K. Sjoberg. Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise. *IEEE Transactions on Software Engineering* 33(2): 65-86.(2007)
- [30] Nosek, J. T. The Case for Collaborative Programming. *Communication of ACM* 41(3): 105-108.(1998)
- [31] Salleh, N., E. Mendes, J. Grundy and G. Burch. An Empirical Study of the Effects of Conscientiousness in Pair Programming Using the Five-Factor Personality Model. *Proceedings of ICSE '10 The 32nd ACM/IEEE International Conference on Software Engineering*, Cape Town, South Africa, ACM: 577-586.(2010)
- [32] Ma, Z. Intelligent System Dynamics Modeling and Decision Analysis for Software Schedule Slippage Recovery. *Doctoral Dissertation*. Arizona State University. (2000)
- [33] Maxwell, K., L. Van Wassenhove and S. Dutta. Software Development Productivity of European Space, Military, and Industrial Applications. *IEEE Transactions on Software Engineering* 22(10): 706-718.(1996)
- [34] Capretz, L. F. and F. Ahmed. Why Do We Need Personality Diversity in Software Engineering? *ACM SIGSOFT Software Engineering Notes* 35(2): 1-11.(2010)
- [35] Ricca, F., M. Di Penta, M. Torchiano, P. Tonella and M. Ceccato. How Developers' Experience and Ability Influence Web Application Comprehension Tasks Supported by Uml Stereotypes: A Series of Four Experiments. *IEEE Transactions on Software Engineering* 36(1): 96-118.(2010)
- [36] Chulani, S., B. Boehm and B. Steece. Bayesian Analysis of Empirical Software Engineering Cost Models. *IEEE Transactions on Software Engineering* 25(4): 573-583.(1999)
- [37] Madachy, R. A Software Project Dynamics Model for Process Cost, Schedule and Risk Assessment. *Doctoral Dissertation*. University of Southern California. 1994)

Masrah Azrifah Azmi Murad received her PhD in Artificial Intelligence from the University of Bristol, UK in 2005. She is currently an associate professor at the Department of Information Systems, Universiti Putra Malaysia. She is a member of IEEE and IEEE Computer Society. Her current research interests include text mining, applied informatics, and automated software engineering.

Mostafa Farshchi received his Master of Science in Software Engineering from Universiti Putra Malaysia (UPM) in 2011. His research interest includes software engineering, system dynamics, and agent technology. He has more than 6 years experience on enterprise software development and works as a senior software engineer in enterprise software projects.

Mostafa Farshchi, Yusmadi Yah Jusoh, and Masrah Azrifah Azmi Murad

Yusmadi Yah Jusoh received the B.Econs. and M.IT. degrees from Universiti Kebangsaan Malaysia (UKM) in 1996 and 1998, respectively. She received the PhD from the same university in 2008. She is a lecturer at Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM) since 1998. Her research interest includes information systems, information technology strategic planning, software engineering and management information system.

Received: May 25, 2011; Accepted: March 08, 2012.