A SITUATION-DEPENDENT SCENARIO GENERATION FRAMEWORK FOR PROJECT MANAGEMENT SKILL-UP SIMULATOR

Koichi Iwai[1], Masanori Akiyoshi[1], Masaki Samejima[1] and Hiroshi Morihisa[2]

[1]*Graduate School of Information Science and Terchnology, Osaka University, Yamadaoka 2-1, Suita, Osaka, Japan*

[2]*NS Solutions Corporation, 20-15 Shinagawa 2 Chome, Chuo-ku, Tokyo, Japan*

Keywords: Project management simulation, Scenario generation, Trouble event control.

Abstract: This paper addresses a new framework aiming situation-dependent scenario generation for project management skill-up simulator. Project management is inherently human-centric activities, and research work for education has been done by using simulation. Project management covers several aspects on software development such as planning, scheduling, progress management and negotiation. We especially focus on the progress management phase to provide high fidelity of project status and well-configured learning situation towards pedagogical achievement. First three design principles are argued for such viewpoints. Second simple but fully functional project modeling is proposed for simulating essential aspects of Q(uality), C(ost) and D(elivery) criteria. Third situation-dependent scenario generation is described with "Events" and "Trigger control of trouble events". The proposed framework is implemented and shows effective scenario generation when having a trainee's interactive operations.

1 INTRODUCTION

Management of software development is often argued from Q(uality), C(ost), and D(elivery) viewpoints. Of course such "QCD" is significant criteria from both developer and outsourcer enterprise activities. Therefore several tools or a reference guide book (PMI, 2009) are provided for the software project management. However, even if such tools and books are well-prepared, one of the significant factors that lead to success of the project is still under project managers' knowledge and skill so far. Therefore education on project managers are crucial aspects and educational framework is discussed, for instance, OJT(On the Job Training), PBL(Project Based Learning), RP(Role Playing) and so forth. The difficulty of project manager education is due to the lack of definitive way of tutoring methodology, because guidance along with pre-defined scenarios are not effective to learn deep insight on project management. From such characteristics on project manager education, expert tutors are indispensable to provide good pedagogical environment to trainees, which costs much and limits the number of trainees.

Based on these aspects on project manager education, there exists strong need to establish the method on hands-on computational learning environment with situation-dependent scenarios and pedagogical guidance. One of the most expected approach is to apply simulation technology with reactive functionality to a learner's operations. Since the optimized or complete operational sequence as to the project depends on the situation and is inherently difficult to solve, the requirement on pedagogical guidance is sufficiently explanatory itself even if it provides semi-optimal operations. On the other hand, the requirement on situation-dependent scenarios have strict criteria such as no contradiction to our real project management world. Scenarios need to provide fidelity in a sense. This paper addresses such a situation-dependent scenario generation method with project simulation. Project simulation model is also discussed to achieve our final goal. In the following sections, design principle, scenario generation method, experimental results, and related work are described.

Iwai K., Akiyoshi M., Samejima M. and Morihisa H..
A SITUATION-DEPENDENT SCENARIO GENERATION FRAMEWORK FOR PROJECT MANAGEMENT SKILL-UP SIMULATOR.
DOI: 10.5220/0003609504080412
In *Proceedings of the 6th International Conference on Software and Database Technologies* (ICSOFT-2011), pages 408-412
ISBN: 978-989-8425-77-5

2 DESIGN PRINCIPLE OF THE PROJECT MANAGER SKILL-UP SIMULATOR

As mentioned before, detail investigation on design principle is necessary to construct the project manager skill-up simulator with scenario generation. First of all, we set the target scope on project management education, because it includes planning, scheduling, progress management and negotiation in general. Since "QCD" is directly related to progress management, this paper focuses on progress management aspect, which might fit to use simulation.

2.1 Principle from Management Metrics

As for the progress management, "Gantt chart" is usually used as interfaces for grasping overall project status in practical software development project. In fact, from the gantt chart, expert project managers predict several factors which cause some obstacles in the project. For instance, trouble symptoms on progress delay are implicitly appeared in the chart bar length change of actual results. Recent EVM(Earned Value Management) methodology is also useful to predict performance of the project. High fidelity is significant factors to educate trainees, the first design principle of our proposed simulation model is to fully generate the project dynamics from such "EVM" metrics.

2.2 Principle from Educational Contradiction-free

Under the condition that such project dynamics is provided by the simulation model, if any operation by trainees resorts to project management failure, it is not effective as educational functionality. This is contradiction as educational methodology. So it is necessary to guarantee that some operations lead to project management success and others lead to project management failure in accordance with situations. Since the situation is generated based on initial project status and operations by trainees, such contradiction is not defined in advance. Therefore the second design principle of our proposed scenario generation is to guarantee the project dynamics from such educational contradiction-free.

2.3 Principle from Learning Difficulty Adjustment

A trainee's operation drives the project status by using the simulation model. If a trainee of novice project manager level faces to complex situation, he/she fails to make appropriate operations even if he/she has sufficient knowledge. Of course having some failures is effective to awaken his/her consciousness and knowledge, but it is also necessary to avoid confused situation when he/she cannot judge the situation. This indicates adjusting scenarios along with acceptable disorder on the project status. The educational aim is not to provide failure experience but to make trainees awaken their knowledge and skill when facing some symptoms on project obstacles. During the session of "project manager skill-up simulator", it provides a sense of tightrope walking and finally crossing over the gap. The third design principle is to control the scenario with learning difficulty adjustment.

3 SCENARIO GENERATION WITH SIMULATOR

Our three design principles provide high fidelity of real project management and situation-dependent scenario of pedagogical viewpoints. Figure 1 shows outline of the learning environment. The most distinctive feature of the environment is project model with generating dynamics from "EVM" metrics and scenario generation event triggering mechanism with a sense of tightrope walking.

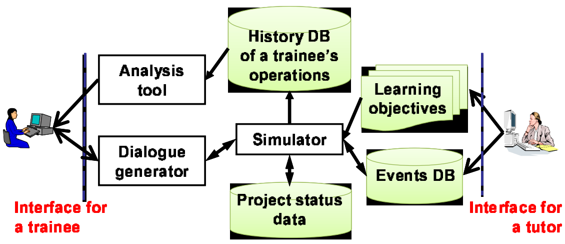

Figure 1: Outline of the learning environment.

3.1 Project Model

The project model mainly consists of "Project", "Module", and "Person" as they exist in the real project. "Project" defines a set of modules to be developed and members to be assigned to the project. The dependency of modules is depicted as a whole. "Module" defines each estimated man-hour and technical domain such as database, and its difficulty ranked with "A(difficult)", "B(normal)" and “C(easy)”. “Person” defines each member’s skill on technical domains ranked with “A(expert)”, “B(normal)” and “C(novice)”. These three level description on module difficulty and person skill generates project dynamics from “EVM” viewpoints. In fact, mismatching between module difficulty and person skill is main cause of schedule delay and quality loss of a project. And the most crucial task as a project manager is to detect/predict such schedule delay and quality loss from project crisis management and to make proper operation, for instance, “overtime directive”, “member assignment change” and “increase personnel number”. In a certain situation, “overtime directive” is accompanied with supervising action which means well-skilled members teach members under trouble, especially debugging of detected bugs. In fact, the process of debugging generates new bugs and some members cannot solve such iterative chain of generating bugs without supervising. In addition to the detected bugs, latent bugs are also modeled.

Thus our project model really mimics the real world project essential dynamics with schedule delay from member assignment and quality loss from bug generation. The schedule delay influences to D(elivery) metric and the quality loss including latent bugs does to Q(uality) metric. C(ost) metric is calculated from overtime, supervising and increase personnel number for recovery operation against schedule delay and quality loss.

EVM(Earned Value Management) traces the three project management indices; planned value, earned value, and actual cost along with time-line. The difference of planned value and earned value indicates either delay or ahead of schedule. The difference of planned value and actual cost indicates either addition or reduction of man-hour. In our project model, the planned value corresponds to planned man-hour cost depicted in the “Module” attribute. The earned value corresponds to converted man-hour cost which is derived from the dynamical progress model using the difficulty attribute of “Module” and the skill difficulty of “Person”. The actual cost corresponds to man-hour cost which basically adds surplus cost to initially planned man-hour cost for recovery operation. Estimation error on initially planned man-hour is not modeled so far, therefore surplus cost comes from debugging in our simulation.

Table 1 shows attributes of the model, related project dynamical aspects and related EVM metrics.

Table 1: Model, project aspects and related EVM metrics.

Model attributes	Project aspects	EVM metrics
project bug ratio	standard no. of bugs	actual cost
module dependency	member assignment	—
module difficulty	no. of bugs, member assignment	earned value actual cost
module man-hour	overtime directive	actual cost
person skill	no. of bugs, member assignment	earned value actual cost
person overtime	earned value actual cost	actual cost

3.2 Scenario Generation

3.2.1 Events

The trainee basically receives daily report as to each module and grasp the project status. Daily reports include daily progress and problem if exists, that is, symptoms on schedule delay and troublesome debugging related to quality loss. Hence such reports sometimes do not reflect true status in spite of quite critical ones for the project manager. Therefore individual review, we call progress check, is necessary in addition to formal review meetings. Such prophylactic action skill is regarded as the most indispensable one as well-skilled project managers. The progress check discloses hidden progress delay and accumulated bugs, and arouse a trainee to make operations or leave as it is.

Scenario is generated based on the above-mentioned intertwined interaction between model-driven dynamics and a trainee’s operations. Figure 2 shows a screenshot on the prototype simulator with “Gantt chart” and “interaction panel” interface. The module status is displayed with two bars that represent initial planned duration and changed duration of starting date and completing date, and progress bars indicating each module progress in a green and red color. The “green” portion means the completed progress, and the “red” portion means the remaining man-work. This progress bar provides quite significant information to a trainee from practical viewpoints. If the green colored portion remains unchanged, the module is probably under debugging process. If the change rate of the green colored portion is lower than the day progress indicator that is gray-filled in the chart area, the module has possibility of progress delay. In real projects, such investigation is necessary to grasp the project status deeply.

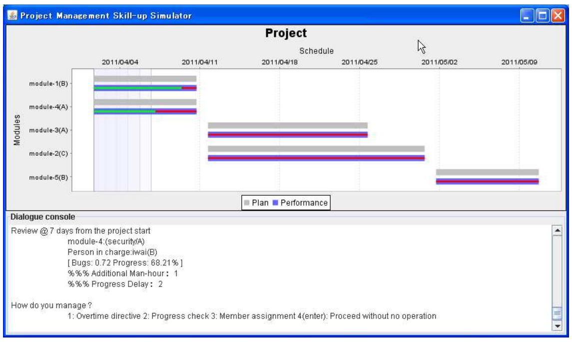

Figure 2: A sample screenshot of simulator interface.

What we explained so far is normal events in a sense, since the project model generates daily dynamics on itself and operations by a trainee are mostly normal reactions if he/she detects the symptoms on project crisis. Therefore a trainee who has knowledge on project management such as PMBOK contents and experience on ordinal projects such as small delay and standard volume of bugs, he/she could accomplish the project management under allowable "QCD" criteria. From educational viewpoints as indicated before, a sense of tightrope walking is necessary to enhance a trainee's skill, which is really our intended goal as a skill-up simulator. To achieve it, we introduces trouble events such as sudden increase of detected bugs at the progress check and much progress delay occurrence under well-ongoing module development so far. The former case may be implicitly induced by planning error based on design specification in the real world project, but we do not care about the cause to manage the progress in this simulator. The latter case may be caused by a member's health condition or motivation in the real world project. However we also do not care about the cause. Trouble events currently designed are shown in Table 2.

3.2.2 Trigger Control of Trouble Events

Both normal and trouble events are injected into the simulator whenever they are triggered. Therefore it is necessary to control them from the principles of "educational contradiction-free" and "learning difficulty adjustment". Heuristic rules seems not to be practical, because the project status is inherently infinite in a sense. However a trainee is expected to notice and manage from "QCD" criteria with "EVM" metrics. Therefore our approach is simply mapping the project status to "QCD" criteria score. As discussed in the section 3.1, Q(uality) is measured by the latent bugs that cannot be perceived before the project end date. Therefore intermediate Q(uality) is measured by the detected bugs not to be removed so far. From this point, the "QCD" criteria score needs reference score to control as a whole. The formula 1 uses possible man-hour as numerator and minimum necessary man-hour as denominator, which means the project status is fair if this score equals 1. If the score is less than 1, the project status is difficult to achieve the initial plan and vise versa.

$$QCD(t) = \frac{N_p * R_d * W_h + C_{overtime}}{C_{mh} + B_{mh}} \quad (1)$$

N_p denotes "no. of project members", R_d does "remaining project days", W_h does "standard working hours", $C_{ovetime}$ does "possible overtime hours", C_{mh} does "remaining coding man-hour", and B_{mh} does "detected bug removal man-hour".

The trigger control of trouble events are based on separation from 1, presumably depending on discrepancy. Assume that tutors also uses this simulator to set attributes of project models, such discrepancy is set in advance by them. During the project simulation, such discrepancy may frequently occur by a trainee's operation. Therefore it is necessary to suppress trouble events. Moreover alleviation on schedule delay or troublesome bugs are injected as change of internal parameters so that numerical difference of mismatch under equal level of the difficulty attribute of "Module" and the skill attribute of "Person" is canceled or daily outbreak of bugs is changed to every few days.

As a whole, the proposed simulator introduces "QCD" criteria score and controls firing and suppression of trouble events by it, and makes alleviation by changing attributes of the project model.

Figure 3 shows our experiment on this trigger control results, which presents "QCD" is almost controled around "1".

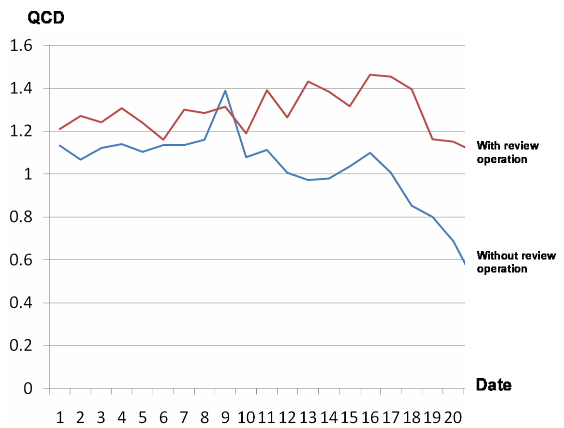

Figure 3: Example of trainee's operation and QCD value.

Table 2: Trouble events.

Event name	Description	Required skill
Bug explosion	Sudden increase of bugs are fired.	A trainee should estimate how much time is necessary to remove and what effect is propagated towards the project.
Bug latent	Detected bugs are almost stable along with the project progress.	A trainee should make frequent reviews and decision on supervised mode.
Progress drop	Progress gradually drops even if mismatch on module difficulty level and person skill level does not exist.	A trainee should make proper decision on member re-assignment or supervised mode.

4 RELATED WORK

Simulation is often used in educational environment and has been discussed several approach in software engineering domain. Research work (Drappa and Ludwig, 2000) concentrates question and answer sessions through showing several status reports on the project. Their system is model-driven approach. However scenario generation is not enough to make a trainee face to unexpected situation, since their model does not include trouble generation, mostly focusing on member assignment task. Member assignment is one of project management tasks and it should be related to schedule delay or bug trouble shooting. When using the learning environment, feedback is significant factor to enhance a trainee's knowledge and skill. Such feedback functionality is discussed in (Mandll-Striegnitz, 2001), which provides several sessions from "introduction", "simulation", "feedback", and second-time sessions to notice lessons learned from the first-time sessions. Same as the work (Drappa and Ludwig, 2000), simulation does not generate various situation, which our approach intends to provide high fidelity situation. Recent serious game approach with high fidelity GUI(Graphical User Interface) is opening the door to new methodology in software engineering education. Research work (Carlos et al., 2007)(Hainey et al., 2011) discuss how to teach requirement collection and analysis in software project management. These works focuses on upper stream in software management life cycle, of course their approach could be applicable to the lower stream in the software management life cycle to some extent. However, our fully functional project model is more fit to attain the progress management phase.

5 CONCLUSIONS

This paper proposed a new framework for project management skill-up simulator, especially focusing on a situation-dependent scenario generation. Our project model is simple but fully functional from project management skill-up. Implementation is still ongoing and will be evaluated by practical project management members in near future. Our future research topic also includes how to guide a trainee by using logs of a trainee's operations.

REFERENCES

Carlos, M., Zapata, J., and Gabriel, A. (2007). Requirements game: Teaching software project management. *CLEI Electrical Journal*, 10.

Drappa, A. and Ludwig, J. (2000). Simulation in software engineering training. In *ICSE 2000*, pages 199–208. ACM.

Hainey, T., Connolly, T. M., Stansfield, M., and Boyle, E. A. (2011). Evaluation of a game to teach requrements collection and analysis in software engineering at tertiary education level. *Journal of Computers & Education*, pages 21–35.

Mandll-Striegnitz, P. (2001). How to successfully use software project simulation for educating software project managers. In *the 31st ASEE/IEEE Fronties in Education Conference*, pages 19–24. ASME/IEEE.

PMI (2009). *The PMBOK Guide*. Project Management Institute, fourth edition.