

Web Application to Manage and Improve Software Development Projects by SEMAT Essence

Yuichiro Senzaki, Siyuan Liu, Hironori Washizaki,
Yoshiaki Fukazawa
Waseda University
Tokyo, Japan
yuitiro.senzaki@ruri.waseda.jp

Hiroshi Kobayashi, Masaharu Adachi
System Information Co. Ltd.
Tokyo, Japan

Abstract— As part of the rapid advances in software engineering, each year a vast amount of new knowledge and ideas are proposed. However, a gap often arises between new ideas and current methods due to a lack of fundamental theory. To bridge this gap, SEMAT (Software Engineering Methods and Theory) Essence has been proposed as the common ground in software engineering. Using SEMAT Essence, developers can track the progress and health of a project more efficiently from various viewpoints. However, SEMAT Essence has some limitations. In practice, only a few tools implement SEMAT Essence. Most of these tools are problematic and do not sufficiently satisfy the requirements for practical developments. Therefore, we develop a tool called OCMS (Online Checklist Management System), which improves existing tools. An experiment where students manage an ET robot contest project using OCMS confirms its effectiveness and demonstrates that OCMS can help developers improve efficiency.

Keywords— SEMAT Essence; SEMAT Kernel; Software Engineering

I. INTRODUCTION

Software engineering is a rapidly advancing field. Numerous methods and practices are proposed each year and incorporated into current projects. The introduction of these new ideas creates a gap between the current and new methods [1]. To bridge this gap, SEMAT (Software Engineering Methods and Theory) Essence has been proposed [1]. SEMAT Essence, which defines the set of most useful elements during a development, identifies the common ground between current and new methods. These elements called alphas can depict the progress and health of a development. Although teams can improve their efficiency and reduce risks using alphas [2], alphas have limitations. Only a few tools can be used to practically implement SEMAT Essence in software development projects. Each tool has issues such as the inability to support multi-platforms, low customizability, etc.

For example, a tool called EssWork Practice Workbench [7], which supports SEMAT Essence, can customize SEMAT Essence's context to adjust a software project. To check other items, SEMAT Essence can be added to create a tailored

checklist. This function is useful because SEMAT Essence only includes minimal items. An organization may have to develop unique check items to manage a software development project. This tool is useful, but expansion can be challenging. After creating a checklist, the progress of a project must be managed. Thus, it is necessary to verify the results of the checked items (e.g., whether the item is complete or in progress). Results verification is beyond the scope of EssWork Practice Workbench. Hence, other tools are necessary to assess a checklist based on SEMAT Essence.

To solve these issues and to increase the usability of SEMAT Essence, we developed a browser-based system that implements SEMAT Essence. This browser-based system is called OCMS (Online Checklist Management System). OCMS not only provides a new choice among existing SEMAT tools, but also offers novel features based on existing software.

The contributions of this paper are:

- We propose a new tool called OCMS to manage the checklist based on SEMAT Essence.
- We confirmed the usefulness of OCMS by conducting experiments.

In this paper, we explain SEMAT Essence and its support tools. Then we propose our tool, OCMS, and subsequently evaluate it.

II. BACKGROUND

A. SEMAT Essence Overview

SEMAT was founded in 2009. The goal was to fundamentally change the method for software development. In the same year, a call for action statement [6] was written to explain the existing problems in software engineering, the rationale for the actions, and suggested solutions [1]. SEMAT Essence was created based on this action statement. It defines a kernel, which includes seven basic elements called alphas, such as opportunity, stakeholders, requirement, etc. Alphas can be

universally reused in most software projects. Each alpha contains some states, where each state has a checklist to describe the alpha status. Using the checklists, the progress and health of a project can be grasped. Additionally, SEMAT Essence defines a simple language to describe methods and practices.

B. Existing SEMAT Tools and Their Problems

Previous tools have implemented SEMAT Essence. The rest of this section introduces three related tools and discusses their merits and demerits. Additionally, existing problems or requirements are described.

1) EssWork Practice Workbench

EssWork Practice Workbench [7] is a development environment focused on customizing alphas and the state of SEMAT Essence. It enables SEMAT Essence to be used in more extensive situations. Organizations often employ a unique checklist to manage a project's progress. Although the checklist may be suitable, it may not be comprehensive. On the other hand, SEMAT Essence may not fully grasp the elements due to abstract expressions. Therefore, SEMAT Essence must be tailored to each project. Practice Workbench can create new elements to adapt to different projects, but once a project is initiated, it does not record the state of alphas. In practice, analyzing historic data is the most crucial method to elucidate new values.

2) Alpha State Explorer

Alpha State Explorer is a tool to help a team track their project via a simple checklist [8]. This tool focuses on recording the state of alphas. Each state of an alpha can be recorded simply, and all alpha states can be verified via a dashboard. Project members can easily follow a project's progress. Each alpha in a project can be saved individually and the results can be shared via email to the team.

Alpha State Explorer has limitations for practical use. First, only iPhone and iPad devices are supported. Additionally, this tool is no longer supported on the latest iOS version [8]. This impacts the collaboration among teammates, especially when multiple platforms are used. Second, sharing is only supported via email. This means that when a user wants to share a result, another software program must be deployed, deteriorating user experience and increasing both time and costs.

3) SEMAT Accelerator

SEMAT Accelerator [9] is a tool that can record the progress of alphas. Furthermore, progress can be tracked as radar charts or bar charts. Moreover, compared to the above tools, the most prominent advantage of Accelerator is the sharing method. Accelerator is a browser-based program. Thus, a person can share results with other team members rapidly using one browser and one account.

SEMAT Accelerator is a good tool to trace the states of alphas. However, new states and checklist items cannot be added into alphas and the status of a checklist item cannot be changed. That is, the status cannot be changed as a project progresses

from unfinished, in progress, and finished. Additionally, Accelerator only supports to the checklist of SEMAT Essence, and cannot be extended to other checklists. In practice, companies always need to combine their own checklist with SEMAT Essence to manage projects.

III. PROPOSAL

A. Requirements

OCMS should meet the following requirements:

- OCMS should be applicable to diverse environments.
- The results of SEMAT Essence should be shared rapidly.
- OCMS should be extendable with other checklists.
- Each project should save its own SEMAT Essence results.
- Alphas and states should be customizable.

B. OCMS Overview

In this chapter, the system structure of OCMS is introduced. Then a specific example is used to explain how to deploy the checklist in OCMS to track a development. Finally, the storage mode of the checklist in OCMS is explained.

OCMS implements and extends SEMAT Essence. It includes a dashboard to grasp the overall progress of a project and a checklist (Fig. 2) to track the progress in detail. Figure 1 overviews OCMS. A web service receives a search request from a user inquiring about the situation of the current project. The request is distributed to the service module. The service module includes the corresponding service to deal with this request. In this case, the request is processed by a checklist service. The checklist service then accesses the database to construct a checklist from the data. Finally, the service module replies to the user with the checklist.

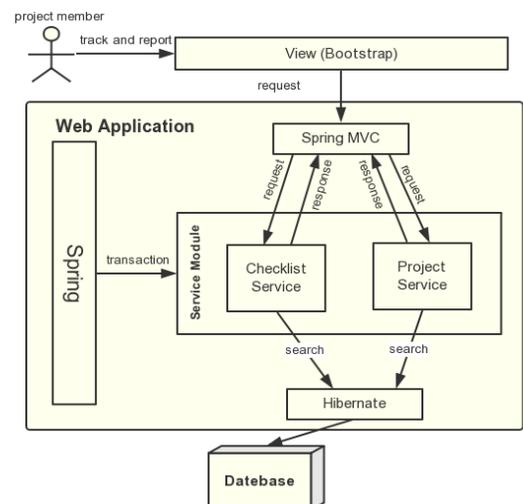


Fig. 1. OCMS System Structure

Figure 2 shows the dashboard of OCMS. In this screen, we can see the progress of alphas at a glance. For example, you can see what alpha is advanced or not from previous sprint. Each state has a score which is calculated by the ratio of achievement of check items. This score will be described later. States are color-coded by score. If its score is 100, its color is green. Similarly, if its score is 0, its color is red. And if its score is not 100 or 0, its color is yellow.

This screen also shows the description of alphas and the status to aim. As shown in Figure 2, the alpha of Opportunity has the description, "The set of circumstances that makes it appropriate to develop or change a software system.", and the goal such as, "A good opportunity is identified addressing the need for a software-based solution", or "A good opportunity has established value".

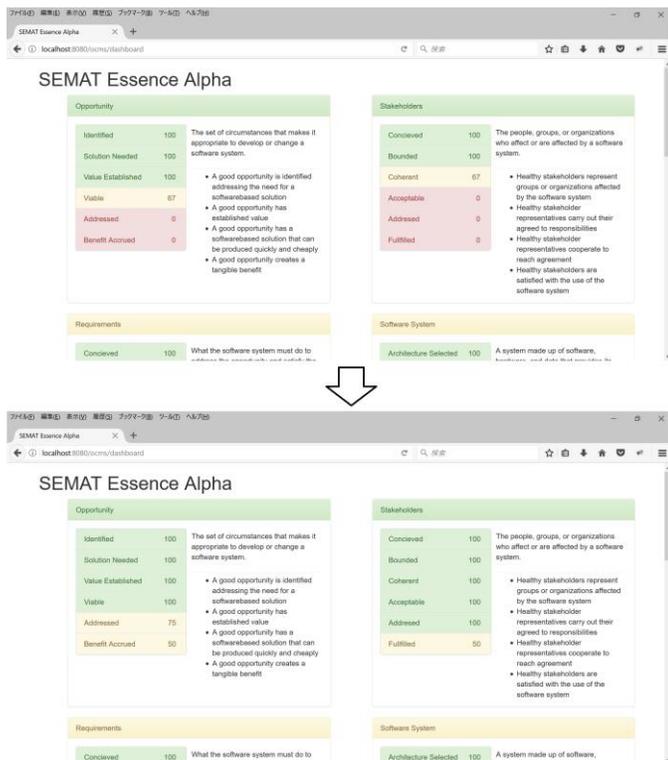


Fig. 2. OCMS dashboard and progress of alphas

Figure 3 shows the constructed checklist. The columns denote areas of concern, alphas, states, and check items implemented from SEMAT Essence. Other columns are extended items. They can be saved by the save button, and the database is updated automatically. Each check item can have some statuses such as finished or solving. Status influence score of state. For example, if all check items are finished, the score is 100, and if all the items are solving, its score is 50. In this screen, you can see more details of alphas or check items. For example, the reach rate can display the progress of project quantitatively.

Comments can be used to annotate reminders to other members. Developers can use this checklist to understand the situation of the current project. In Fig. 3, the comments, "Broke some rules during this sprint. In the next sprint, teams should address." is added against the check item, "Opportunity identified that could be addressed by a software-based solution." In this manner, both the status of the check item and which product is related to the check item are determined. Consequently, SEMAT Essence can be customized, improving its usefulness.

For instance, if a project manager wants to know the situation of his project, he can use OCMS to search the checklist of the current project. Figure 3 also shows part of the results. Because the reach rate of the cell called identified in column of state is 83%, the project manager realizes that something is wrong in the project. Thus, he can continue to use the checklist. The status that corresponds to a check item called 'other stakeholders who share opportunity identified' is not resolved. Therefore, the project manager knows that information sharing with other stakeholders is problematic. In this way, the project manager can adjust the current and future work to address this issue.

Checklist							
Back to Dashboard Save Checklist Delete Checklist							
Area of Concern	Alpha	State	Reach Rate	Check Item	Explanation	status	comments
Customer	Opportunity			Opportunity identified that could be addressed by a software-based solution		finished	Broke some rules during this sprint. In the next sprint teams should care about it.
Customer	Opportunity	1/19	Identified	A stakeholder wishes to make an investment in better understanding potential value		finished	
Customer	Opportunity			Other stakeholders who share opportunity identified		solving	

Fig. 3. Part of the Checklist in OCMS

OCMS can incorporate additional information. However, we are planning to increase its extensibility. Because OCMS's checklist is strongly based on SEMAT Essence, its text cannot currently be altered. In the future, we plan to implement a function to add check items or categories. Finally, OCMS will become a comprehensive tool to manage checklists, especially tree structure checklists. (A tree structure means that the root category has some categories, which have unique sub-categories. Finally, the sub-categories contain the check items.)

C. Contribution

OCMS implements and extends SEMAT Essence. OCMS has five advantages. First, compared with Alpha State Explorer, which only runs on iPhone and iPad, OCMS can span multiple platforms because it is a web-based application. Second, compared with EssWork Practice Workbench, which does not support sharing of the results of SEMAT Essence, OCMS can share results rapidly in a browser. Third, compared with EssWork Practice Workbench, which cannot record the results,

OCMS can save the results in a database and analyze historic data from a database. Forth, compared with Alpha State Explorer, OCMS can customize the status of a check item as finished, unfinished, in progress, etc. Fifth, the database design allows OCMS to be extended to other checklists.

Table 1 compares OCMS and existing tools. The numbers in the top row correspond with the chapter of Requirements. ‘Y’ means that the related problem is solved effectively and efficiently. ‘Y*’ means that the related problem is solved, but not efficiently. ‘n’ means that the related problem is not solved. For example, EssWork Practice Workbench can customize alphas. So, the column is Y. However, the column of saving result is n because it is not implemented. On the other hand, Alpha State Explorer is evaluated as Y* in the column of data because data is shared by email and only used in this tool. OCMS is not only advantageous compared to existing tools, but it also adds new features to improve the efficiency.

TABLE I. COMPARISON BETWEEN OCMS AND EXISTING TOOLS.

	1	2	3	4	5
OCMS	Y	Y	Y	Y	Y
EssWork Practice Workbench	n	n	n	n	Y
Alpha State Explorer	n	Y*	n	Y	n
SEMAT Accelerator	Y	Y	n	Y	n
Microsoft Excel	Y	Y*	Y*	Y*	Y*

IV. EVALUATION

Two research questions were posed to evaluate OCMS.

- RQ1: Does OCMS effectively improve a development?
- RQ2: Does OCMS suggested in this paper make SEMAT Essence easier to use?

To answer these research questions, we conducted an experiment using OCMS. Four students joined an ET robot contest as a team for a several months. In this contest, the team used the same robot, but each member developed unique software. The team wanted to improve the ease of checking the project’s progress. Therefore, we introduced SEMAT Essence and OCMS.

The students answered the before questionnaire (BQ). Then each week for a month, the students used OCMS to reflect on the project. During the experiment, they used OCMS to save the status of the check items. They denoted what was successful and what failed during the week. After the experiment, they completed the after questions (AQ). The students indicated their agreement with the following statements on a four-point scale (Strongly agree, Agree, Disagree, and Strongly disagree).

- BQ1: I have considered the environment surrounding software development such as stakeholders in my development to date.

- BQ2: I should consider the surrounding environment more.
- BQ3: I have developed my software to date by considering the whole picture of development such as requests and architectures by tool.
- BQ4: I should consider the whole picture more.
- BQ5: I have developed with awareness of working efficiently and employing support tools.
- BQ6: I should consider working efficiency and support tools more.
- AQ1: I am able to develop software while remaining conscious of the environment surrounding software development such as stakeholders by tool.
- AQ2: I have noticed specific improvements in the environment.
- AQ3: I am able to develop software while being conscience of the whole picture of solution development such as requests and architectures by tool.
- AQ4: I have noticed specified improvements in the whole picture.
- AQ5: I am able to develop software while being conscience of work efficiency, support tools, etc. using OCMS.
- AQ6: I have noticed specified improvements in work efficiency and use of support tools.
- AQ7: It is easier to grasp the situation of my project by OCMS.
- AQ8: OCMS make it easier to use SEMAT Essence.

V. RESULTS AND DISCUSSION

A. RQ1: Does OCMS effectively improve?

Figure 5 shows the questionnaire results. From the answers of BQ 1~6, although students were conscious of various points in software development, they hoped for additional improvements. From the results of AQ 1~6, their conscious improved. Concrete improvement was observed using SEMAT Essence and OCMS.

B. RQ2: Does OCMS make SEMAT Essence easier to use?

OCMS made it easier to grasp the situation of a software development project and to use SEMAT Essence, as demonstrated by the results of questionnaires AQ7 and 8 in Fig. 5. On the other hand, the experiment with the tool revealed several issues. Most students said, “The contents of SEMAT Essence were difficult to understand”. For example, the alpha for Way of Working has a checklist item, ‘Communication be-

tween members is open and honest'. However, when the students read this alpha, they thought it meant "how to measure whether this alpha is achieved". In this case, the checklist should define the conditions. The tool should record this definition.

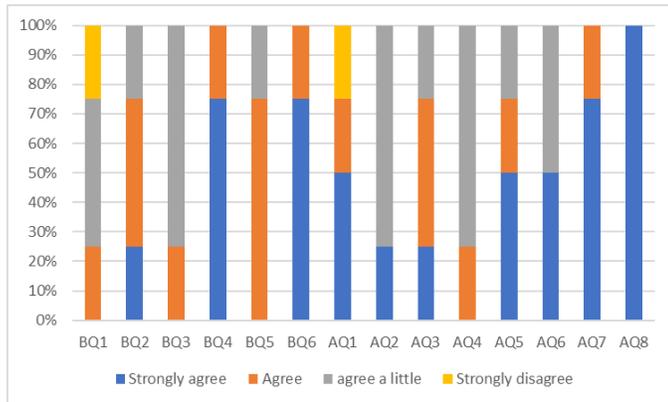


Fig. 4. Questionnaire results

This experiment has some threats to validity. First, we introduced SEMAT Essence and OCMS in the middle of the project. It is possible that the improvement in the development is due to a different reason such as the participants' skills evolve as the project progresses, realizing development efficiency. On the other hand, our tool was used for part of a project. For a more accurate evaluation, the experiment should cover a longer time period and we should conduct a control experiment.

VI. CONCLUSION

SEMAT Essence provides a framework to solve the gap between several software development methods. SEMAT Essence also can track the progress of project by tracking alphas. We developed a tool called OCMS, which makes SEMAT Essence more useful because existing tools have issues in actual software developments. Herein we examined whether OCMS makes it easier to implement SEMAT Essence and improve a software development project. The experiment reveals that SEMAT Essence and OCMS are more effective for software development.

In the future, OCMS should be more user-friendly. For example, the ability to save and display the interpretation of a checklist item should be added for non-SEMAT experts. Furthermore, other methods which using SEMAT Essence should be implemented in OCMS. For example, SEMAT Essence can be used for root cause analysis [10]. On the other hand, we should do more additional case studies. To evaluate OCMS more fully, additional experiments are necessary.

REFERENCES

[1] Ivar Jacobson, Pan-Wei Ng, Paul E. McMahon, Ian Spence, Svante Lidman, "The Essence of Software Engineering: The SEMAT Kernel," *Queue – Networks*, vol. 10, October, 2012.

[2] Ivar Jacobson, Pan-Wei Ng, Ian Spence, Paul E. McMahon, "Major-league SEMAT—Why Should an Executive Care?," *Queue – The Internet*, vol. 12, February, 2014.

[3] Cécile Péraire, Todd Sedano, "State-Based Monitoring and Goal-Driven Project Steering: Field Study of the SEMAT Essence Framework," *Companion Proceedings of the 36th International Conference on Software Engineering*, pp. 325-334, 2014.

[4] Cécile Péraire, Todd Sedano: Essence Reflection Meetings, "Field Study. In: EASE '14 Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering," Article No. 25, ACM, London, England, United Kingdom.

[5] Pan-Wei Ng, Shihong Huang, "Yumei Wu: On the Value of Essence to Software Engineering Research: A Preliminary Study", In: 2013 2nd SEMAT Workshop on a General Theory of Software Engineering (GTSE), pp. 51-58. IEEE, San Francisco, CA, USA (2013).

[6] SEMAT call for action, <http://semat.org/ja/call-for-action>, last accessed 2017/07/26.

[7] Brian Elvesæter: Extending the Kernel with Practices to Create Methods. In: Essence Workshop. Berlin, Germany (2013).

[8] ALPHA STATE EXPLORER APP | AGILE SERVICES, <https://www.ivarjacobson.com/alpha-state-explorer-app>, last accessed 2017/10/24.

[9] Graziotin, D. & Abrahamsson, "A Web-based modeling tool for the SEMAT Essence theory of software engineering", *Journal of Open Research Software* 1(1), e4 (2013).

[10] Hironori Washizaki, "Pitfalls and Countermeasures in Software Quality Measurements and Evaluations," *Advances in Computers*, Vol. 107, pp.1-22, Elsevier, 2017.