A CAST Analysis of an Accident of Software Project

Shigeru Kusakabe[1,*] and Azuma Miwa[2]
[1] University of Nagasaki, Japan
[2] SCSK Corporation, Japan

ABSTRACT

We report a CAST analysis of a software project by a team across the primary and subcontractor companies focusing on a kind of safety, psychological safety. By using CAST, we can understand the context of the behaviours leading to the unintended event and avoid personal blames to keep psychological safety. The project was a software maintenance project, in which an online trading system went down in the production migration, the final phase before the system release. The end users had no damages in this case, but some members were going to lose respects from other members and stakeholders. The system down was caused by a defect, and the first investigation report simply blamed the subcontractor members related to the defect injection and leakage. The senior manager, who cares to build and maintain psychological safety in the team, wanted to conduct a deeper analysis to avoid personal blames. The CAST helped to uncover that process changes over time from the originally defined one contributed the injection and leakage of the defect. It turned out the process owner and manager at the primary contractor side had a part of the responsibility for the defect leakage. The senior manager made this conclusion shared among team members and succeeded to avoid losing psychological safety. We conclude the CAST based on STAMP is useful in analyzing issues of software project management such as psychological safety.

Keywords: Software Process; Emergent Property; Psychological Safety.

1. INTRODUCTION

In this paper, we report a CAST analysis of a software project by a team across the primary and subcontractor companies focusing on a kind of safety, psychological safety (Edmondson, A., 1999). The domain of the project is a financial domain and software also plays an important role in the system of the domain. Software defects in such a system may have severe impacts on our society, and we ideally need to develop defect-free software. As the defect was found before the final production migration, they did not incur unintended losses from an end user's point of view. However, some members were about to lose trust from key stakeholders, and the team had damage on psychological safety, which is as severe as the losses of trust from the customer and executives in this case. In the early investigation, a defect was injected in a part of the code base outside of the project scope and leaked through some project phases due to the immaturity of the project members in the subcontractor company.

The senior manager of the project, who cares to build and maintain psychological safety in the organizations the team belonged to, wanted to conduct a deeper analysis to uncover the reasons of the unsafe behaviour leading to the event. He decided to use CAST to avoid conclusions of personal blame to damage psychological safety of the teams and organizations. As a result of Google's Project Aristotle indicates, psychological safety is very important for the success of software projects (Duhigg, 2016). Although there are other norms that seem important as well, such as making sure teams had clear goals and creating a culture of dependability, the Google's result indicated that psychological safety was critical to making a team work. In the relatively advanced software process

* Corresponding author: kusakabe@sun.ac.jp

area, modern agile process area, psychological safety is also regarded as one of the most important aspects of the agile process[†]. Some evangelists think psychological safety so important that they named activities related to psychological safety as “Anzeneering” (“Anzen” is a Japanese word for safety.)

We mainly report the analysis from a viewpoint of the primary contractor to which the senior manager belonged, while there exist other key stakeholders such as the customer and the subcontractor. He needed to manage the projects and the teams in terms of multiple aspects such as productivity and psychological safety. He realized they behaved as a system and his role included a role of controller of the emergent properties in the system as indicated in Figure 1.

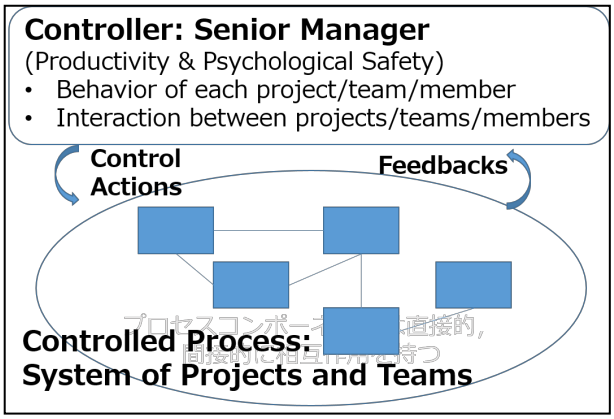

Figure 1: Controlling emergent property

The senior manager decided to seek the flaws of the control related to the event and take measures to prevent similar events by using methods other than their conventional ones. We used the System-Theoretic Accident Model and Process (STAMP) (Leveson, 2012) in modelling the control and management structure of the project. We analyzed this case with Causal Analysis using System Theory (CAST) (Thomas, 2017) through the following steps:

1. Identify the system-level hazard(s) involved in the loss.
2. Identify the system safety constraints and system requirements associated with that hazard.
3. Document the safety control structure in place to control the hazard and enforce the safety constraints.
4. Determine the proximate events leading to the loss.
5. Analyze the loss at the physical system level.
6. Analyze higher levels of control to determine how and why each successively higher level contributed to inadequate control at the current level.
7. Examine overall coordination and communication contributors to the loss.
8. Determine the dynamics and changes in the system and the safety control structure relating to the loss and any weakening of the safety control structure over time.
9. Generate recommendations.

In the rest of the paper, section 2 introduces the software maintenance project in which the event occurred. This section includes the CAST step 2 and 3 explained above. Section 3 mainly focuses on the CAST step 4, 5 and 6. Section 4 explains the rest of the CAST steps. Section 5 makes a conclusion.

[†] http://modernagile.org/

2. SOFTWARE MAINTENANCE PROJECT AND RELATED STRUCTURE

2.1. Background of the Project

The senior manager of the primary contractor company organizes and manages several teams, consisting of members from the primary contractor company and the subcontractor company. The teams are in charge of developing and maintaining an online financial system, consisting of several subsystems. They cover from development, product migration, operation and to maintenance of 24 hours 365 days. Each member basically belongs to a team and works for projects in need. Figure 2 shows a schematic view of the combination of the projects and teams, while the classification of the teams is not the real one due to some business reasons. There are various requirements to them, such as periodical repair of existing functions, the addition of new service, improvement of interface, modification according to the legislative revision. Each project is different from other projects in terms of several aspects such as frequency, duration, amount of work. Human resource allocation becomes difficult if the skill and knowledge of each member are limited to a single subsystem area. Each member is expected to cover a wide range of skills and knowledge and sometimes encouraged to work for other teams.

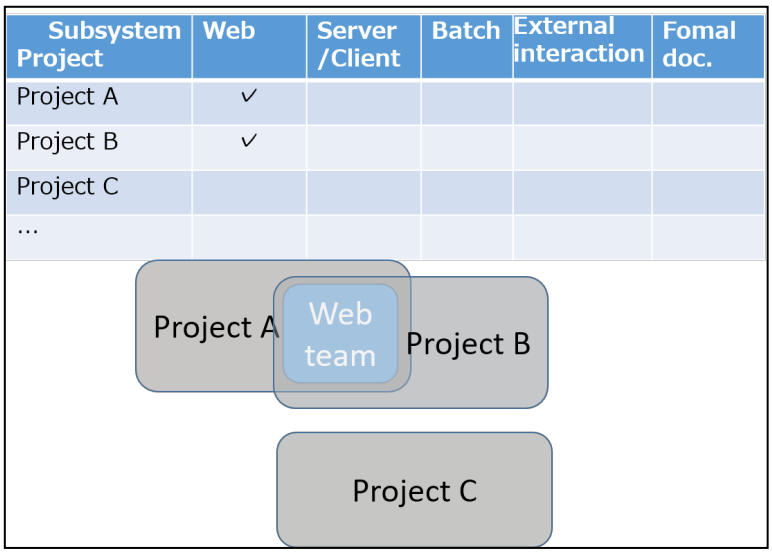

Figure 2: Schematic view of the combination of projects and teams

The team reported in this article was one of the teams explained above. The team members belong to the primary contractor or the subcontractor. The senior manager, project managers, team leaders, and team members belong to the primary contractor, and the managers organize a committee with the senior manager for high-level decision making. The development level sub-team leaders and developers belong to the subcontractor. Conceptually, they form a hierarchical structure as shown in Figure 3, although they do not always form a so-called “tree structure” in the actual cases.

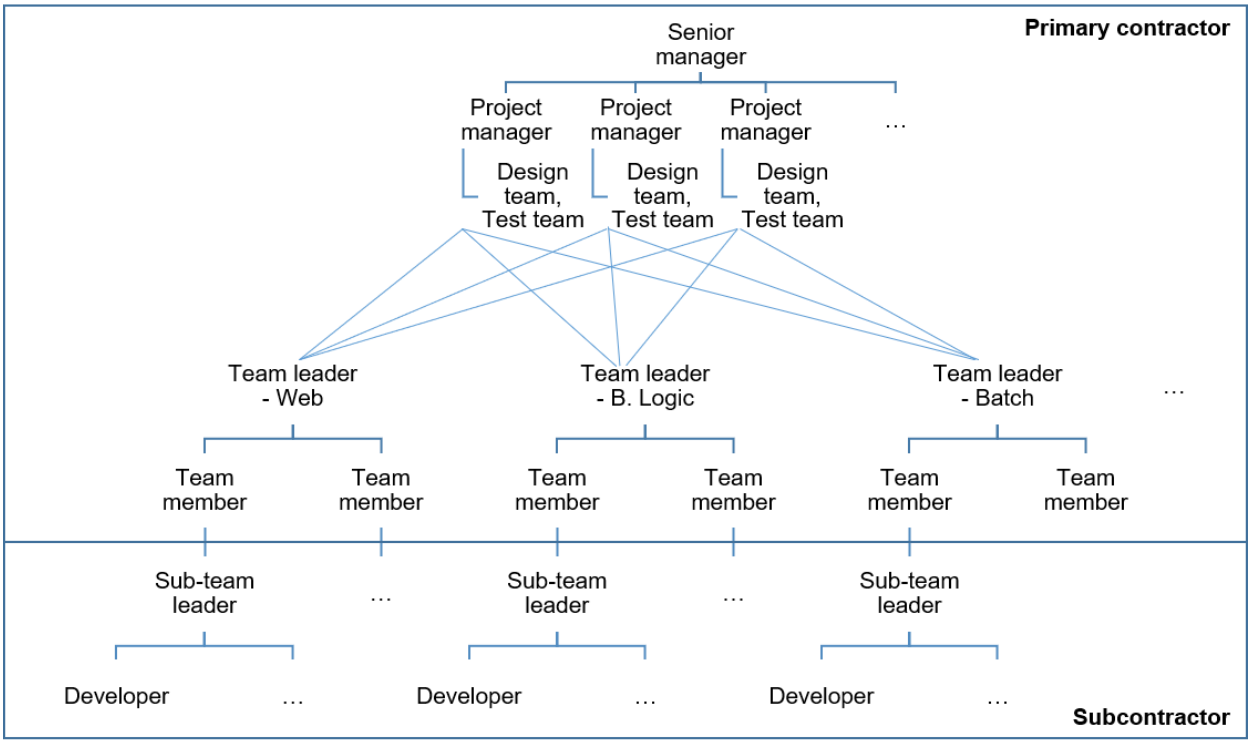

Figure 3: Schematic view of organizational structure

2.2. Early CAST Steps

The system safety requirements, in this case, were to avoid injecting new critical defects and find existing critical defects in deliverables while satisfying customer's needs and keeping QCD (quality, cost, and delivery schedule). The safety requirements were reflected in measures to prevent the injection of new defects and find injected defects. The project leader in the primary contractor needs to make requirements detailed enough to derive the appropriate design while reflecting the customer's needs. The team leaders and team members are responsible for implementing the requirements for each project. They examine the requirements, make the specification and design with experts and share them with the development team in the subcontractor. As multiple projects run concurrently while they conduct a project which adds and modifies the code base, the scope of the target system in each project is determined with care to avoid unexpected interferences.

The control structure related to this event was rather straightforward. Figure 4 shows an outline of the control structure. In the case, a code region out of the scope of the project's specification was modified at the bottom level control loop L_1 in the control structure. The unsafe behaviour seemed quite naïve within the simple control structure. The question here is why measures in the control loops to prevent injection of new defects and find existing defects did not work as expected, in addition to the reason for the unsafe behaviour itself.

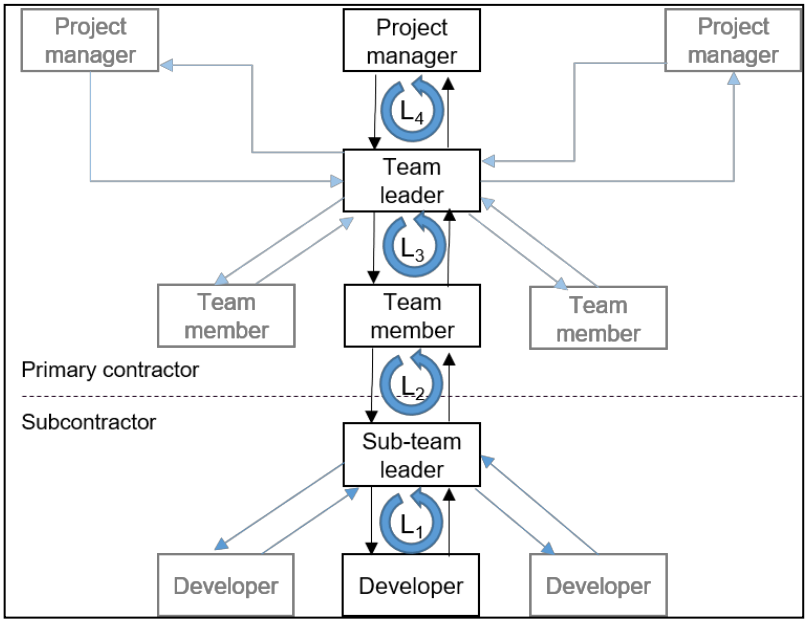

Figure 4: Outline of control structure

3. CAST AT EACH LEVEL

In order to avoid personal blame, we analyze the reasons why it seemed reasonable for the engineer to make such an unsafe behaviour at that time and why the countermeasures placed in the control structure did not work as expected.

3.1. Process Model of the Developer

Engineers generally try to make useful artefacts and pursue quality of the artefacts. The engineer of this case also modified the code base outside of the project with his intention to make things better. The team usually have a backlog of minor items for the code base in the team, and the engineer already had experiences to fix some items in the backlog but outside of the scope of the specific project during the official project period. It was not unusual for the engineer to modify the code base outside of the currently running project. He did not think he did an undesired behaviour at that time

In the defined process for the implementation phase, unit testing is required after adding and modifying the code as a mandated activity. The engineer followed the process except for the backlog item, the code modification outside of the project scope. This unsafe exception was caused by the inappropriate process model of the engineer at that time.

3.2. Process Model of the Team Leader

In the defined process, reviews and testing are included as measures to prevent the injection of new defects and find injected defects. Reviewing work products is one of the important tasks of the sub-team leader in the subcontractor. As a result of a code review, the team leader recognized the modification outside of the project scope and pointed out the deviation to the engineer. After hearing the engineer's past experiences on the backlog items, the sub-team leader approved the modification. As another issue, the sub-team leader pointed out the modified code outside of the project scope was not covered in testing. After hearing the engineer's appropriate explanation that it would be covered in the testing activities in the higher levels, the sub-team leader approved this lack of coverage in testing.

The reason for these unsafe decisions was the lack of appropriate defined procedure after recognizing the deviation related to the backlog item. There were no explicit descriptions on this matter in the defined process. The sub-team leader needed to compensate for this shortage from somewhere. As the engineer is senior to the team leader in age, the team leader considered the engineer's decision was approved one like an order from the higher level in the control structure and did not consider the need of sharing this decision with controllers in the higher level of the control structure.

3.3. Process Model at Higher Levels

At the higher levels, reviews and testing were in the defined process, and they were recognized as important activities. The code base in the project scope was actually reviewed and covered in testing activities. They thought the issues on the backlog items be voluntarily reported from the sub-team leader while this expectation was not clear in the defined process nor in the communication with the sub-team leader.

4. OVERALL COORDINATION AND COMMUNICATION

4.1 Overall Coordination and Communication

This section examines overall coordination and communication contributed to the event. There existed the gap regarding the process of handling the backlog items. The process model of the controller in the higher level of the control structure had no explicit awareness of the backlog as shown in Figure 5.

Team leader and member in the primary contractor
Safety-Related Responsibilities:
- Instruct the specification including the scope
- Take measures to prevent injection of defects
- Take measures to find injected defects

Context:
- Paid special attention for new sub-leader from the defined process view

Unsafe Decisions and Control Actions:
- Not changing the defined process after delegating parts of the process

Process Model Flaws:
- Expecting question assuming sub leader knows what he does not know

Figure 5: Analysis of process model of the team leader and member in the primary contractor

The process model of the sub-team leader in the subcontractor company had no knowledge on the handling of the backlog items as shown in Figure 6. He was a newcomer to the team while the former sub-team leader seemed to have the process model in handling the backlog items.

Sub-team leader
Safety-Related Responsibilities:
- Follow the specification including the scope
- Take measures to prevent injection of defects
- Take measures to find injected defects

Context:
- Had no experiences on backlog items as a new member

Unsafe Decisions and Control Actions:
- Approval of backlog handling in an inadequate way
- Approval of inadequate test coverages for the fixed backlog

Process Model Flaws:
- Not aware of the necessity of sharing the issue of backlog handling.
- Biased by the idea of seniority.

Figure 6: Analysis of process model of the sub-team leader in the subcontractor company

The process model of the engineer lacks awareness of sharing issues outside of the defined process and the project scope as shown in Figure 7. Detailed investigation revealed the former sub-team leader was taking care of sharing this kind of bottom-up issues with the higher level of the control structure. That kind of considerable attitude made things tacit in a sense.

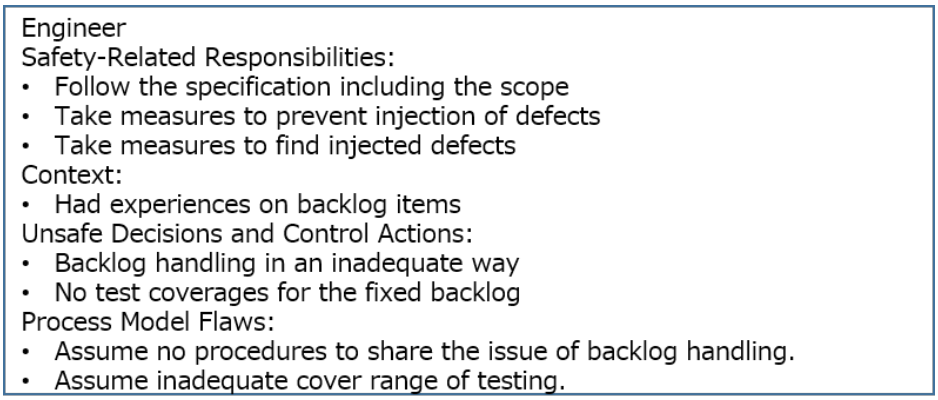

Figure 7: Analysis of process model of the engineer in the subcontractor company

4.2. Dynamics and Changes in the System and the Safety Control Structure

This section discusses the dynamics and changes in the system and the safety control structure relating to the event over time.

Collaborative work between the primary contractor company and the subcontractor company started about ten years ago. In the beginning, the subcontractor leaders mastered their job by detailed advice from members of the primary contractor company. They gained tacit knowledge such as the way of communication and procedures in the process model, as well as visible knowledge such as explicit process definition, deliverables and development environment and those that are difficult to see such as communication methods.

Members of the subcontractor company mastered the job and reached the level where they could work according to the same process as the primary contractor company. After the subcontractor reached this level, the primary company gradually transferred the roles to the subcontractor. During this transfer, some of the important issues became tacit knowledge out of the defined process in both sides. Along with this transferring process, the recent trend to promote mobility of human resource between different subsystem teams migrated the organization to a hazardous state.

5. CONCLUDING REMARKS

We reported a CAST analysis of an unintended event in a software maintenance project conducted by a team across the primary contractor and subcontractor companies. As the early investigation of this event started to threat psychological safety of the team members in the subcontractor, the senior manager caring psychological safety in the team tried CAST as a deeper analysis to uncover the reasons of the unsafe behaviour leading to the event. The CAST helped to uncover that process changes over time from the originally defined one contributed the injection and leakage of the defect leading to the event. It turned out the process owner and manager at the primary contractor side had a part of the responsibility for the defect leakage. The result of the CAST analysis was useful to make stakeholders understand the reasons why this event happened and recover psychological safety in the project team. We conclude CAST based on STAMP is useful in analyzing issues of software project management such as psychological safety. However, a very limited number of project members could make STAMP models and conduct CAST analysis. We will continue to use CAST for other software project management cases to show the feasibility as our future work

REFERENCES

Duhigg, C. (2016, Feb. 25). What Google Learned from Its Quest to Build the Perfect Team. *The New York Times Magazine*. Retrieved January 30, 2019, from https://www.nytimes.com/2016/02/28/magazine/what-google-learned-from-its-quest-to-build-the-perfect-team.html

Edmondson, A. (1999). Psychological safety and learning behavior in work teams, *Administrative Science Quarterly, 44:2*, 350

Leveson, N. (2012). *Engineering a Safer World*. Cambridge, MA: MIT press.

Thomas, J., Malmquist, C. (2017). Learning from Accidents that are a Consequence of Complex Systems, *Presented at the 48th Annual International Society of Air Safety Investigators*. San Diego, CA, August 22-24, 2017, Retrieved January 30, 2019, from http://www.isasi.org/Documents/library/technical-papers/2017/Thurs/4.%20Learning%20from%20Accidents%20that%20are%20a%20Consequence%20of%20Complex%20Systems.docx