A CAST Analysis of an Accident of Software Maintenance Project

Shigeru Kusakabe[1], and Azuma Miwa[2]

[1] University of Nagasaki, Japan

[2] SCSK Corporation, Japan

Nov, 2, 2018

Agenda

- Introduction
- Background Information
- CAST (respective)
- CAST (Holistic)
- Concluding Remarks

Event

Online trading system went down in the production migration, the final phase of a maintenance project.

- No damages on the end users.
- Loss of "**psychological safety**" in the project members.

Preliminary report

- Subcontractor side
  - Developer: Rough, touched code outside the specified code scope. Immature, injected & leaked defects.
  - Sub-team-leader: Immature, allowed process violation, and failed to check the defects.
  - (Leading to loss of respects, …)
- Primary contractor side
  - None

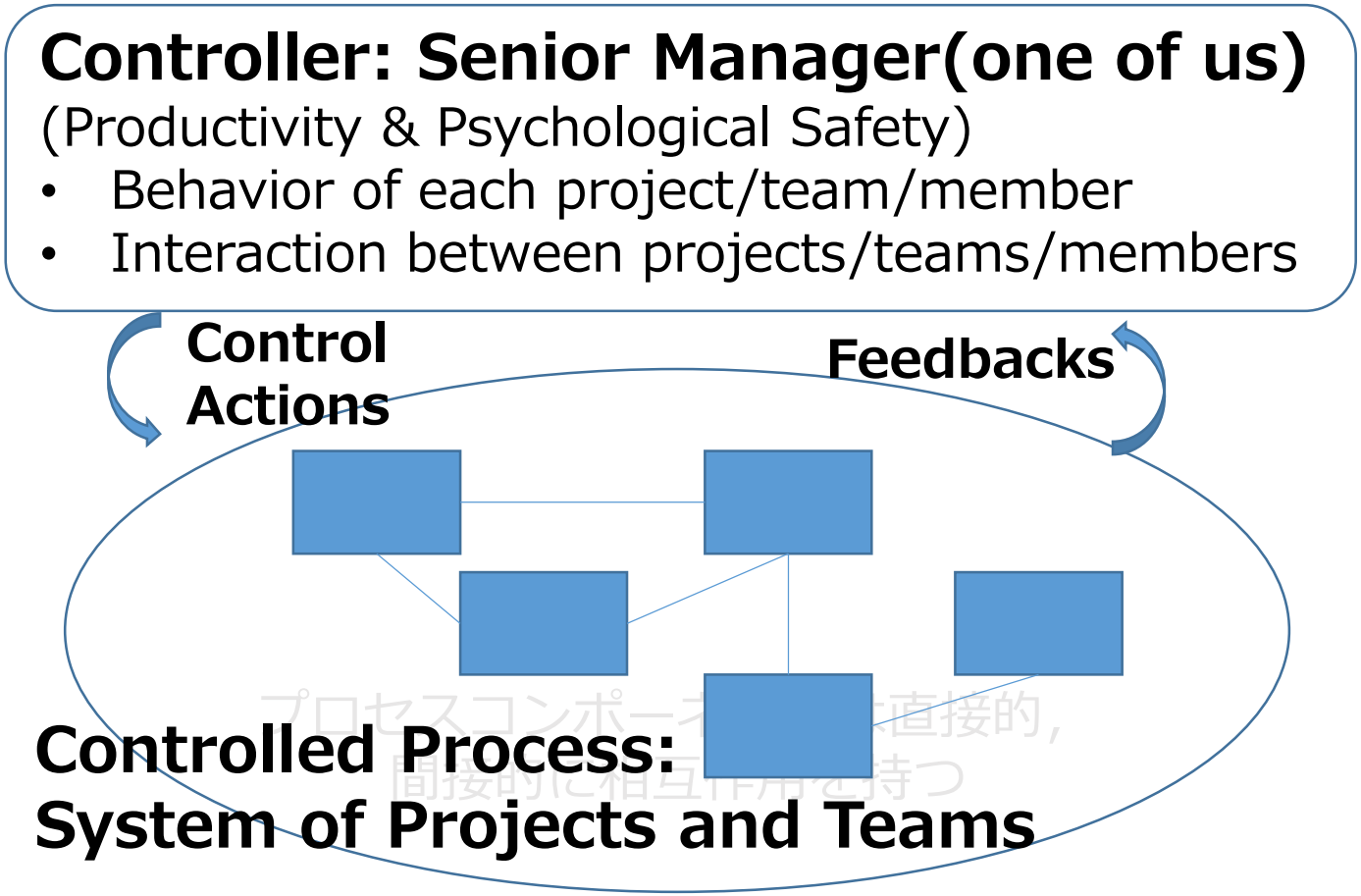
Psychological Safety

- Important for success of software projects
  - Team members feel accepted and respected, being able to show and employ one's self without fear of negative consequences of self-image, status or career.

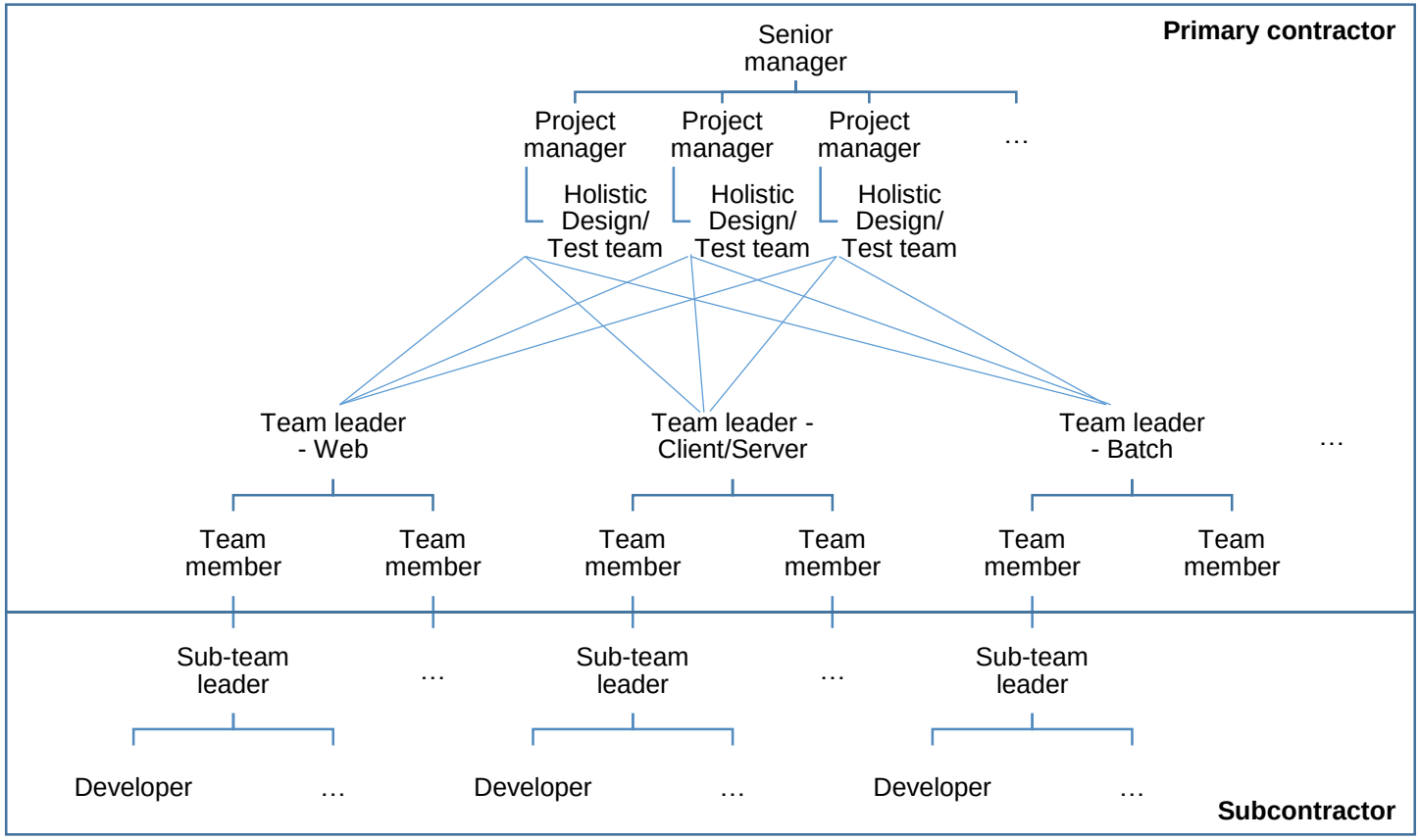
Example:

- Google’s Project Aristotle
- One of the guiding principles of modern agile process
  - Make People Awesome
  - **Make Safety a Prerequisite**
    - ✓“**Anzen**eering” (“Anzen” is a Japanese word for safety.)
  - Experiment & Learn Rapidly
  - Deliver Value Continuously

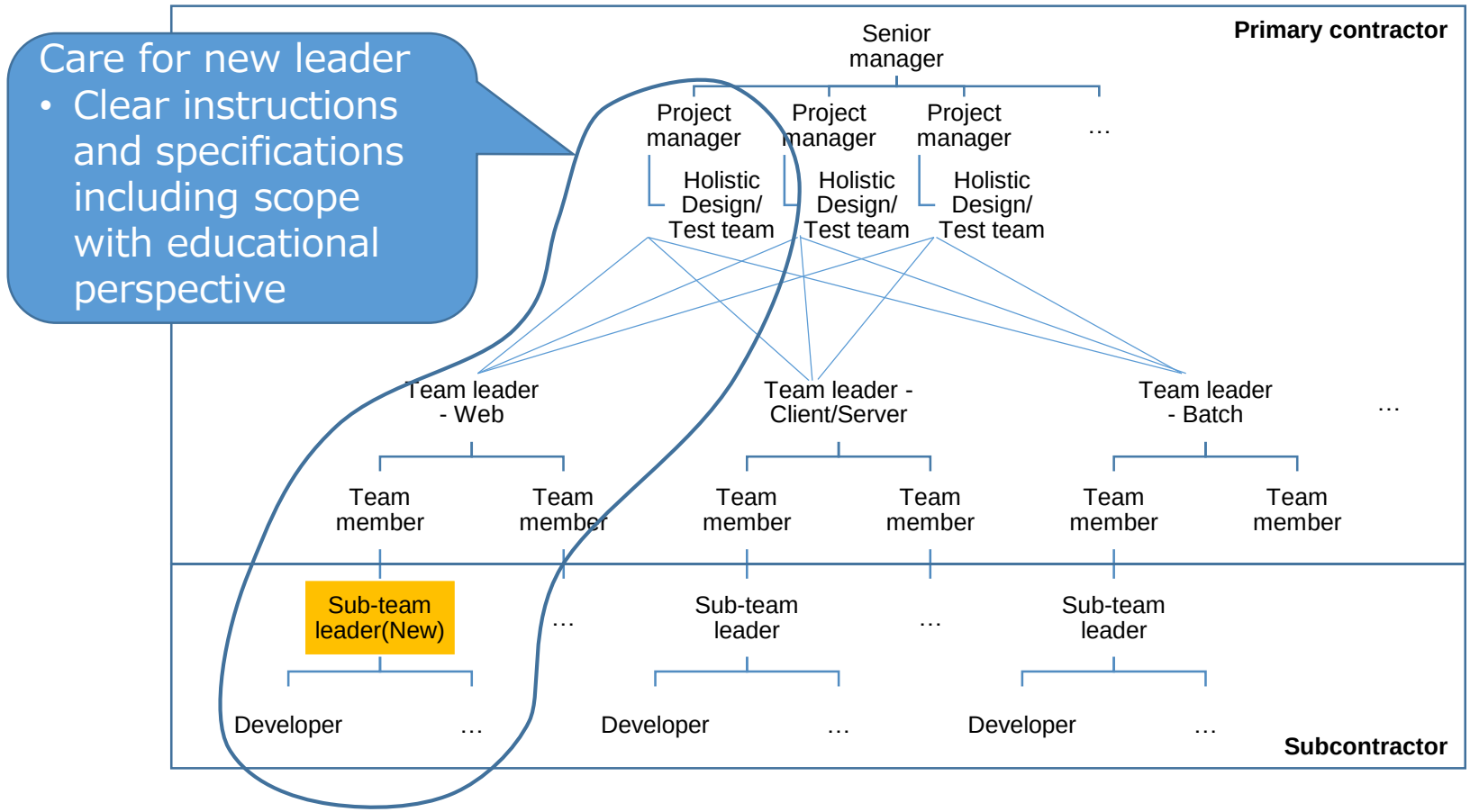
Controlling Emergent Property



Structure of projects across companies



Project info: Educational Consideration



CAST Steps

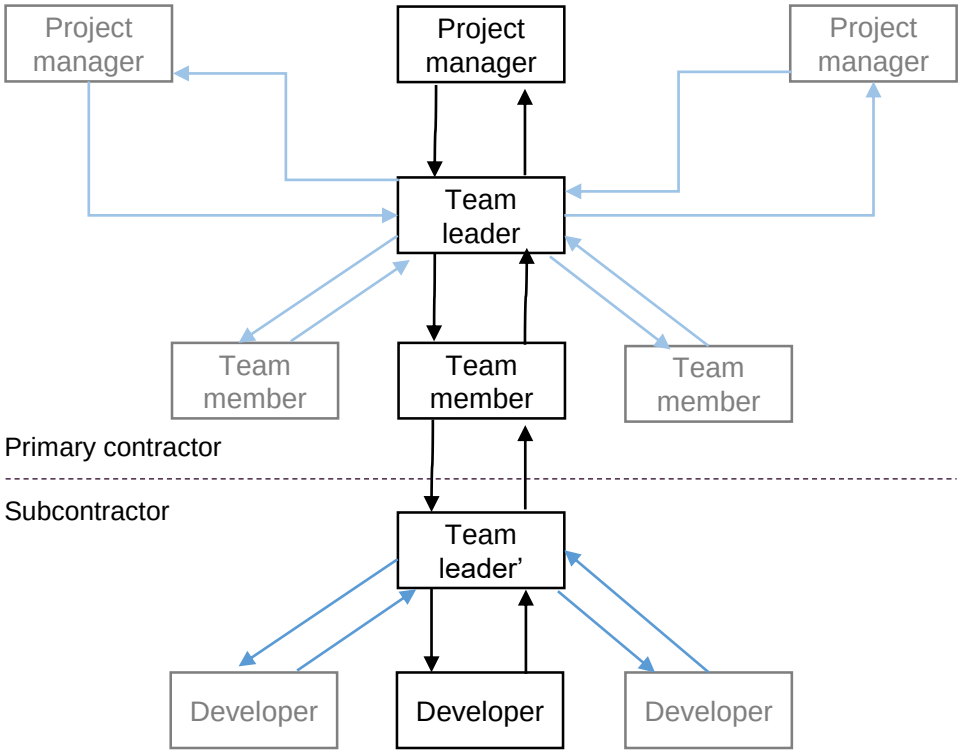
1. Identify the System-level Hazard(s) involved in the loss
2. Identify the control structure in place to control the hazard
3. Determine the proximate events leading to the loss
4. Analyze the loss at the physical system level
   a. Identify the physical controls and equipment involved
   b. Identify any physical safety requirements and constraints meant to prevent this accident
   c. Identify any failures or inadequate controls in the physical equipment
   d. Identify contextual factors that explain the physical failures or inadequate controls
5. Analyze higher levels of control to determine how and why each successive higher level contributed to inadequate control at the current level
   a. Identify the safety-related responsibilities for the next higher level of control
   b. Identify the unsafe decisions and control actions
   c. Identify process model flaws (beliefs) that explain the unsafe decisions and control actions
   d. Identify contextual factors that explain why the behavior seemed appropriate at the time
6. Examine overall coordination and communication contributors to the loss.
7. 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.
8. Generate recommendations.

Step1: Identify hazards

- Scope management, failed
  - Modification of code outside of scope
- Verification, failed
  - Modification w/ defect was not covered in test

Step2 Identify the control structure

- Review
- Testing



Step3 Events leading to the loss

- Developer modified code outside of scope.
- He injected defects in the modification.
- He made mistake, not including the modified part in testing.
- Modification of code outside of scope was allowed
- Engineer injected defects in the modification
- Engineer made mistake, not including the modification
- Testing not including the modification was authorized
- ...

Step4: Analyze loss at low level

Developer

- Code modification outside of scope
- Injected defects and failed to find/remove them

Safety-Related Responsibilities:
- Follow the specification including the scope
- Take measures to prevent injection of defects
- Take measures to find and remove injected defects

Context:
- Had experiences on backlog items, to do things in the base code
- Supervised by the younger and less experienced leader

Unsafe Decisions and Control Actions:
- No test coverages for the fixed backlog
- Backlog handling without considering the rest of process

Process Model Flaws:
- Misunderstood cover range of testing.
- No idea of necessary formal procedures in handling backlog.

Step5 Analyze higher levels of control

Sub-team leader

- Pointed out the scope violation, but approved
- Pointed out test cover leakage, but approved

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 comer

Unsafe Decisions and Control Actions:
- Approval of backlog handling without sharing the information
- Inadequate approval of no coverage for the modified backlog

Process Model Flaws:
- Biased by the idea of seniority.
- No process knowledge in terms of backlog handling.

Step5 Analyze higher levels of control

Team leader/member in primary contractor

- Care for subcontractor along with defined process

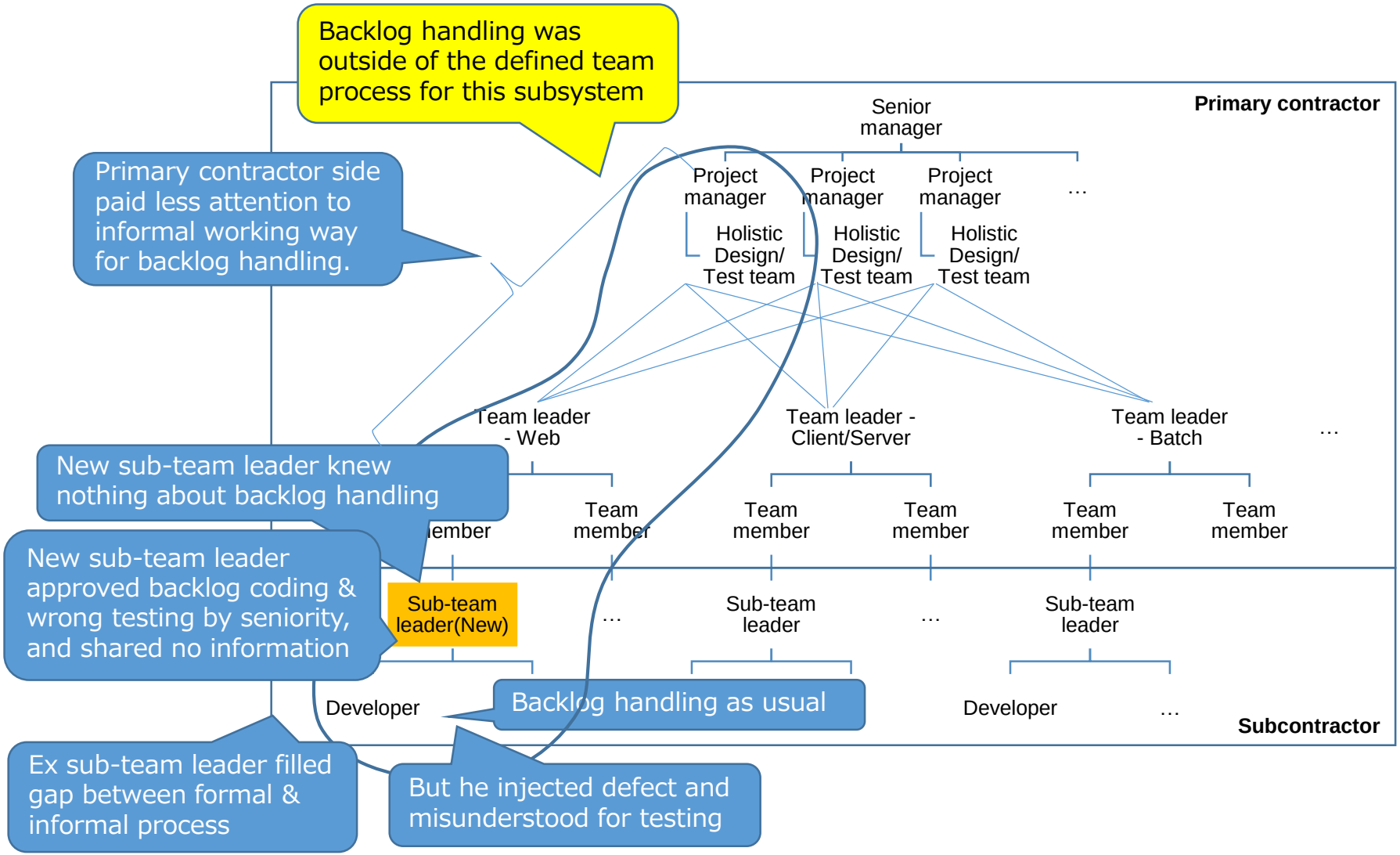
Safety-Related Responsibilities:
- Instruct the specification including the scope
- Take measures to prevent injection of defects
- Take measures to find injected defects

Context:
- Special care for new sub-leader along with the defined process

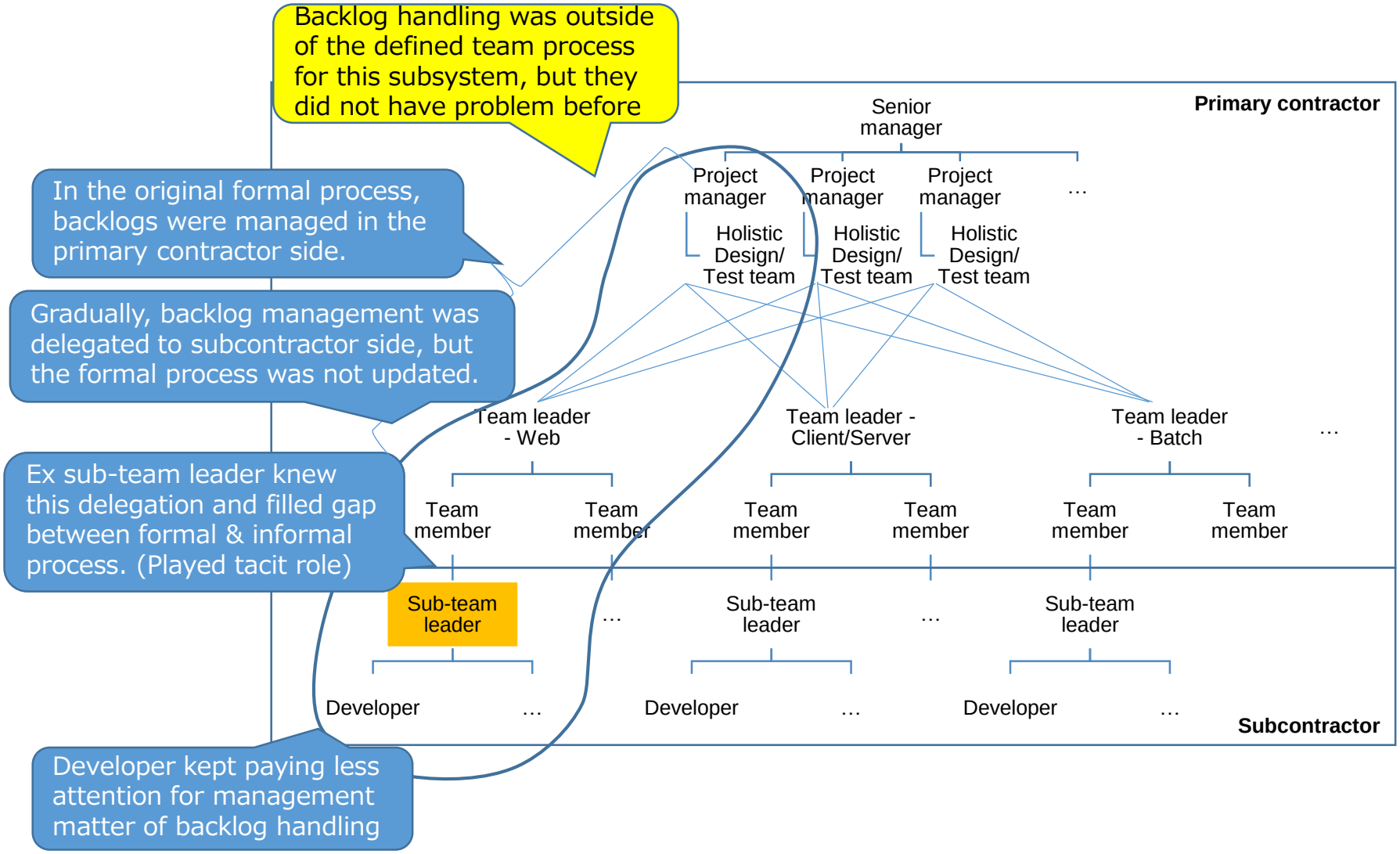
Unsafe Decisions and Control Actions:
- Passive attitude for non-project-specific matters, like backlog items
- No change management for process while delegating some parts

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

Step6: Overall coordination



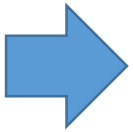
Step7: Dynamics and Changes



Step8: Recommendations

Preliminary report

- Subcontractor side
  - Developer: Rough, touched code outside the specified code scope. Immature, injected & leaked defects.
  - Sub-team-leader: Immature, allowed process violation, and failed to check the defects.
  - (Leading to loss of respects, …)
- Primary contractor side
  - None



- Subcontractor side
  - Developer: need to improve coding & testing skill.
  - Sub-team-leader: avoid non engineering judging grounds such as seniority.
- Primary contractor side
  - Need change management for formal process.
  - Need to pay attention to tacit matters, such as backlog management in this case, if keep facilitating OJT in multiple subsystems.

Concluding Remarks

CAST was useful in

- avoiding personal blames.
- understanding and sharing the reason of decision making and behavior.
  - Helped to maintain psychological safety.
- developing constructive counter measures and improvement plan.
  - Need improvements in both side.

- However, it is not easy ...
  - Not everyone can apply CAST