Questionnaire

*Notation:

1. This questionnaire is composited of two parts. Part 1 is the questions from Comparative Survey with some modified questions, which is specific for the evaluation of research question 1. Part 2 is the questions extended from Comparative Survey, which is specific for the typical safety environment to evaluate the research question 2. We mix them for the better understanding and comparison.

2. Each question should be answered with the number 1 to 5.

1	2	3	4	5

Negative Positive

1. Team Work

(1) Team Work Composition

(Part 1)

□ The whole team, including Scrum Master and Product Owner, is no more than 12 people. (deleted)

□ Team members are kept as long as possible.

□ Specialists are willing to work outside their specialists to achieve team goals.

□ Everyone required to go from requirements to finished system is on the team.

□ People are not on more than two teams.

(Part 2)

○ Safety expert is kept as long as possible with the team members.

○ Safety expert is willing to help other members to achieve team goals.

○ Safety expert is on the team from the requirements to finished system.

○ People including safety expert are not on more than two teams.

(2) Team Work Management

(Part 1)

□ Management rarely changes the team's priorities during an iteration.

□ Team members don't have to work on tasks that they deem to not add value.

□ Management sets goals but doesn't tell team members how to achieve them.

□ Team members choose which tasks to work on.

(Part 2)

○ Including the safety expert’s opinion, management rarely changes the team’s priorities during an iteration.

○ Team members don’t have to work on tasks that they deem to not add value, including safety values.

○ Management and safety expert set goals but don’t tell team members how to achieve them.

○ Team members choose which tasks (including safety related tasks) to work on.

(3) Communication

(Part 1)

□ Standup meetings are effective at synchronizing work.

□ The team is not concerned about knowledge gaps when someone goes on vacation (or is otherwise unavailable).

□ Team members from one team communicate with team members from other teams in a high-bandwidth manner without undue interference.

□ Formal written documents are used to supplement rather than replace faster, more informal communication.

□ Team members communicate in a high-bandwidth manner without undue interference.

(Part 2)

○ Standup meeting are effective at synchronizing work (safety work is including).

- The team is not concerned about the knowledge gaps, and can take the safety consideration into it when someone goes on vacation (or it otherwise unavailable).
- Team members including safety expert from one team communicate with team members from other teams in a high-bandwidth manner without undue interference.
- Safety documents are used to supplement rather than replace faster, more informal safety information communication.
- Team members including safety expert communicate in a high-bandwidth manner without undue interference.

2. Requirements

(1) Emergency

(Part 1)

□ Requirements are represented at different levels of detail based on how soon the team expects to implement them.
□ The product owner is available to discuss upcoming features and work-in-progress.
□ Non-functional requirements are determined early enough to appropriately influence design and testing.
□ The whole team embraces change and emergent opportunities in an efficient, low-ceremony way.
□ Teams are able to start projects with incomplete requirements.

(Part 2)

- **Safety requirements** are represented at different levels of detail based on how soon the team expects to implement them.
- The **safety expert** is available to discuss upcoming features and work-in-progress.
- **Safety requirements** are determined early enough to appropriately influence design and testing.
- The whole team embraces **safety** change and emergent opportunities in an efficient low-ceremony way.
- Teams are able to start projects with incomplete **safety requirements**.

(2) Technical Design

(Part 1)

□ Projects do not begin with an extensive technical design phase.
□ The team performs iterative technical design throughout a project.

(Part 2)

- Projects do not begin with an extensive **safety** technical design phase.
- The team performs iterative **safe** design throughout a project.
- The team performs safe design throughout the iteration.

3. Planning and Execution

(1) Planning Levels

(Part 1)

□ Technical team members and the product owner are included in the planning process in a way that they can meaningfully and appropriately affect scope and deadlines.
□ The product owner maintains a prioritized product backlog.
□ Technical team members and product owners collaborate in determining what features will be included in the release plan.
□ At the start of each iteration, the team performs sufficient just-in-time planning to be confident of what it can complete in the iteration.

(Part 2)

- Technical team members, the product owner, and the **safety expert** are included in the planning process in a way that they can meaningfully and appropriately affect scope and deadlines.
- The product owner and the **safety expert** will both maintain a prioritized product backlog.
- Technical team members, product owners and **safety expert** collaborate in determining what **safety features** will be included in the release plan.
- The safety expert performs a draft safety plan before each sprint, and the safety plan is detailed during each iteration in parallel with the development team.

(2) Critical Variables

(Part 1)

□ All work is done in iterations of no more than 30 days.
□ Iterations focus on creating features with value to customers and infrequently focus on infrastructure specific work.
□ One or more of scope, schedule, or resources is allowed to change during a project.

(Part 2)

- All work (including safety work) in done in iterations of no more than 30 days.
- Iterations focus on creating features (including safety features) with value to customers and infrequently focus on infrastructure specific work.
- One or more of scope (including safety), schedule (including safety), or resources (including safety) is allowed to change during a project.

(3) Progress Tracking

(Part 1)

□ Teams know their velocity.

□ There is a highly visible representation of the team's progress within a release.

□ Each day (week), there is a highly visible representation of the team's progress within an iteration.

□ Each feature has a well-defined completion criteria that can be used to determine if the feature is done or not done. We do not consider a partially completed feature done.

(Part 2)

- There is a highly visible representation of the team’s progress (including safety progress) with a release.
- Each day (week), there is a highly visible representation of the team’s progress (including safety progress) within an iteration.
- Each safety feature has a well-defined completion criteria that can be used to determine if the **safety feature** is done or not done. We do not consider a partially completed feature done.

(4) Sources of dates and estimates

(Part 1)

□ Estimates are created collaboratively by the people who will do the work.

(Part 2)

- The safety estimations are created collaboratively by the people who will do the work.

(5) When do we plan

(Part 1)

□ Upfront planning is helpful without being excessive.

(Part 2)

- Upfront planning for safety is helpful without being excessive.

4. Technical Practice

(1) Unit Test-Driven Development

(Part 1)

□ Most code is written using unit test-driven development.

(Part 2)

o Most safety-related code is written using unit test-driven development.

(2) Pair Programming

(Part 1)

□ Code is written using pair-programming.

□ Team members pair program at appropriate times.

(Part 2)

o Safety-related code is written using pair-programming.

(3) Refactoring

(Part 1)

□ Refactoring is performed whenever needed.

□ Technical debt (i.e., accumulated undone or poorly done work) is made visible to both technical team members and stakeholders.

(Part 2)

o Refactoring is performed when safety requirements needed.

o Safety related technical debt is also made visible to both technical team members and stakeholders.

(4) Continuous Integration

(Part 1)

□ The entire system is built automatically at least once per day.

□ Automated unit and acceptance tests are run as part of each automated build.

(5) Collective Code Ownership

(Part 1)

□ Within our team, anyone can change anyone else's code.

□ The team can change any code in the system, even code written by other teams.

(Part 2)

o Within our team, anyone can change anyone else’s code (safety-related code is including).

5. Quality

(1) Customer Acceptance Test

(Part 1)

□ Product owners actively participate in the creation of the acceptance criteria for each feature.

(Part 2)

o Safety expert actively participate in the creation of the acceptance criteria for each safety feature, and decide together with the product owner.

(2) Timing

(Part 1)

- □ All bugs are fixed during the iteration in which they are found.
- □ At the end of each iteration there is little or no manual testing required.
- □ The team performs a variety of types of testing including functional, performance, integration, and scalability each iteration.
- □ Testers are involved and productive right from the start of each iteration.

(Part 2)

- All safety requirements related failures are fixed during the iteration in which they are found.
- At the end of each iteration there is little time for safety testing.
- Safety expert (also for safety analysis and test) are involved and productive right from the start of each iteration.

(3) Quality Focus

(Part 1)

- □ At the end of each iteration, the team has high-quality working software that it is comfortable being tested by people outside of the team.
- □ The team has pre-defined and agreed-upon criteria for considering a feature done.

(Part 2)

- At the end of each iteration, the team has a high-safety working software.
- The team has pre-defined and agreed-upon criteria for considering a **safety** feature done.

6. Culture

(1) Response to stress

(Part 1)

- □ When faced with a situation where scope cannot be met with the allotted resources in the allotted time, the team's initial reaction is to prioritize and explore tradeoffs.
- □ The team considers the economics of its choices when we make decisions.
- □ The team maintains a steady rate of productivity without being overworked.

(2) Title and salary alignment

- □ Bonuses, annual reviews, and compensation promote team behavior.
- □ Titles are not significant in how team members interact with one another.

7. Knowledge Creating

(1) Reflection

(Part 1)

- □ The team acts on retrospective feedback in a timely manner.
- □ The team holds retrospective meetings at the end of each iteration in which the team evaluates how they are doing and discuss how to get better.
- □ Iteration reviews are attended by product owners, stakeholders, and team members who provide actionable feedback.

(Part 2)

- The team evaluates also how to they are doing about safety assurance and discuss how to get better.
- Safety expert attends also the iteration reviews to provide safety-related actionable feedback.

8. Outcomes

(1) Measure

(Part 1)

□ The team has been more productive since we started using an agile approach.

□ Our customer(s) have been more satisfied with the functionality of our products since we started using an agile approach.

□ Our business has recognized greater economic value since we started using an agile approach.

□ Our customer(s) have been more satisfied with the usability of our products since we started using an agile approach.

□ The team has had higher morale since we started using an agile approach.

□ We have delivered functionality to users more quickly and/or more often since we started using an agile approach.

□ The team has produced higher quality products since we started using an agile approach.

(Part 2)

o The productivity is not influenced by increasing safety assurance in an agile approach.

o Our customer(s) have been more satisfied with both the functionality and the safety of our products since we started using an agile approach.

o The safety does not influence the great economic value brought by using an agile approach.

o The usability is also not influenced by safety of our products since we started using an agile approach with safety assurance.

o The morale is not influenced by increasing the safety assurance since we started using an agile approach.

o The high quality (except safety quality) of products is not influenced by increasing safety assurance since we started using an agile approach.