**As many of these requirements are for analyzing software to prevent ambiguity the**

**term TOOL is used for the web-tool that you are programming.**

**In this document items that are conducted by the:**

* **Software requirements but not displayed to the user will be shown in BLUE font.**
* **To be displayed as USER instructions are shown in PURPLE**
* **Black font is used for general discussion and explanations to support requirements**
* **TOOL INPUT = the tool should autofill that field from the source named or user selection (such as from check boxes). Retain the ability for the user to edit the field if needed. These are displayed in RED.**
* **USER = the user/analyst is to fill in that field. These are displayed in GREEN. USER ENTERS means that the Tool should present a fillable space for that item.**
* **If USER field says OPTIONAL then IGNORE.**
* **If USER field says REQUIRED then do not let user continue until space is field.**

**STEP 1**

* Display the following to the users
* Are you investigating an incident or accident or something that has already occurred?
  1. Direct to CAST analysis *Step 2 CAST*
* Are you designing something new (physical system, training, procedure, organization or other)?
  1. A system is a set of interdependent parts sharing a common purpose. The performance of the whole is affected by each and every one of its parts.
  2. Direct to STPA analysis *STEP 2 STPA*

[The screen should look something like this:]

A white paper with black text

AI-generated content may be incorrect.

**STEP 2 [go to STEP 2 CAST or STPA as appropriate]**

**STEP 2 CAST [the following items displayed in entirety]:**

1. **Define the scope of your investigation.** This is normally limited to those individuals or entities who can implement your recommendations. This can include regulators who work for the government (but not elected politicians), manufacturers, operators, and personnel, and local factors that might affect their decisions or operations. You will not include anything outside of the scope you define here in your analysis. This is to prevent trying to solve problems that are beyond your ability to solve and to frame your thinking.

[USER **open text - required**].

1. List the basic sequence of events that led to the incident or accident as bullet items.

* Do not use the word “fail” unless it is describing a mechanical component that broke as it will bias your analysis.
* For people or software describe their actions in a neutral way.
* Correct: “the pilot did not extend the landing gear”
* Incorrect: “the pilot failed to extend the landing gear”

[USER to fill **open text - required] Display one field at a time with capacity for at least five lines. Show a + sign to the right at the end of the line/paragraph that user can click on to create another field.**

The following is an example of how these should look to the user, modified as appropriate of course:

A screenshot of a report

AI-generated content may be incorrect.

.List the losses that occurred in your accident. Choose from the following list, check all that apply

[USER **check box next to each item**]:

* + - Loss of life
    - Loss of property (aircraft damage, other physical damage or loss)
    - Loss of revenue
    - Loss of security (software or physical)
    - Loss of mission
    - Loss of regulatory compliance
    - Loss of brand or reputation
    - Loss of customer satisfaction
    - Other [USER filled item – fillable space if checked]

*Tool should display chosen items in sequential order, displayed vertically as shown below. Each item is listed with a preceding tag for traceability, e.g.:*

L-1: Loss of Life

L-2: Loss of Revenue

L-3: Loss of brand

[the following text is now displayed]

* **Identify system-level hazards**.
  1. These are hazards that affect the entire system. Note the following:
     + 1. What were the hazards that occurred that resulted in the losses you identified? A hazard is defined as “A *hazard* is a system state or set of conditions that, together with a particular set of worst-case environmental conditions, will lead to a loss.”
       2. A hazard **must** be something that we can control through the design of the system. It is NOT something like a meteor, a storm or a mountain.
       3. For example, we *cannot* control the existence of a mountain, but we *can* control how close we get to the mountain while flying!
       4. System hazards are not about components. For example, a brake failure is not the system hazard (but it can lead to one), the system hazard is “moving too fast when an obstacle is ahead.”
       5. Hazards are *system* states or conditions (not component-level causes or environmental conditions).
       6. Hazards *will* lead to one of the identified losses in some worst-case environment
       7. Hazards must describe states or conditions to be prevented
  2. You will now be led through the process to write out your hazards
     1. *[Lead the user to create a properly formatted hazard, and each hazard identified. The hazard is constrained by the chosen losses, in other words, each hazard MUST lead to at least one loss that has been identified. All of the losses must be accounted for by at least one hazard as an example]*:

Display the following, with the color coding:

Here is what a hazard looks like:

H-1: Aircraft violate minimum separation standards inflight [L-1, L-2..]

[system component] [environmental conditions and system state] [trace to associated losses]

System component – this should be the most basic thing you are trying to control in your system. For any aircraft investigation we are concerned with the aircraft itself. For a nuclear power plant it would be the nuclear power plant. In a medical procedure it is the patient.

Component failures by themselves are not hazards. Component failures can cause hazards but are not in themselves hazards. For example, a brake failure is not a hazard, but inability to stop or slow the vehicle when obstacles are present is a hazard.

*[user selects a loss first, then builds the hazard as follows*

*Web-tool provides a list for ]*

**Select the System component** (this can be expanded later).

* Aircraft
* Ship
* Spacecraft
* Satellite
* Powerplant
* Vehicle
* Patient
* Software system
* Other [USER defined item – required if checked]

**System State and Environmental Condition (these are examples, more are possible)**:

The following items can be used to add context to the rest

* [ENVIRONMENTAL CONDITION -optional] Inflight [ENVIRONMENTAL CONDITION -optional]
  + - For an aircraft many things that can happen during flight are a hazard that are not necessarily a hazard on the ground, such as inability to control the pitch attitude or roll the aircraft
* [ENVIRONMENTAL CONDITION -optional] On the ground [ENVIRONMENTAL CONDITION -optional]
  + - Some hazards are unique to being on the ground. Inability on the runway is not a hazard until the aircraft lands. The pilot might choose to land on a long enough surface that this is not a problem
* [ENVIRONMENTAL CONDITION -optional] Too high a speed [ENVIRONMENTAL CONDITION -optional]
  + This can be a hazard to any vehicle, inflight, on the ground or in space
* [ENVIRONMENTAL CONDITION -optional] Too low speed [ENVIRONMENTAL CONDITION -optional]
  + This can be a hazard to any vehicle, inflight, on the ground or in space
* [ENVIRONMENTAL CONDITION -optional] Too close to [ENVIRONMENTAL CONDITION -optional]
  + This can potentially be a hazard in the air or the ground
* [ENVIRONMENTAL CONDITION -optional] Releases dangerous materials [ENVIRONMENTAL CONDITION -optional]
  + A hazard if these materials can move to an area where they can hurt people or property
* [ENVIRONMENTAL CONDITION -optional] Security is breached[ENVIRONMENTAL CONDITION -optional]
  + A hazard if malicious people gain access.
* [ENVIRONMENTAL CONDITION -optional] Unable to complete mission[ENVIRONMENTAL CONDITION -optional]
  + This can apply to many areas so should be further refined later
* [ENVIRONMENTAL CONDITION -optional] Exit doors not available [ENVIRONMENTAL CONDITION -optional]
* [ENVIRONMENTAL CONDITION -optional] Regulations not followed when[ENVIRONMENTAL CONDITION -optional]
* [USER filled option]

*[combine system component and system state and environmental conditions into one line also connecting it to at least one loss, automatically include the initial loss selected by user but prompt user to consider if the hazard might lead to other losses, create hazards and format as follows]*

Hazard specification [TOOL create in numbered order, H-1, 2, 3, etc.]

System component [TOOL IMPORT based on USER selection]

System state and environmental condition [TOOL IMPORT based on USER selection and entry]

Trace to associated losses [based on what USER selects from losses]

Following are examples. All fields are filled by TOOL based on user selections/inputs

H1: Aircraft moving too close to a severe storm [L-1, L-2]

H-1:= [Hazard Specification]

Aircraft = [system component]

moving too close to a severe storm = [environmental conditions and system state]

[L-1, L-2] = [trace to associated losses]

H-2: Aircraft violates minimum separation standards inflight [L-1, L,2]

H-1 =[Hazard Specification]

Aircraft = [system component]

Moving too close to a severe storm [Environmental conditions and system state]

[L-1, 2] [trace to associated losses]

After each hazard is completed, display a field that states [create an additional hazard?] that user can select to add another.

**STEP 2 STPA Analysis**

[Following text is displayed]

* Define the scope (boundaries) of your analysis.

Define the scope of your investigation based on the losses and hazards you have identified. This is normally limited to those individuals or entities who can implement your recommendations. This can include regulators who work for the government (but not elected politicians), manufacturers, operators, and personnel, and local factors that might affect their decisions or operations. You will not include anything outside of the scope you define here in your analysis. This is to prevent trying to solve problems that are beyond your ability to solve and to frame your thinking. [open text].

1. List the losses that you are trying to prevent. Choose from the following list, check all that apply [**check box next to each item**]:
   * + Loss of life
     + Loss of property (aircraft damage, other physical damage or loss)
     + Loss of revenue
     + Loss of security (software or physical)
     + Loss of mission
     + Loss of regulatory compliance
     + Loss of brand or reputation
     + Loss of customer satisfaction
     + Other [USER filled item – fillable space if checked]

*display chosen items in sequential order, displayed vertically as shown below. Each item is listed with a preceding tag for traceability, e.g.:*

L-1: Loss of Life

L-2: Loss of Revenue

L-3: Loss of brand

* NOTE that the following section for identifying hazards is identical to the CAST section, differences only occur when establishing system requirements after the hazard identification:
* **Identify system-level hazards**.
  1. These are hazards that affect the entire system. Note the following:
     + 1. What were the hazards that occurred that resulted in the losses you identified? A hazard is defined as “A *hazard* is a system state or set of conditions that, together with a particular set of worst-case environmental conditions, will lead to a loss.”
       2. A hazard **must** be something that we can control through the design of the system. It is NOT something like a meteor, a storm or a mountain.
       3. For example, we *cannot* control the existence of a mountain, but we *can* control how close we get to the mountain while flying!
       4. System hazards are not about components. For example, a brake failure is not the system hazard (but it can lead to one), the system hazard is “moving too fast when an obstacle is ahead.”
       5. Hazards are *system* states or conditions (not component-level causes or environmental conditions).
       6. Hazards *will* lead to one of the identified losses in some worst-case environment
       7. Hazards must describe states or conditions to be prevented
  2. You will now be led through the process to write out your hazards
     1. *[Lead the user to create a properly formatted hazard, and each hazard identified. The hazard is constrained by the chosen losses, in other words, each hazard MUST lead to at least one loss that has been identified. All of the losses must be accounted for by at least one hazard as an example]*:

Display the following, with the color coding:

Here is what a hazard looks like:

H-1: Aircraft violate minimum separation standards inflight [L-1, L-2..]

[system component] [environmental conditions and system state] [trace to associated losses]

System component – this should be the most basic thing you are trying to control in your system. For any aircraft investigation we are concerned with the aircraft itself. For a nuclear power plant it would be the nuclear power plant. In a medical procedure it is the patient.

Component failures by themselves are not hazards. Component failures can cause hazards but are not in themselves hazards. For example, a brake failure is not a hazard, but inability to stop or slow the vehicle when obstacles are present is a hazard.

*[user selects a loss first, then builds the hazard as follows*

*Web-tool provides a list for ]*

**Select the System component** (this can be expanded later).

* Aircraft
* Ship
* Spacecraft
* Satellite
* Powerplant
* Vehicle
* Patient
* Software system
* Other [USER defined item – required if checked]

**System State and Environmental Condition (these are examples, more are possible)**:

The following items can be used to add context to the rest

* [ENVIRONMENTAL CONDITION -optional] Inflight [ENVIRONMENTAL CONDITION -optional]
  + - For an aircraft many things that can happen during flight are a hazard that are not necessarily a hazard on the ground, such as inability to control the pitch attitude or roll the aircraft
* [ENVIRONMENTAL CONDITION -optional] On the ground [ENVIRONMENTAL CONDITION -optional]
  + - Some hazards are unique to being on the ground. Inability on the runway is not a hazard until the aircraft lands. The pilot might choose to land on a long enough surface that this is not a problem
* [ENVIRONMENTAL CONDITION -optional] Too high a speed [ENVIRONMENTAL CONDITION -optional]
  + This can be a hazard to any vehicle, inflight, on the ground or in space
* [ENVIRONMENTAL CONDITION -optional] Too low speed [ENVIRONMENTAL CONDITION -optional]
  + This can be a hazard to any vehicle, inflight, on the ground or in space
* [ENVIRONMENTAL CONDITION -optional] Too close to [ENVIRONMENTAL CONDITION -optional]
  + This can potentially be a hazard in the air or the ground
* [ENVIRONMENTAL CONDITION -optional] Releases dangerous materials [ENVIRONMENTAL CONDITION -optional]
  + A hazard if these materials can move to an area where they can hurt people or property
* [ENVIRONMENTAL CONDITION -optional] Security is breached[ENVIRONMENTAL CONDITION -optional]
  + A hazard if malicious people gain access.
* [ENVIRONMENTAL CONDITION -optional] Unable to complete mission[ENVIRONMENTAL CONDITION -optional]
  + This can apply to many areas so should be further refined later
* [ENVIRONMENTAL CONDITION -optional] Exit doors not available [ENVIRONMENTAL CONDITION -optional]
* [ENVIRONMENTAL CONDITION -optional] Regulations not followed when[ENVIRONMENTAL CONDITION -optional]
* [USER filled option]

*[combine system component and system state and environmental conditions into one line also connecting it to at least one loss, automatically include the initial loss selected by user but prompt user to consider if the hazard might lead to other losses, create hazards and format as follows]*

Hazard specification [TOOL create in numbered order, H-1, 2, 3, etc.]

System component [TOOL IMPORT based on USER selection]

System state and environmental condition [TOOL IMPORT based on USER selection and entry]

Trace to associated losses [based on what USER selects from losses]

Following are examples. All fields are filled by TOOL based on user selections/inputs

H1: Aircraft moving too close to a severe storm [L-1, L-2]

H-1:= [Hazard Specification]

Aircraft = [system component]

moving too close to a severe storm = [environmental conditions and system state]

[L-1, L-2] = [trace to associated losses]

H-2: Aircraft violates minimum separation standards inflight [L-1, L,2]

H-1 =[Hazard Specification]

Aircraft = [system component]

Moving too close to a severe storm [Environmental conditions and system state]

[L-1, 2] [trace to associated losses]

After each hazard is completed, display a field that states [create an additional hazard?] that user can select to add another.

* **Define system-level constraints (SC)**

High level system-level constraints are essentially inverses of each hazard. If

H-1 Aircraft moving too close to a severe storm [L-1, L-2]

SC-1 Aircraft must not move too close to severe storm [H-1]

The following will guide you to create the SCs:

Create the following for each hazard identified in order, H-1, 2, 3, etc.

Constraint identification [Tool automatically numbers these, led by SC]

System component [TOOL IMPORT from hazard]

User chooses:

* Shall not
* Must not

[Insert Environmental Conditions and system state]

Trace to associated hazard [insert from hazard]

**Define sub-hazards if appropriate – create sub-hazard using same approach**

H-2: Aircraft moving too close to the ground while not over runway surface [L-1, L2]

To prevent this hazard thrust and pitch attitude (angle of attack) must be controlled

H-1.1: Thrust is not sufficient to prevent descent during flight [H-2]

H-1.2: Ange of attack not sufficient to prevent descent during flight [H-2]

Hazard is shown. User to define this with automatic tracing to hazard. Example

* Do you want to create a sub-hazard?

[Insert sub-hazard number in sequence] [user defined text] [insert hazard number]

* Do you want to create another sub-hazard?

If checked, repeat process for next item

Additional constraints can be created for each sub-hazard in the same manner.

**STEP 3**

The goal of this analysis is to control or constrain the behavior of the system to prevent an accident or any unwanted behavior.

To model this you will be creating a hierarchical control structure. This is not a schematic or organizational chart. Instead, the items on this structure are either something that is controlling something else (a controller), which can be software, human or organization, or a controlled process/item (Figure 3-1).

This tool will guide you through this process.

Start at the most basic component in the system you intend to control. In most systems this is the physical system, such as a vehicle, an aircraft, a satellite or similar. If you are analyzing just a component that that is the most basic, which could be something like the brakes, a steering wheel, the engine, or perhaps even a door. It can also be a process if you are analyzing an organization on how it manages a department or even a person, such as the case in health care where the patient might be the most basic process. Regardless, it is the lowest level you are analyzing.

Think about your system. What is the most basic component in your system?

**[this should be imported from Step 2 but user can modify]**

**Do you want to analyze the controls within your component, as an example with an aircraft the flight controls. This might be the elevator, ailerons and rudders, or just one of these. If this is a new design then it might be important to not specify how the control is facilitated, in which case use pitch control, roll control, yaw control in place of the specific flight control.**

* **Yes [USER enter the control] if yes, then tool should automatically add another field to include more than one.**

*Graphically, Tool creates a box on the bottom of the page that is labeled with the name of that component*.

Is there more than one basic component? An example might be a fleet of unmanned drones.

* **No*.*** *Move to next item*.
* **Yes.** Please name the other components. If they are identical (such as drones) then provide the number of identical components. If they are different, then name each component with a unique name. [USER enters]

*Create parallel boxes at the bottom of the page named accordingly on the bottom of the page. With identical components number them as appropriate, e.g. UAS 1, 2, 3…n*

What is controlling the behavior (this could be direct control, guiding, constraining, providing rules and procedures or similar**)? [USER Choose a name for it.]**

**What type of controller is it? Choose one of the following:**

* S- Software/computer
* H - Individual human
* T - A small team of people (such as on a large aircraft, a ship, a powerplant control room, etc.)
* O - An organization (this would be company, government agency, association, union, or similar)

*Place that controller, named as above, with its chosen categorization (S, H, T, O) labeled in a box centered over the controlled component/process or group of components and label it according color coded for ease of visual separation.*

*Box to be displayed with black font but filled with the respective color of the type of controller*

*Draw an arrow down from the controller to each controlled component, one for each, coming down from the left side of the controller, and attaching to the left side of the box of the controlled component. On the arrow place a small box labeled “actuator”*

What controls does the controller have over the component?

*Adjacent to each down (control) arrow label the arrow with what specific controls it represents.*

Is there any sort of feedback from the component to the controller?

* **Yes**
* **No**

*If NO, then place a RED dashed line from the component to the controller to the right of the down (control) arrow.*

*If YES, place an upward arrow from the controlled process to the controller with a box on it that states “sensor”*

*Note: the sensor and actuator boxes are not shown in the below images)*

Is there more than one controller sharing the control for the component?

*If NO, go to next step.*

*If YES then:*

Does the other controller have more authority than the first controller?

* **YES**
* **NO**

*If NO, then place the controller adjacent to the first controller at the same level on the page and add a downward control arrow.*

If YES, then place new controller at a higher level on the page as the FLIGHT CREW in Figure 3-2.

**If it is a different but parallel controller, label and place adjacent as in Figure 3-3.**

If two controllers are adjacent and communicating, portray as in top diagram in Figure 3-2.

**If this is a Team, then continue to Team’s section**.

Either way, place a DOWN arrow from the controller to the controlled component as previously described.

Is there feedback from the component/controlled process to the controller? **YES/NO**

*If YES, place an upward feedback arrow from the component to the controller.*

*If NO, is there indirect feedback, such as via another controller? If YES, add an upward feedback arrow.*

Figure 3-

*A diagram of a control system

AI-generated content may be incorrect.*

Figure 3-

A diagram of a control system

AI-generated content may be incorrect.

Figure 3-

A diagram of a plane

AI-generated content may be incorrect.

If the controller is S or H, skip to STEP 3.xx. If controller is O, skip to Step 3.xx

**TEAMS**

If controller is T:

If Team are all equal members with no ranking, ask the USER how many members there are. Display these as independent parallel boxes over the process.

If the analysis would like to only consider the team as a single group (such as “pilots”) use the HUMAN framework for analysis (later portions will enable the user to revisit this, so it will display with the option on the control structure)

List the command gradient

Do you want to consider the team as a single unit?

Display as a single box

Are all team members of equal rank?

Yes – How many members of the team are there?

Select from the following and add labels as appropriate:

* For command rankings, as appropriate to the analysis, the following authority (limited to 5 for this tool) can be defined
* CM-1 Command 1, or highest authority in the team
* CM-2 Command 2
* CM-3 Command 3
* CM-4 Command 4
* CM-5 Command 5
* GR – Grouped team. Below this level group members into one controller, such as all the cabin crew, or perhaps a platoon. This group can be independently analyzed in the same way as the higher level team if pertinent to the analysis.
* These can be labeled by the analyst if desired. If not labeled they default to the above listing, but one example is listing them as
  + Captain
  + First Officer
  + Cabin Crew
* If the analyst selects different names, use these names in the control structure for the team analysis as depicted in Figure 4.

Note: The above rankings can apply to any team, e.g. Captain is CM-1, First Officer is CM-2, Relief Pilot CM-3, Cabin Crew (GR). This is applicable also to military ranking, or such managerial such as Senior Manager, Manager, Employee group at the bottom, and within the GR than this can be divided, such as for cabin crew, Purser, or lead, with other ranking below.

In a team people will assume different roles and duties, which are assigned by a combination of rules, policies and procedures and training, but also as designated by the command gradient, e.g., the captain will designate which pilot is the Pilot Flying (PF), sometimes called the “handling pilot” and who is the pilot monitoring (PM). If a pilot is flying the aircraft the monitoring pilot (in addition to monitoring the PF) is also tasked with other duties that the PF cannot accomplish due to the need for them to be directly controlling the aircraft flight path.

List the roles within the team. An example for aircraft would be:

* Pilot Flying (PF)
* Pilot Monitoring (PM) – responsible to monitor the PF and, if necessary, constrain the behavior or control actions of the PF as needed to prevent UCAs.
* Relief Pilot (RP) – this pilot will also observe and monitor when on the flight deck and prevent UCAs by the PF and PM
* Cabin Crew (CC) if appropriate to your analysis, e.g. you are also analyzing events involving the cabin or the cabin crew providing other information, such as smelling smoke or hearing a critical sound.

For other teams, list them as needed, so user can select as many as are appropriate for the analysis, so the tool selection is as follows, with the open items left to the analyst as each system is different. The analysist should consider each position on the team where there is a different role. Include only those positions that are actively involved in the operation. Exclude those that are on a scheduled break or off-duty.

* PF
* PM
* RP
* CC
* [user fills in blank role]
* [user fills in blank role]
* [user fills in blank role]

Depict the team roles are shown in Figures 3-4 – 6, depicting the possible combinations of roles. The facet being captured here is that there can be dual and conflicting roles. The CM-1/Captain is a control over all other members of the crew, while at the same time, the PM (who may be CM-2/First Officer) is acting as a control to the Captain when the Captain is PF. In these diagrams Captain and First Officer have been shorted to C and F, respectively. This is not necessarily how the tool needs to display the information, but if desired then there should be a legend available. Listed on the control side are the aircraft controls for the perspective controller. Here the division is between PF and PM. Notice that the controller operating the controls is listed below the other controller, and the PM is working different controls than the PF.

For CAST, Analysis will what the actual user roles were if known. If NOT known, continue as with STPA. If known, such as who was PF and PM with Captain and First Officer, then ONLY show the pertinent combinations. Similarly, only display pertinent automation modes as appropriate (below examples, who was PF, was autopilot on, were autothrottles on, etc.)

**For STPA, analysist should investigate all the combinations.**

Figure 3-

A diagram of a plane

AI-generated content may be incorrect.

Note: Autopilot and Autothrottle OFF condition Add also actuators and sensors.

Figure 3-

A diagram of a plane

AI-generated content may be incorrect.

NOTE: Autopilot OFF, Autothrottle ON condition

Figure 3-

A diagram of a plane

AI-generated content may be incorrect.

NOTE: Autopilot and Autothrottle ON condition, PF controlling the aircraft via the Flight Guidance System (FGS)

The extended diagrams for teams (Figures 3-6) should be a selectable pop-out, but are generally hidden to reduce screen clutter.

**Organizational Analysis**

This is built in a similar manner as previous control scenarios. There can be multiple departments, some are working over a shared process (such as in an airline the aircraft and flight are controlled under the flight department, dispatch, maintenance and others). The modeling is done in the same way.

These should be attached to the larger control structure (unless only an organizational analysis is being conducted), so will be listed over the other controllers (usually). See Figure 10 for the entire control structure, where the operations and direct control are only in the right-hand corner. All the rest represent some type of organization.

* **More than one controller controlling the same process**

**If checked, than show as in Figure 7.**

* **Controllers are sharing information between them (departments sharing),**

**then depict as in Figure 3-9.**

Figure 3-

A diagram of a process

AI-generated content may be incorrect.

* One controller controlling multiple separate items

Figure 3-

A diagram of a senior management

AI-generated content may be incorrect.

Figure 3-

A diagram with arrows and text

AI-generated content may be incorrect.

Figure 3-

A diagram of a company

AI-generated content may be incorrect.

**STEP 4**

List controls from control structure

Tool prompts analyst to name what controls each controller has available to it, **starting at bottom of the control structure, moving left to right across that level, then moving up to the next level**, within the boundaries (scope) of the analysis.

User is prompted to eliminate those control actions that are completely out of scope of the analysis. An example would be “apply brakes” for the analysis of something that is only occurring in flight.

Name and type of controller autofilled from tool, starting from bottom of control structure and moving laterally across each level and then upwards

**[TOOL INPUT NAME OF CONTROLLER] [TOOL INPUT TYPE – S, H, T, O]**

In this step the *SOURCE* and the *CONTROL ACTION* are defined.

The SOURCE of the control is the CONTROLLER [Tool autofills from control structure].

The control action is what the controller is doing.

For example in the case of a team of pilots with the autopilot and autothrottles off:

Pilot Flying Controls:

Pitch (angle of attack) which might be further defined by the analyst as elevator and trim, roll, yaw, thrust.

SOURCE = PF

CONTROL ACTION. Note that each can be further refined, as in the elevator example below, at the discretion of the user. So for each control action, ask “do you want to further refine this control action?”

* Increase pitch – refine further? Display a + and if selected user can add a line.
  + Move elevators UP
  + Trim NOSE UP
* Decrease pitch +
  + Move elevators DOWN
  + Trim NOSE DOWN
* Increase thrust +
  + On all engines?
* Decrease thrust +
  + On all engines?
* Roll LEFT +
* Roll RIGHT +
* Yaw LEFT +
* Yaw RIGHT +

This process is then done with each other control action that is possible.

What controls does this controller have? [TOOL automatically presented the following for either CAST or STPA depending on what analysis the USER chose].

* CAST analysis: Exclude controls that could not possibly have been involved in the accident (such as landing gear brakes for an inflight event, or roll control on a brake failure event)
* STPA analysis: Exclude controls that fall outside the analysis (such as the brakes if evaluating inflight controls).

For the roles, as displayed in Figures 3-6, each control uniquely available to the controller is provided, however, in most cases all of the controls are available regardless of position, so, for example, PF and PM can access all controls even if they are not ordinarily doing so. The separation of their duties is addressed in Steps 5 and 6.

Pilot monitoring controls:

* Flaps,
* landing gear,
* flight guidance system

For the airline over the pilots they might have:

* Training,
* policies and procedures

*At this stage it is already possible to identify facets that are needed, or should be in place. Any control path should be associated with a way to control the process that is being controlled and feedback on that process. The feedback is some way to measure that the control is working. For a human that feedback can be in the form of any of the five senses, depending on the process being controlled. Software would need sensors (as would humans for many things). Organizations might use checks and audits.*

**STEP 5 – Determination of UCAs**

Start at bottom of control structure, this process will do the following, but most will be behind the scenes and the user will be guided without necessarily being aware of the following:

1. Identify hardware and electro-mechanical failures and unsafe interactions
2. Identify UCAs in control structure
3. Identify UCCA for teaming control – macro scale
4. Identify UCA/UCCAs based on role divisions
5. Identify UCA/UCCAs for organizations

**Hardware and Electro-Mechanical Component Analysis**

Analyst should consider the hardware and electro-mechanical failures. This will be further developed as part of causal scenarios but some preliminary evaluation can occur here.

* Probability assessments for hardware and electro-mechanical components (import if known).
* Consider combinations of failures or interactions that can create a hazard:
  + One engine fails catastrophically and that then sends shrapnel that damages other components.
  + A design that results in one component mechanically blocking another component
  + Factors that can affect all components (fuel contamination).

**1. System & Component Overview**  
☐ Review key documentation (schematics, block diagrams, spec sheets)  
☐ List all major components (motors, gearboxes, sensors, actuators, controllers, wiring)

**2. Failure Mode Brainstorm**  
☐ For each component, note common failure types:  
  • Mechanical wear (bearings, belts, gears)  
  • Electrical faults (shorts, opens, insulation breakdown)  
  • Sensor drift or dropout  
  • Control electronics/software glitches

**3. Unsafe Interaction Scan**  
☐ Identify critical interfaces where two or more elements meet (e.g. motor-coupling, sensor-actuator loop)  
☐ Consider how one component’s failure could drive another into a hazardous state (e.g. stalled motor → gearbox overload → housing rupture)

**User should document the findings for each hardware or electro-mechanical component.**

*Control actions from previous step are provided to the user starting at the bottom of the control structure, then branching out as necessary for items on the same level – moving from left to right across that layer, but does* ***NOT*** *consider the ROLES or SHARING of controls. The user does not need to see the control structure here, but the system must ensure that each control previously identified for each controller is presented.*

*User can add additional control actions at this step if they realize they missed one.*

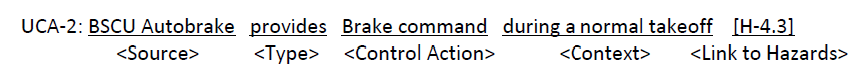
**At each stage the user should be reminded to eliminate items that are outside the scope of the investigation. For example, if applying brakes was listed as a possible control action but the investigation only involves what happens in flight, then brakes should be excluded. The following is incorporated as a reminder each time a UCA is created. The user has the option to exclude it, or add it back later if necessary.**

Tool presents each control action previously identified in step 4 in turn, moving across all of the lowest level of controls.

USER is asked to consider if there is any way that the control action can be unsafe considering each of the following ways that a control action can be unsafe, is there any context where the control action can be unsafe:

* 1. Not providing the control action leads to a hazard.
  2. Providing the control action leads to a hazard (this includes not providing enough or providing too much, for example, a driver might provide too much gas or not enough gas or a pilot might add too much elevator or not enough.
  3. Providing a potentially safe control action but too early, too late, or in the wrong order (think of a driver turning left, right, left instead of right, left, right, as an example of the wrong order).
  4. The control action lasts too long or is stopped too soon (for continuous control

Each UCA consists of five parts (source, type, control action, context and link to hazard – tool to constrain the user such that they complete the UCA correctly. Tagging is automated, UCAs numbered in the order they are analyzed. Following is how a completed UCA should look, without the labels on the second line:



The UCAs are numbered in order, UCA-1, 2, etc.

Expanding on the second line, these are the parts that comprise a UCA

<source controller> <control action> <type, e.g. provides, not provides, too soon/late, too long/short> <context – circumstances that would make this lead to one of the hazards – but not the “why”> <link to at least one of the hazards>.

The parts of the UCA are completed as follows:

UCA is automatically numbered in the order that it was created, UCA-1, 2, 3, etc.

* The system should automatically NUMBER the UCAs as in UCA-1, UCA-2, etc. and automatically connect each one to the Hazard that the user identified. The system does NOT allow the user to complete the UCA unless each part is complete.

***Source*** and ***control action*** were already defined in Step 4 so these are automatically filled in order. So at this point the following is displayed to the user with items to be completed shown in red. (as an example for the pilot-flying (PF) as the controller and “increase thrust” as one of the previously listed controls):

**UCA-1 PF <TYPE> increase thrust <CONTEXT> <LINK TO HAZARDS>**

The system prompts each TYPE of UCA in turn, asking if there is a way that each of these might lead to one of the **previously identified** hazards, asking

**“Would the following result in one of the previously identified hazards?”:**

**NOT PROVIDING** [**fill in control action identified in Step 4]** can be unsafe? YES/NO – if YES, [*under what conditions will that lead to one of the hazards*] *and which hazard will it lead to* (H-1, H-2, etc.)

**PROVIDING** [**fill in control action identified in Step 4**] can be unsafe (consider providing but not enough or too much)? YES/NO - if YES, [*under what conditions will that lead to one of the hazards*] *and which hazard will it lead to* (H-1, H-2, etc.)

**PROVIDING** [**fill in control action identified in Step 4**] **TOO EARLY** can be unsafe? YES/NO if YES, [*under what conditions will that lead to one of the hazards*] *and which hazard will it lead to* (H-1, H-2, etc.)

**PROVIDING** [**fill in control action identified in Step 4**] **TOO LATE** can be unsafe? YES/NO - if YES, [*under what conditions will that lead to one of the hazards*] *and which hazard will it lead to* (H-1, H-2, etc.)

**PROVIDING** [**fill in control action identified in Step 4**] **IN THE WRONG ORDER** can be unsafe? YES/NO - if YES, [*under what conditions will that lead to one of the hazards*] *and which hazard will it lead to* (H-1, H-2, etc.)

**PROVIDING** [**fill in control action identified in Step 4**] for **TOO LONG A PERIOD OF TIME**? YES/NO - if YES, [*under what conditions will that lead to one of the hazards*] *and which hazard will it lead to* (H-1, H-2, etc.)

**PROVIDING** [**fill in control action identified in Step 4**] **TOO SHORT A PERIOD OF TIME** can be unsafe? YES/NO - if YES, [*under what conditions will that lead to one of the hazards*] *and which hazard will it lead to* (H-1, H-2, etc.)

* The tool should provide a PULL DOWN menu of the previously listed hazards so the user can select as many as apply to each hazard. At least ONE must be selected. If NONE are possible, the USER is prompted to consider adding an additional hazard.

In this way each UCA is identified at the basic level. UCAs that have been completed are not displayed, however the user can always pull them up to review them.

**Simultaneous Control Actions – Single Controller**

Consider UCCAs with a single controller. An example is an aircraft pilot with control over pitch, roll, yaw, flaps/slats/spoilers/brakes and thrust. Some of these are more than one control and should be included if applicable. USER should be prompted for:

* Engines [moving thrust separately can be a UCA]
* Brakes [differential wheel braking can be a UCA]
* Other

Independently one of these may be safe, but when conducted in combination with other control actions it can be unsafe. As an example, increased pitch can be unsafe when thrust is not increased.

***Cx is the controller*.** *Control actions are listed here as u1, u2, u3…uN.but are replaced with the actual control actions developed in Step 4 in the UCCA. Tool should INPUT the name of the controller and then list each Control Action identified from Step 4.*

* *Cx* **does not provide {**𝑢1, 𝑢2, 𝑢3…n} **and does not provide** {*u*1, 𝑢2, 𝑢3…n} when… *[H1,2..]*
* *Cx* **does not provide {**𝑢1, 𝑢2, 𝑢3…n} and **provides {**𝑢1, 𝑢2, 𝑢3…n}when… *[H]*
* *Cx* **provides {**𝑢1, 𝑢2, 𝑢3…n} and **provides {**𝑢1, 𝑢2, 𝑢3…n}when… *[H]*
* *Cx* **starts providing {**𝑢1, 𝑢2, 𝑢3…n} before **starting providing {**𝑢1, 𝑢2, 𝑢3…n} when… *[H]*
* *Cx* **starts providing {**𝑢1, 𝑢2, 𝑢3…n} before **ending providing {**𝑢1, 𝑢2, 𝑢3…n} when… *[H]*
* *Cx* **ends providing {**𝑢1, 𝑢2, 𝑢3…n} before **ending providing {**𝑢1, 𝑢2, 𝑢3…n}when… *[H]*

**The next step analyzes the roles which have some similarities to simultaneous control actions.**

Human roles on flight deck with dynamic control sharing

* Considerations given for unsafe interactions.
* Both PF and PM accessing the same controls is considered (such as moving a particular control together, same or opposite direction)
* **Roles addresses when each are using the controls as part of their unique role, but those are working in a way that is unsafe**.
* UCAs established using standard method for each item
* UCCAs established for each control action when combined with the other controller, for example
  + UCCA-1: PM provides flaps too early when PF provides speed reduction too late so flap speed is exceeded [H-3]

*The same level includes the comparison between the same controller but in different roles, i.e., the FO in one role below the Captain, but also the PF in a role below the PM, etc. The roles and control actions are dynamic, and the control actions between them become hazardous only in certain combinations. As there are only two controllers at this level but multiple control actions, the analysis is fairly straightforward.*

* Complete analysis as above for Captain and First Officer and PF and PM. PM is both a control to PF and also takes direction from PF for certain functions. The iteration will result in several additional UCAs, including those of differing roles as well as between the dynamic hierarchies between individuals.
* UCCAs Analyze interactions between PF and PM to identify Unsafe Combinations of Control Actions. Each pilot

*Analyze interactions as separate controllers, each with access to all the controls. The two controllers can be considered the same for this purpose. The context is controller agnostic for this portion of the analysis.*

*Cx is one controller, Cy is the other listed here, one representing PM and the other PF. These are replaced with PM and PF in the written out UCCA that is presented. Control actions are listed here as u1, u2, u3…uN. but are replaced with the actual control actions developed in Step 4 in the UCCA. For the pilots this consists of pitch, roll, yaw, thrust, speed brakes, wheel brakes, flaps, slats, landing gear and flight guidance system (FGS) controls (speed, heading, altitude, engage FMS vertical mode- thrust, pitch, engage FMS lateral mode).*

*Tool automatically swaps whether Cx and Cy represents PF and PM respectively, or vice versa, so all conditions are mapped but conditions that the outcome is agnostic to which is which are only shown once as PF/PM instead of either one. The following are controller agnostic:*

* *Cx* **does not provide {**𝑢1, 𝑢2, 𝑢3…n} and *Cy* **does not provide** {*u*1, 𝑢2, 𝑢3…n} when… *[H]*
* *Cx* **does not provide {**𝑢1, 𝑢2, 𝑢3…n} and *Cy* **provides {**𝑢1, 𝑢2, 𝑢3…n}when… *[H]*
* *Cx* **provides {**𝑢1, 𝑢2, 𝑢3…n} and *Cy* **provides {**𝑢1, 𝑢2, 𝑢3…n}when… *[H]*
* *Cx* **starts providing {**𝑢1, 𝑢2, 𝑢3…n} before *Cy* **starts providing {**𝑢1, 𝑢2, 𝑢3…n} when… *[H]*
* *Cx* **starts providing {**𝑢1, 𝑢2, 𝑢3…n} before *Cy* **ends providing {**𝑢1, 𝑢2, 𝑢3…n} when… *[H]*
* *Cx* **ends providing {**𝑢1, 𝑢2, 𝑢3…n} before *Cy***ends providing {**𝑢1, 𝑢2, 𝑢3…n}when… *[H]*

When there are three or more controllers the analysis is conducted using the following abstraction level.

**In this case controllers are provided in groups for comparison, such as “pilots” or “control room operators”.**

As with the previous UCA’s the tool will automatically number the UCA and will fill in the name of the controller and control action.

* *Any* 𝑐x **does not provide** 𝑢1 and *any other* **does not provide** 𝑐x {𝑢2 or 𝑢3} *when multiple controllers interfere with each other[H]*
* Any 𝑐𝑁 **does not provide** 𝑢1 and *any other* **provides** {𝑢2 or 𝑢3} *when multiple controllers interfere with each other[H]*
* Any 𝑐𝑁 **provides** 𝑢1 and *any other* **does not provide** {𝑢2 or 𝑢3} *when multiple controllers interfere with each other[H]*
* Any 𝑐𝑁 **provides** 𝑢1 and *any other* **provides** {𝑢2 or 𝑢3} *when multiple controllers interfere with each other[H]*
* 𝑐𝑁 **starts providing** 𝑢1 before it **starts providing** {𝑢2 or 𝑢3} when… *[H]*
* 𝑐𝑁 **starts providing** 𝑢1 before it **ends providing** {𝑢2 or 𝑢3} when… *[H]*
* 𝑐𝑁 **ends providing** 𝑢1 before it **starts providing** {𝑢2 or 𝑢3} when… *[H]*
* 𝑐𝑁 **ends providing** 𝑢1 before it **ends providing** {𝑢2 or 𝑢3} when… *[H]*

A screenshot of a diagram

AI-generated content may be incorrect.

**Organizational Analysis – The same as UCCA but organization/department name is substituted at the controller.**

**Last step: User is prompted to filter any remaining UCAs/UCCAs that do not apply**

* CAST analysis: Eliminate UCAs that were not accident factors but not before considering the interaction between apparently unrelated UCAs.
* STPA analysis: Only consider UCAs within the scope of the analysis – but not before considering interaction between apparently unrelated UCAs

**STEP 6**

**Causal Scenarios**

In this section, display the class of the scenario for each part from the following image so the user can get a mental model of what they are looking for.

Now that the UCAs have been identified, why would the controller make that decision?

In this step each type of controller is evaluated for each of the classes of scenarios.

A diagram of a class

AI-generated content may be incorrect.

The controllers were previously designated for if they are software, human, small team or an organization. Based on that classification, the UCAs are analyzed in order (which is actually starting from the bottom of the control structure, as stated in the previous section).

**[>>Tool instructions for this section**

The following is to provide the analyst for considerations on why it “made sense” for the controller to make a UCA. These are called “causal scenarios.”

* The “Describe scenario” prompt appears if a box is checked, and a field will be presented where the analyst should describe how the particular item could result in the UCA under consideration.
* The UCA will display to the analyst until the analysis is complete for that UCA.
* Boxes are presented, the analyst should only check the box if the item, or one similar to it, applies. The analyst should NOT limit their analysis to these items but rather use them as a way to generate ideas on why the UCA occurred.
* If Analyst checks the box for a certain item then the tool shall automatically be directed to a field to enter how that item might explain the causal scenario.
* At all times an open field will be provided called “additional causal scenario” so if the analyst thinks of a causal scenario (unrelated to any items presented below) they can enter that scenario.

In the below fields:

TOOL IMPORT = tool should automatically fill the answer drawing from control structure or previous steps.

*Always maintain an Option for analyst to modify or enter data if necessary.*

USER = Analyst fills in data.

Tool stores user filled data of causal scenarios below. A causal scenario prompt will state: []

UCA Identification – the following items will be autofilled by the tool at each appropriate point.

<Automatically IMPORT controller and UCA from step 6, for each case below substitute the controller and UCA, in order, starting from the bottom of the control structure, working up each tier and then laterally across the control structure to capture each UCA that has been identified>

**Controller:** <TOOL IMPORT>

*The list of all UCAs associated with the particular controller will display on the RIGHT side of the page for that controller*

*A small diagram of the particular controller with its control and feedback paths as well as the process it is controlling and lines showing inputs to the controller with the items named will be shown on the right below the UCA list*

**UCAs:** < TOOL IMPORT and display a list of ALL UCAs for that controller from previous steps. >

All UCAs are shown at once so analyst can consider all of them together for each item and save some time. These are not displayed yet, but held.

>>Display the following with the following checkboxes. ]

**Feedback/Input (s) to Controller (from control structure): <** TOOL IMPORT **from control structure showing source>**

☐ **The controller has feedback: USER inputs the types of feedback that are available. If others are needed, a field is provided to add those as requirements [USER provides other feedback that should be present, requirement that the controller must be provided the {USER PROVIDED} feedback}]**

☐ **The controller does NOT have feedback: USER inputs the types of feedback that the controller should have. A requirement is generated that Feedback must be provided that contains the following [USER provides].**

**Analyst to describe the feedback to this controller (often more than one source of feedback is available. Provide only the feedback types pertinent to the analysis)**

**Green USER [Tool provides a box to check if there is another type of feedback for the controller]**

☐ **USER**

☐ **There other sources of information to the controller in addition to feedback**

**Analyst to describe other sources to this controller USER**

**Show controller from control structure with output control and feedback plus external factors, with the aspect under consideration highlighted in the figure.**

Tool automatically will know the type of controller from **Step 4 identification** tag:

S- Software/computer

H - Individual human

T - A small team of people (such as on a large aircraft, a ship, a powerplant control room, etc.)

O - An organization (this would be company, government agency, association, union, or similar)

Based on the type of controller the user will be directed to the appropriate analysis by the tool:

Software> go to 1.1

Human> go to 1.2

Team> go to 1.2.2 then to 1.3

Organization> go to 1.4

**Tool requirement>Based on the type of controller go directly to that section. Progress through all four classes of scenarios for each controller before moving to the next controller.**

***For each controller display all the UCAs associated with that controller continuously. This may need to be scrolled by the user.***

***For each scenario, number scenarios sequentially as follows:***

***SC-1, 2, 3…n***

***Each scenario ends with a trace to the UCA, so a full scenario might look like***

***SC-1: Captain commands first officer to extend flaps at too fast a speed resulting in flap mechanical failure [UCA-2]***

***Tool should discretely code each scenario with the following additional information:***

* ***Controller***
* ***Class of scenario***

***This identification will be used to sort items with the requirements***

**Following are the next portions of the analysis. USER should check boxes for items that are deemed applicable to their analysis:**

1. **Class 1 Scenarios**
   1. **[Section Class 1 CS – Software]**

**If controller is designated S = Software then direct HERE.**

* + 1. **Causal Factor Evaluation for Software**

*Check if applicable and describe*

* + 1. **Hardware & Platform Failures**  
       ☐ Hardware failure (power loss, overheat, sensor/processor fault)

☐ Environment mismatch (OS/runtime/container differences) :   
☐ Clock/timer glitch (system‐time jumps, NTP corrections)   
☐ Resource exhaustion (file-descriptor, thread-pool, memory leaks) :

* + 1. **Code & Configuration Defects**  
       ☐ Coding errors or bugs (including regressions)   
       ☐ Update/patch introduced or worsened bugs   
       ☐ Incorrect modification/adaptation (requirements mismatch)   
       ☐ Configuration drift (out-of-sync config or library versions)   
       ☐ Third-party component/library failure:   
       ☐ Security breach (malicious code or data tampering)
    2. **Missing/Flawed Functionality**  
       ☐ Feature degradation (functions lost or reduced capability)   
       ☐ Incomplete/broken features due to insufficient testing:   
       ☐ State-serialization or schema failures (persistence bugs)
    3. **Timing & Concurrency Violations**  
       ☐ Feedback/input timing violation (arrives too early/late, or stops too soon/continues too long)  
       ☐ PM (process model) not updated when needed  
       ☐ Concurrency/race‐condition bugs (unexpected thread interleaving)   
       ☐ Task priority inversion or starvation (one routine blocks another)
    4. **Performance & Resource Issues**  
       ☐ Performance degradation (slowdowns, high CPU/memory/disk use)   
       ☐ Float‐rounding or precision loss anomalies  
       ☐ Middleware/network latency or message corruption
    5. **Control-Logic & Fallback Design**  
        Controller defaults to fallback mode and ignores key functions

☐ Default mode

☐ Trigger condition

☐ Ignored functions  
☐ No alternative control actions available (design limitation)   
☐ Controller received misleading data that justified wrong action   
☐ Lame-duck mode (server shutting down or migrating)

* + 1. **Feedback & Input Integrity**

☐ Conflicting or contradictory inputs from other controllers, either higher or parallel *<tool provides names from control structure>*☐ Missing feedback caused guesswork or inactivity  
☐ Input/output semantic mismatch (wrong data type, locale/format error)  
☐ Dropouts (missing signals/data) or data corruption  
☐ Feedback in unusable format for controller (e.g. UI ambiguity)

* + 1. **Adaptivity & Learning Risks**  
       ☐ Adaptive logic error (model drift, over-fitting, wrong reward signals)  
       ☐ Opaque reasoning (cannot explain decision path) :

TOOL now displays:

SOFTWARE [controller name] process model

TOOL displays all of the items that USER selected from this section with open check boxes.

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**Tool displays remaining selected items.**

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**This is repeated until all selected items are accounted for.**

**>> Go to Class 2 Software Section 2.1**

* 1. **[Section Class 1 CS – Human controller]**
     1. **For each human in the system conduct the following. For teams a category can be used.**
     2. **The following should be completed prior to beginning further analysis. The first section is for the analysis of personality traits using the Five Factor model. User input from analysis (can be compiled from direct testing, or indirectly through interviews with family and associates) or select a generic category from Chaparro et al (2020) – Scale is a 5-point likert scale, 0 through +5 with 2.5 = average. For generic categories the average values for the category are applied for all pilots in that category.**

**For TEAMS use one of the categories below to apply to the entire team. If the team is analyzed individually (post accident when unique data is possible) then use the average of the scores for the entire team.**

**Analyst selects one of the following three categories:**

☐ **General Population (tool enters all 2.5 for behind the scenes weighting)**

☐ **Commercial Pilot (tool uses the following values for behind the scenes weighting)**

**Neuroticism = 1.5**

**Extraversion = -2.0**

**Openness = 2.0**

**Agreeableness = 2.5**

**Conscientiousness = 3.5**

☐ **Military Pilot (tool uses the following values for behind the scenes weighting)**

**Neuroticism = .7**

**Extraversion = 4.7**

**Openness = 2.75**

**Agreeableness = 1.1**

**Conscientiousness = 3.5**

☐ **Analysis obtained data. Obtain data working with psychologists compiled from direct testing of the individuals or via proxy through interviews and other information.**

**Analyst to normalize each item to one of the below values as appropriate.**

5 = Well above average

4 = Above average

3 = Somewhat above average

2.5 = Average

2 = Somewhat below average

1 = Below average

0 = Well below average

For each of the categories:

Neuroticism\_\_\_

Extraversion\_\_\_

Openness \_\_

Agreeableness\_\_\_

Conscientiousness\_\_\_

**The values are then implemented into the analysis below showing the score.**

**If evaluating individual humans continue to Step 1.2.3**

**If evaluating a TEAM go to Section 1.3**

**STPA Class 1 Scenarios for Human Controllers**

* + 1. **Preliminary Context & Causal Contributors**
       1. **Open Brainstorm**
          1. Why might the human controller make this decision?
          2. Consider this question in your answer to all of the facets involving humans: *Does the system enable the person to have information and ability to adapt when procedures or system design is not moving in a safe manner?*

Open answer -

☐ **Does the person have feedback from the process [if no, then create a requirement for it]?**

☐ **Missing/Conflicting Communication**

**USER Check all that apply in the following sections.**

* + - 1. **Training & Procedure Flaws**

☐ Flaw in training or procedures

☐ Incorrect new training or procedures

☐ Trained to act a specific way if “***,” then “***,” ignoring “***” & “***.”

☐ Conflicting training made action appear reasonable

☐ No alternate means available; action seemed only feasible choice

* + 1. **A. OBSERVE**
       1. **Situational Awareness (Endsley)**  
          ☐ Level 1 (Perceive): Are critical cues noticed? ☐ Level 2 (Comprehend): Are cues correctly interpreted?   
          ☐ Level 3 (Project): Are future states anticipated?
       2. **Perceptual Factors**

☐ Display not selected / wrong indicators

☐ Data stale (failure to refresh)

☐ Distraction / tunnel vision

☐ Fatigue / overload

* + - 1. **Time & Cognitive Load**

☐ Time pressure (compressed window)

☐ Inadequate time to gather/process (intrinsic vs. extraneous load)

* + - 1. **Communication Threats**

☐ Missing/misleading feedback from others

☐ Language/cultural barrier, style difference

☐ Power-distance inhibited feedback

☐ Non-verbal loss (radio/remote)

* + - 1. **Personality & Arousal**

>>[For the following, the tool should ONLY display the applicable item based on the values from section 1.2.2, meaning that if an item is above average do not display the below average option].

**Neuroticism** [TOOL IMPORT values from section 1.2.2, automatically SELECT item for user if value meets criteria]. If imported value is:

* + - [Autoselect if Above average and display] Hyper-alert, may be concerned about everything to the point of distraction and tunneling into a higher perceived threat. More susceptible to availability and representative heuristics resulting overestimating importance of a particular perceived threat that seems intuitive over more subtle factors that actually present higher risk. Less likely to conduct systematic scanning.

**Extraversion** [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] Above average the person can be more reliant on socially reinforced cues and not methodologically evaluate all signals.

**Openness** [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] may lead to broader information gathering. More curious scanning for new or unusual information.

[Autoselect if below average and display] If below average input the may result in missing novel cues. May overlook novel patterns that do not fit familiar patterns, anchoring bias (relying on the first set of information encountered) and confirmation bias.

**Agreeableness** [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average might tend to follow the flow of the group overriding other concerns affecting both initial inclinations and deeper contemplation.

**Conscientiousness** [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average supports thorough and organized observation. Habit-driven attention to familiar signals and diligent systematic monitoring of all inputs, more systematic analysis.

[Autoselect if below average and display] Lower than average may lead to neglecting details, attentional bias of focusing more on certain stimuli while neglecting others.

* + - 1. **Human-Software Interaction**

☐ Operator unaware of mode changes

☐ Important elements not distinct / hidden

* + - 1. **Countermeasures**

☐ Operator deliberately identify potential threats

* + 1. **B. ORIENT**
       1. **Mental Model Alignment**

☐ Flawed or outdated model of system/automation

☐ Uncertainty in mental model of the controlled process

* + - 1. **Goals & Conflicts**

☐ Safety vs. performance conflict

☐ Misaligned incentives/bonus structures

☐ External pressures (deadlines, revenue targets)

* + - 1. **Communication & Team Cognition**

☐ Conflicting or ambiguous feedback

☐ Transactive memory: “Who knows what?”

☐ Power-distance / deference effects

* + - 1. **Psychological Traits**

>>[For the following, the tool should ONLY display the applicable item based on the values from section 1.2.2, meaning that if an item is above average do not display the below average option].

**Neuroticism** [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average can distort situational understanding under pressure or emotional stress. Worry-driven decisions which can derail systematic reframing of the problem, less likely to systematically frame the problem:

**Openness** [TOOL IMPORT values from section xx]. If imported value is:

[Autoselect if Above average and display] Higher than average affects cognitive flexibility; high openness allows considering alternative interpretations. Less likely to take the intuitive approach to problem solving, more likely to use a systematic analysis. Less likely to stick with the status-quo. Less likely to stereotype. More likely to be open to new information and change direction accordingly:

[Autoselect if Below average and display] Lower than average – may be closed to alternative interpretations: .

**Agreeableness** – [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average may bias orientation toward social cues or perceived group consensus. May accept the first group-shared model:

[Autoselect if Below average and display] Lower than average – may bias against group consensus: .

**Conscientiousness** score [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average promotes consistent interpretation and comparison against standards or rules. More willing to rebuild or test mental model.: .

[Autoselect if Below average and display] Lower than average may lead to neglecting details:

☐ **Self-Efficacy & Locus of Control:** Internal vs. external

[score] Self-efficacy is a belief in one's own competence and effectiveness. Locus of control is the extent to which someone believes they can control events that affect them

☐ **Emotional Intelligence:** Self/social awareness

* + - 1. **Metacognition**

☐ Monitoring of thought processes

* + 1. **C. DECIDE**
       1. **Decision Dynamics**

☐ Loss Aversion / Prospect Theory impact

Prospect theory affects decisions by showing that people evaluate outcomes relative to a reference point—becoming risk‐averse when facing potential gains but risk‐seeking when trying to avoid equivalent losses.

☐ Sunk Cost Fallacy/Continuation bias

☐ Ambiguity Effect (choosing the known quantity even though it might be more risky but it is familiar)

* + - 1. Personality Factors

[For the following, the tool should ONLY display the applicable item based on the values from section 1.2.2, meaning that if an item is above average do not display the below average option].

**Neuroticism** [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] High neuroticism can make a person more risk averse so they lack confidence and then delay decision-making but then get forced to make a quick decision absent deep consideration, based on gut choices. They are unable to make calm, deliberate decisions. Factors such as ambiguity effect create difficult scenarios

**Extraversion** – [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average may influence preference for action-oriented or assertive decisions

**Openness** – [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average less likely to make decisions based on intuition, the person will consider a wider range of possibilities:

**Agreeableness** [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average may lead to conflict-avoidant decisions or deferring to others, overweighting of other people and organizational interests: Bandwagon effect, less likely to challenge other’s decisions.

[Autoselect if Below average and display] Lower than average – might not give deference to others opinions.

**Conscientiousness** score [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average enhances decision reliability and adherence to procedures.

[Autoselect if Below average and display] Lower than average – may reduce decision reliability and adherence to procedures.

* + - 1. **Hazardous Attitudes**

☐ Anti-authority

☐ Impulsivity

☐ Invulnerability

☐ Macho (over confidence bias) :

☐ Resignation

* + - 1. **Organizational Pressures**

☐ Performance metrics vs. safety tension

☐ Decision concentrated in few roles

* + - 1. **Means-Ends Reasoning (Rasmussen)**

☐ Purpose aligned with goals?

☐ Constraints respected?

☐ Abstract/Generalized/Physical function clarity?

* + 1. **D. ACT**
       1. **Execution Quality**

☐ Correct action applied too early/late/long/short

☐ Procedural compliance vs. improvisation

☐ Action bias / Illusion of control / Reactance

* + - 1. **Controls & Affordances**

☐ Ergonomic accessibility under stress

☐ Tactile/auditory/visual feedback present

☐ Safeguards (interlocks, alarms) engaged?

* + - 1. **Follow-up & Learning**

☐ Reflection on outcome (retrospection)

☐ Adjustments to training/P&P?

* + - 1. Personality factors

[For the following, the tool should ONLY display the applicable item based on the values from section xx, meaning that if an item is above average do not display the below average option].

**Neuroticism** [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] High neuroticism may lead to hesitations or overcorrections during execution. This can translate to higher gain control response. In decision making this can result in impulsive decisions and less thoughtful and controlled interactions:

[Autoselect if Below average and display] Lower than normal – More anticipatory control patterns, less impulsive response.

**Extraversion** [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average supports assertive action and rapid execution:

**Openness** [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average supports more flexibly adopting new procedures:

**Agreeableness** score [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average may result in going along without questioning procedure :

**Conscientiousness** score [TOOL IMPORT values from section 1.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average ensures precise and disciplined execution of actions:

[Autoselect if Below average and display] Lower than average – may reduce decision reliability and adherence to procedures.

TOOL now displays:

HUMAN [controller name] process model

TOOL displays all of the items that USER selected from this section with open check boxes.

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**Tool displays remaining selected items.**

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**This is repeated until all selected items are accounted for.**

**>> Go to Class 2 for humans – Section 2.2**

* 1. **Class 1 scenarios for small teams**
     1. **Consider this question for all facets as you work through them: *Does the system enable the team to have information and ability to adapt when procedures or system design is not moving in a safe manner?***

**Open answer here:**

☐ **Does the team have feedback [if not, create a requirement for it]?**

* + 1. **Observe**

*(Gather & monitor data, detect anomalies)*

* + - 1. **Shared Situational Awareness**

☐ CAST - Did all members maintain a shared situational awareness (mental model)?

☐ Are there design attributes to ensure all members maintain a shared situational awareness (mental model)?

☐ Model of the system misaligned across team

☐ Shared situational awareness intact?

☐ Feedback not shared across members, does the design provide for ways to ensure feedback is shared?

☐ Feedback arrive too late or too early?

☐ Input overloaded, missing, or misunderstood?

☐ Inputs or feedback received by one member but not shared?

☐ Information withheld, lost, or miscommunicated?

☐ Feedback formats (verbal/non-verbal/tools) presented in ways that adversely affect understanding?

☐ Feedback contradictory or difficult to resolve?

Personality metrics [TOOL IMPORT values from section 1.2.2 and average them]. If imported value is:

[Autoselect if Above average and display] If the average score for **Openness** is below average and high in the Extraversion/Agreeableness facets the group may may latch onto the first “obvious” hazard or cue and be blind to subtler signs of problems.

[Autoselect if Above average and display] If the group has an above average score for **Neuroticism** they can be hyper-alert to false positives and chasing more minor threats rather than real threats .

* + 1. **Orient**

*(Make sense of what you’ve seen, build/update your model)*

* + - 1. **Process Models & Mental Models**

☐ Was the team’s understanding of the process model accurate and up to date?

☐ Shared situational awareness intact?

☐ Were mental model parts misaligned across the team?

☐ Was there confusion about boundaries of team or individual responsibilities?

☐ Are there power-distance issues or perceived authority conflicts?

* + - 1. **Threat & Error Precursors**

☐ Fatigue or cognitive overload present?

☐ Time compression influencing decisions?

☐ Conflicting motives between team and organizational goals?

☐ Organizational culture pressure influencing risk tolerance?

* + - 1. **Personality & Behavior**

Personality metrics [TOOL IMPORT values from section 1.2.2 and average them]. Automatically display score. If imported value is:

[Autoselect if Above average and display] Teams that are high on **Openness** are more likely to seek and use new information to change their framing of a problem.

[Autoselect if Above average and display] Teams that are above average on the **Agreeable and Extraverted** scales may coalesce too quickly around an initial mental model and resist deeper evaluation (Groupthink).

[Autoselect if Above average and display] Very high average scores on **Conscientiousness** across the team can result in the group getting stuck in the process and lose decisional momentum.

☐  **Anti-Authority (“Don’t tell me what to do”)**

* Team members routinely bypass or ignore established procedures or protocols
* Open dismissals of “that’s just bureaucracy” or “we don’t need that formality”
* Reluctance to escalate concerns or to consult supervisors

 **Impulsivity (“Do something—quickly!”)**

* Decisions made without gathering necessary data or viewpoints
* Frequent “let’s just try it” mindset overriding risk assessment
* Cutting off discussion or ending meetings as soon as any choice emerges

 **Invulnerability (“It won’t happen to us”)**

* Overconfidence in team capabilities or past success stories
* Underestimating external threats or dismissing “what-if” scenarios
* Low uptake of lessons learned from past mistakes

 **Macho (“I can do it—show me!”)**

* Pride-driven risk-taking (“We’ll show them!”) instead of measured analysis
* Demeaning cautionary voices as “too timid” or “not team players”
* Celebrating near-misses as feats rather than warnings

 **Resignation (“What’s the use?”)**

* Fatalistic statements (“Nothing we do matters anyway”)
* Declining to propose improvements or to report hazards
* Low team morale, high passivity, or reluctance to act when concerns arise
  + 1. **Decide**

*(Choose a course of action, plan mitigations)*

* + - 1. **Decision-Making Characteristics**

☐ Was team decision-making responsive enough?

☐ Was the frequency/complexity of decisions overwhelming?

☐ Did the team need to coordinate two or more related decisions?

☐ Was there confusion about roles or decision authority?

☐ Did distributed decision-making lead to conflicting or duplicated actions?

☐ Were team members influenced by each other’s motives?

* + 1. **Act**

*(Implement, monitor effect, loop back if needed)*

* + - 1. **Action Execution & Control Paths that led to hazard**

☐ Was the team action correct but applied too early/late/too long?

☐ Was the same action attempted by multiple members simultaneously?

☐ Were control paths unclear or contested between members?

☐ Were physical or procedural interlocks absent (error containment)?

**Support & Mitigations**

☐ Was training sufficient for coordinated decision-making?

☐ Were standard operating procedures followed or known?

☐ Were mechanisms in place for communication verification?

☐ Were feedback loops explicitly structured in the task?

☐ Was a shared information space/tool used or needed?

☐ Could barriers (alarms, design) or briefings have prevented this?

* + 1. **Team Adaptive Capacity**

☐ Controllers offered/requested assistance to other controllers:

☐ Empathy for others controllers needs:

☐ Adaptive reconfiguration (e.g., envelope protection):

TOOL now displays:

TEAM [controller name] process model

TOOL displays all of the items that USER selected from this section with open check boxes.

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**Tool displays remaining selected items.**

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**This is repeated until all selected items are accounted for.**

**>> Go to Class 2 for Humans or Teams – Section 2.2**

* 1. **Class 1 scenarios for organizations**
     1. **Consider the following question in your answer to every item below: Does the system enable the person/team/organization to have information and ability to adapt when procedures or system design is not moving in a safe manner?**

**Open answer:**

☐ **Does the organization have feedback. [If it does not, create a requirement for it]?**

* + 1. **Observe**

*(What’s happening in and around the organization?)*  
**Inputs & Signals**

☐ Did the org receive accurate, timely data on its own performance, environment, and sub-units?

☐ Were external messages or partner/org communications misleading or wrong?

☐ Multiple inputs conflicted or amplified one another?

☐ Critical feedback was missing (e.g. filtered data, slow audits)?

☐ Information was outdated or incomplete?

☐ Reporting/feedback loops were delayed (quarterly, annual)?

☐ Overload prevented timely response (too many reports, too little analysis)?

* + 1. **Orient**

*(How does the org make sense of what it sees?)*  
**Context & Worldview**

**Why might the org take this action?**

☐ Organization not open to accepting new ideas or concepts.

☐ Mission/goals/metrics drove it?

☐ Policies/culture/procedures steered it?

☐ It lacked or misinterpreted input?

☐ Was there misalignment across divisions (false consensus)?

☐ Did org assume sub-units shared objectives when they didn’t?

☐ Flaws in strategic training or onboarding?

☐ Misaligned or conflicting policies?

☐ Role ambiguity or responsibility gaps?

☐ Performance pressure (deadlines, revenue) warped priorities?

☐ Org change (merger, restructure) introduced new risks?

☐ Legacy systems degraded processes or data quality?

 **Anti-Authority**

* Policies frequently overridden by leadership without accountability
* Compliance seen as optional, “red tape” rather than safety enabler
* Whistleblower or reporting channels are underutilized or discouraged

 **Impulsivity**

* Strategic initiatives launched with minimal analysis or stakeholder input
* Rapid “fire-and-forget” implementations, followed by frequent course corrections
* Reward structures that favor “speed to market” over thorough validation

 **Invulnerability**

* Organizational belief “we’re too big/good to fail” or immune to certain risks
* Risk registers stagnate, with threats labeled “unlikely” without review
* Insufficient investment in contingency planning or crisis simulations

 **Macho**

* Aggressive growth or cost-cutting programs that flaunt industry norms
* Public messaging that equates bold gambits with leadership prowess
* Penalizing dissenting opinions to maintain an image of strength

 **Resignation**

* Pervasive sense “we can’t change this” toward legacy systems or processes
* Innovation or improvement proposals routinely ignored or shelved
* High turnover or burnout among employees who feel powerless to effect change
  + 1. **Decide**

*(How does the org choose its course?)*  
**Options & Trade-offs**

☐ Fallback behaviors under stress (e.g., cost–cutting, delay responses) –

☐ No feasible alternatives due to resource or policy constraints?

☐ Decision-making concentrated in few individuals—bottleneck?

☐ Default incentives reward short-term gain over safe performance?

☐ Market, regulatory, legal, ethical constraints factored in?

☐ Strategic values (safety, quality, transparency) honored vs. sacrificed?

☐ Action emergent (ad hoc) or deliberately designed?

Properties that support adaptive capacity

☐ Is there a proactive preoccupation with failure:

☐ Is there a reluctance to simplify:

☐ Is there a deference to expertise:

* + 1. **Act**

*(Implement, monitor outcomes & loop back)*  
**Execution & Control**

☐ Did the org actually carry out the chosen action at the right time/scale?

☐ Responsiveness to new feedback (did you course-correct)?

☐ Bureaucratic or chain-of-command delays hindered execution?

☐ Conflicts between layers (exec vs. ops) slowed response?

☐ Control path vulnerable (single points of failure in roles or systems)?

☐ Did the organization adjust its resource allocation or budgets in real time?

**(Cross-cutting) Organizational “Personality” & Culture**

*(Factors that influence every OODA phase adapted to incude Five Factor Model)*

☐ How readily does the org adapt, experiment, seek outside input?

☐ Do procedures get followed with rigor and accountability?

☐ Are decisions and rationales communicated transparently?

☐ Is there genuine cross-unit collaboration or silence/dissent?

☐ Does the org panic or freeze under uncertainty/crisis?

TOOL now displays:

ORGANIZATION [controller name] process model

TOOL displays all of the items that USER selected from this section with open check boxes.

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**Tool displays remaining selected items.**

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**This is repeated until all selected items are accounted for.**

**>> Go to Class 2 for Organizations – Section 2.3**

1. **[Section Class 2 CS – Software]**
   1. **Software Class 2 Scenario**

**Software RECEIVES feedback that does not accurately reflect what the process it is controlling is doing.**

* + 1. **Scenario Identification**

Feedback/Input Type: TOOL IMPORT

Controller Receiving the Feedback/Input: TOOL IMPORT

Actual Process State (Context): **USER**

Did feedback/input accurately reflect this context?

* + ☐ Yes – **Skip to Class 3 Section xxx**
  + ☐ No (include designs where no feedback is provided) –

Process output incorrect, but feedback shows it as correct? ☐ Yes



Process output correct, but feedback shows it as incorrect? ☐ Yes

Possible Explanations (if either answer is YES)

(Check all that apply & describe)

☐ Measured metric estimated via mismatched assumptions/methods

☐ Feedback outdated due to delay (e.g., \_\_\_ s)

☐ Stale cache or buffer (interim store returned old data)

☐ Sensor input correct, but sensor output wrong

☐ Aggregation/roll-up errors (metrics combined incorrectly)

☐ Unit/scale mismatch (e.g., °C vs. °F)

* + 1. **Failure & Non-Failure Cases**

☐ Hardware/software failure in feedback path

☐ Concurrency/race condition

☐ Message reordering (out-of-sequence events) Describe scenario: **USER**

☐ Partial update/transaction abort (only some fields applied)

☐ No failure (sensor worked as specified) but feedback still wrong  
(e.g., valid but invalid-context measure):

* + 1. **Design & Concept Gaps**

☐ Feedback/information missing from design.

☐ Feedback not provided when needed.

☐ Feedback overload (too much info at once).

☐ No indication of context change due to process overload.

☐ Self-monitoring gap (controller fails to detect its own stale state).

☐ Heartbeats/timeouts missing (no “I’m alive” signal).

☐ Any other design/conceptual issue.

* + 1. **Feedback Quality & Timing Issues**

☐ Conflicting feedback/information.

☐ Incorrect feedback provided.

☐ Feedback sent too early (before event occurred).

☐ Feedback sent too late (after event occurred).

☐ Feedback/information incomplete.

☐ Measurement inaccuracies.

☐ Dropouts (missing signals/data).

☐ Corruption (data modified/erroneous).

☐ Feedback provided in unusable format (e.g., wrong encoding).

☐ Network-induced latency (delays in protocol, auth).

☐ Lock contention or deadlock (controller stalled).

☐ Authentication/authorization delays.

* + 1. **Software Configuration & Defaults**

☐ Unsafe default thresholds.

☐ Feature-flag misconfiguration.

☐ User Interface (UI) refresh throttling (deliberate update delay).

☐ Notification suppression (alarm silencing).

* + 1. **Security & Integrity**

☐ Data sanitization/filtering (legitimate feedback dropped).

☐ Replay attacks or spoofing (old/fake feedback injected).

* + 1. **Monitoring & Self-Diagnosis**

☐ Heartbeat/timeouts missing (no health checks).

☐ Self-diagnostic failure (feedback-channel health not monitored).

TOOL now displays:

SOFTWARE [controller name] FEEDBACK

TOOL displays all of the items that USER selected from this section with open check boxes.

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**Tool displays remaining selected items.**

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**This is repeated until all selected items are accounted for.**

**>> Go to Class 3 Software, Section 3.1**

**-----------------------------------------------------------------------------------------**

* 1. **Class 2 Scenario**

**Process is human or human team**

***Human-to-Human Feedback (Controlled Process is a Human)***

Use this form when the **controlled process** is a person, and the **controller** (another person or team) receives **feedback** from that human which does **not** match the **actual** human state or intent.

* + 1. **Scenario Identification**
       1. **Feedback/Input Type:** (e.g., verbal report, gesture, written note)

**Controller Receiving the Feedback:** >TOOL INPUT

**Actual Controlled-Person/Team State (Context):** :

**Did feedback/input accurately reflect this person’s state?**

☐ Yes Skip to Class 3

☐ No Describe discrepancy and continue with analysis. Include cases where there is no feedback provided :

* + 1. **Condition Being Analyzed**
* **Person’s behavior/outcome incorrect, but feedback portrays it as correct**
* **Person’s behavior/outcome correct, but feedback portrays it as incorrect**

**Possible Explanations (if either answer is YES)**

*Check all that apply & describe*

☐ **Estimation from mismatched assumptions**  
*(e.g., controller infers mood from tone but person’s affect differs)*:

☐ **Feedback outdated/delayed**  
*(e.g., person reported status earlier and situation changed)*:

☐ **Misinterpretation of non-verbal cues**:

☐ **Ambiguous language or terminology**:

☐ **Cultural/language barrier**:

☐ **Power-distance attenuation**  
*(e.g., subordinate hedges or omits bad news)*:

* + 1. **Failure & Non-Failure Cases**

☐ **Communication channel failure**  
*(e.g., phone static, software glitch during video)*:

☐ **Cognitive lapse by feedback-giver**  
*(memory lapse, fatigue, stress)*:

☐ **Intentional misdirection**  
*(e.g., “cover-up” or deceptive reporting)*:

☐ **No failure in channel, but feedback still inaccurate**  
*(e.g., person misreads their own vitals or misunderstands)*:

* + 1. **Design & Concept Gaps**

☐ **No standard protocol for status updates**:

☐ **Overload of simultaneous reports**:

☐ **No verification step (read-back)**:

☐ **Lack of structured confirmation (e.g., checklist)**:

* + 1. **Feedback Quality & Timing Issues**

*Check and describe how this leads to unsafe control action (UCA)*

☐ **Conflicting reports from same person**:

☐ **Incorrect report content**:

☐ **Report given too early** (before issue manifested):

☐ **Report given too late** (after window for action passed):

☐ **Incomplete report**:

☐ **Misleading or euphemistic language**:

☐ **Overly technical jargon**:

☐ **Non-verbal cues omitted** (e.g., body language):

☐ **Single-channel dependence** (only verbal, no written backup):

* + 1. **Human-Human Communication Factors**

☐ **Language or cultural barriers**:

☐ **Different communication styles** (direct vs. indirect):

☐ **Power-distance effects**:

☐ **Physical proximity / non-verbal nuance lost**:

☐ **Relay errors/latency when via intermediary**:

☐ **Interface limitations** (e.g., poor video quality in tele-ops):

* + 1. **Threat & Error Management (TEM)**

*For each, note whether threats were identified, managed, or led to error:*

☐ **Recognition of communication threat** (“I might be misunderstanding”) :

☐ **Use of cross-check/read-back**:

☐ **Backup channel activated**:

☐ **Escalation to higher authority**:

☐ **Failure to detect or mitigate threat**

* + 1. **Human Factors & Biases**

**Big Five Personality Impact on Feedback Interpretation**

>>[For the following, the tool should ONLY display the applicable item based on the values from section xx, meaning that if an item is above average do not display the below average option]. For teams the average score for each category is used.

[TOOL IMPORT values from section 2.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average the person or team sending the message may indicate that a threat is higher than it actually is. They also might not inform on threats that they do not deem as important or are missed due to the overly high focus on one that seems more salient to them.

**Openness** [TOOL IMPORT values from section 2.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average could result in broader cues.

[Autoselect if Below average and display] Lower than average may not share novel cues.

**Agreeableness** [TOOL IMPORT values from section 2.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average information could be attenuated if the sender believes it might come across as bad news.

]**Conscientiousness** [TOOL IMPORT values from section 2.2.2]. If imported value is:

[Autoselect if Above average and display] Higher than average then expect more thorough verification but also may follow procedures even when not appropriate, this could result in delays in providing information.

[Autoselect if Above average and display] Lower than average then details are more likely to be skipped.

* + 1. **Hazardous Attitudes**

☐ **Anti-authority** (ignored check-back requests)

☐ **Impulsivity** (accepted feedback without pause)

☐ **Invulnerability** (assumed “all is well”)

☐ **Macho** (downplayed conflicting reports)

☐ **Resignation** (did not challenge poor feedback)

* + 1. **Cognitive Biases**

☐ **Anchoring** (stuck on initial report)

☐ **Confirmation** (sought feedback that matched beliefs)

☐ **Attentional** (focused on one cue, ignored others)

☐ **Availability** (over-weighted recent feedback)

☐ **Framing** (interpreted feedback based on wording)

☐ **Observer-expectancy** (expected outcome influenced feedback)

* + 1. **Process Model Misalignment**

☐ **Controller’s mental model not updated**

☐ **Model updated too early/late**

☐ **Model continued changing after decision window**

☐ **Uncertainty in which behaviors are controllable**. Describe: **USER**

TOOL now displays:

HUMAN/TEAM [controller name] FEEDBACK

TOOL displays all of the items that USER selected from this section with open check boxes.

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**Tool displays remaining selected items.**

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**This is repeated until all selected items are accounted for.**

**>>Go to Section 3.2**

* 1. **Process is organization**

***Organizational-to-Human Feedback***

***(Controlled “process” is an organization; controller is an individual or team receiving status from that organization.)***

* + 1. **Scenario Identification**

Feedback/Input Type: (e.g., report, dashboard, email, verbal briefing):

Controller Receiving Feedback: (TOOL INPUT)

Actual Organizational State (Context) :

Did feedback accurately reflect the organization’s true state (organizational feedback over a controlled process - such as a regulator over an aircraft manufacturer - is generally via audits, checks, inspections and reports)?

☐ Yes Skip to Class 3

☐ No Describe discrepancy, including cases where no feedback is provided:

* + 1. **Condition Being Analyzed**

Org output incorrect but feedback portrays it as correct? ☐ Yes ☐ No

Org output correct but feedback portrays it as incorrect? ☐ Yes ☐ No

Describe scenario: **USER**

* + 1. **Possible Explanations**

*Check all that apply & describe*

☐ Data aggregation assumptions mismatch (e.g., KPIs use outdated metrics) :

☐ Reporting delays (batch reports vs. real-time) :

☐ Misinterpretation of qualitative vs. quantitative info :

☐ Mixed communication channels (email, verbal, system logs) :

☐ Intermediary filtering or summary bias (manager trimmed bad news) :

☐ Cultural/political influences (deference, groupthink) :

* + 1. **Failure & Non-Failure Cases**

☐ IT or system failure (dashboard offline, data corruption) :

☐ Organizational process lapse (missed audit, unreported incident) :

☐ Intentional omission (data suppressed for image) :

☐ Valid data but inappropriate metric (measuring wrong performance indicator) :

* + 1. **Design & Concept Gaps**

☐ No standard reporting cadence :

☐ Lack of read-back or confirmation :

☐ Overly complex dashboards (info overload) :

☐ No escalation criteria defined :

* + 1. **Feedback Quality & Timing**

☐ Conflicting reports from different departments :

☐ Incorrect summary insights :

☐ Report issued too early (before data stabilized) :

☐ Report issued too late (missed decision window) :

☐ Incomplete data fields :

☐ Ambiguous language or jargon :

* + 1. **Organizational Communication Factors**

☐ Hierarchical filtering (senior modifies junior’s report) :

☐ Power-distance silencing (subordinates withhold bad news) :

☐ Cross-cultural misinterpretation:

☐ Channel latency or relay errors:

☐ Interface/usability issues (poor BI tools) :

* + 1. **Threat & Error Management**

☐ Recognized reporting threat action: :

☐ Cross-check with alternate data source

☐ Escalation triggered

☐ Mitigation failed

* + 1. **Organizational-Level Biases & Attitudes**

Big Five Cultural Traits. Analyst needs to consider where the particular organization (that is being controlled by another organization) might fall with regard to these metrics.

**Neuroticism:**

☐ High (overcautious, e.g., every communication requires legal review) :

☐ Low (underreacts to risk, e.g. responds without any vetting to ensure data is accurate) :

**Openness:**

☐ High (adopts new metrics, very quickly, organization freely shares information) :

☐ Low (resists new information, minimal sharing of information ) :

**Agreeableness:**

☐ High (consensus bias, may be reticent to report bad news) :

☐ Low (siloed reporting, may only share the minimum required) :

**Conscientiousness:**

☐ High (rigorous audits, organization will ensure that all requisite information is provided) :

☐ Low (lax controls, information might be ad hoc, not reliable) :

* + 1. **Hazardous Organizational Attitudes**

☐ Anti-authority (ignores compliance reports) :

☐ Impulsivity (reacts to incomplete data) :

☐ Invulnerability (assumes “we’re too big to fail”) :

☐ Macho (bravado in communications) :

☐ Resignation (“cannot change culture”) :

* + 1. **Cognitive Biases**

☐ Anchoring (fixated on prior targets) :

☐ Confirmation (seeks data to support existing strategy) :

☐ Availability (overweights recent events) :

☐ Framing (presenting results to influence interpretation) :

☐ Groupthink (suppresses dissent) :

* + 1. Process Model Misalignment

☐ Mental model of org performance not updated:

☐ Model updated too early/late:

☐ Over-reliance on single performance indicator:

TOOL now displays:

ORGANIZATION [controller name] FEEDBACK

TOOL displays all of the items that USER selected from this section with open check boxes.

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**Tool displays remaining selected items.**

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**This is repeated until all selected items are accounted for.**

**>>Go to Class 3, Section 3.2**

1. **[Section Class 3]**

**CS – Software**

* 1. **STPA Class 3 Checklist: “Emulated” Unsafe Control Action**

*Scenario where a controller issues a* ***Safe Control Action (SCA)****, but the* ***Process*** *behaves* ***as if*** *an Unsafe Control Action (UCA) occurred.*

* + 1. **Scenario Identification**

**Controller** (name/role): TOOL IMPORT

***Controller did NOT issue a UCA but the process RECEIVED a UCA (control path problem).***

☐ Yes - Continue

☐ No **Skip to Class 4**

**Issued Safe Control Action (SCA)**: TOOL IMPORT/ **USER**

**Expected Process Behavior**:

**Observed Process Behavior (“as if” UCA)**: :

* + 1. **Control-Path Integrity Checks**

*For each, ☑ if applies; describe how it could transform SCA UCA*

| **Cause Category** | **Checklist Item** | **Notes/Description** |
| --- | --- | --- |
| **Transmission Delays & Ordering** | ☐ Control action delayed/dropped/out-of-order |  |
|  | ☐ Previous action still buffered, overriding SCA |  |
| **Signal Corruption & Resolution** | ☐ Data corruption (bit flips, noise) |  |
|  | ☐ Inadequate granularity (e.g., coarse value rounding) |  |
|  | ☐ Content truncated or incomplete |  |
| **Path Overload & Dropouts** | ☐ Control-path overload (queue / bus saturation) |  |
|  | ☐ Signal dropouts (intermittent connectivity) |  |
| **Security & Integrity** | ☐ Control spoofed, tampered, replayed |  |
|  | ☐ Action repudiated or denied (authentication failure) |  |
| **Logical Flags & Modes** | ☐ “Ignore” or mask bit set in transit |  |
|  | ☐ SCA overridden by fallback/emergency mode |  |

* + 1. **Conflicting Controllers**

☐ **Another controller** issued a stronger/conflicting command:

☐ **Multiple controllers** race on same parameter:

☐ **Intermediary (software/human/organization)** altered or suppressed the SCA :

* + 1. **Process-Side Reception Issues**

☐ Process input interface mis-mapped (e.g., wrong port/address) :

☐ Calibration/interpretation error (e.g., units mismatch) :

☐ Internal sensor/state stale—process “sees” old state, ignores new SCA :

☐ Internal watchdog overrides SCA as unsafe :

* + 1. **Failure vs. Non-Failure Modes**
       1. **Failure cases** (hardware/software fault):

☐ Physical layer break (wiring, bus fault) :

☐ Software exception / crash in control-path module :

☐ Buffer overflow leading to dropped SCA :

* + - 1. **Non-failure cases** (design/operational limitations):

☐ Design lacks coverage for the issued SCA in this context :

☐ Policy prohibits SCA under certain modes (e.g., manual override) :

☐ Resource constraints prevent processing of SCA :

* + 1. **Threat & Error Management (TEM)**

☐ **Anticipated hazards** identified? Mitigation in place? :

☐ **Cross-check path** available (e.g., secondary channel)? Used? :

☐ **Alarms/logging** triggered when SCA ≠ effect? Acknowledged? :

☐ **Fallback procedures** defined if process mis-acts? Followed? :

* + 1. **Analyst Notes & Additional Considerations**
       1. Other potential causes not covered above:

TOOL now displays:

SOFTWARE [controller name] SAFE control action, PROCESS receives UCA

TOOL displays all of the items that USER selected from this section with open check boxes.

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**Tool displays remaining selected items.**

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**This is repeated until all selected items are accounted for.**

**>>Go to Class 4, Section 4.1**

* 1. **STPA Class 3: Human Controller Human Process**

*When a human (or organizational) controller issues a Safe Control Action (SCA), but—due to “control‐path” breakdowns—the process behaves* ***as if*** *an Unsafe Control Action (UCA) occurred.*

* + 1. **Scenario Identification**

**Controller (Person/Org):** TOOL IMPORT

**Issued Safe Control Action:** : Describe: **USER >show list of possible control actions from Step 6**

**Intended Recipient (Human/Team):** >TOOL IMPORT from control structure

**Observed Behavior (“as if” UCA):** TOOL IMPORT UCA list

**Controller** (name/role): TOOL IMPORT

The process received an unsafe control action even though the controller did not issue an unsafe control action?

Yes ☐ Continue

Did the process behavior accurately match the control action?

☐ Yes – **Skip to Class 4**

☐ No Continue

**Issued Safe Control Action**: TOOL IMPORT/ **USER**

**Expected Process Behavior**:

**Observed Process Behavior (“as if” UCA)**:

* + 1. **Communication & Delivery Failures**

*Check any that apply and describe how the instruction was altered or lost*

| **Category** | **Checklist Item** | **Notes/Description** |
| --- | --- | --- |
| **Transmission Breakdowns** | ☐ Instruction not delivered (e.g., absent, unread email, missed call) |  |
|  | ☐ Significant delay before recipient received instruction |  |
| **Miscommunication** | ☐ Message content garbled or misremembered (paraphrasing error) |  |
|  | ☐ Recipient interpreted message differently than intended |  |
| **Channel Interference** | ☐ Noise/distraction during hand-off (e.g., radio static, busy hallway) |  |
|  | ☐ Medium unsuitable (e.g., text for urgent safety instruction) |  |
| **Filtering & Editing** | ☐ Middle-man (another person/org) altered or summarized instruction |  |
|  | ☐ Organizational policy stripped critical qualifiers or context |  |

* + 1. **Conflicting Inputs & Overlaps**

☐ **Another controller** gave a conflicting instruction to the same person/team:

☐ **Policy or rule** contradicted the controller’s instruction:

☐ **Automated system** (alarm, alert) overrode or confused the human instruction:

* + 1. **Recipient Reception & Interpretation**

☐ Recipient **did not hear/see** the instruction acted on default or guess: ☐ Recipient **misunderstood** key terms or priorities:

☐ Recipient’s **mental model** (beliefs/assumptions) led to wrong application:

☐ Recipient in **cognitive overload** or stress so shortcut decision path:

* + 1. **Failure vs. Non-Failure Contexts**
       1. **Failure Modes**

☐ Controller incapacitated or distracted instruction lost:

☐ Organizational breakdown (e.g., no handover SOP) :

* + - 1. **Non-Failure (Design) Limits**

☐ Instruction not within recipient’s role authority:

☐ Policy/procedure forbade immediate compliance:

* + 1. **Threat & Error Management (TEM)**

☐ **Anticipated miscommunications** briefed in training? :

☐ **Redundant confirmation** (read-back, acknowledgement) required & used? :

☐ **Cross-check** by second party available & used? :

☐ **Escalation path** if instruction unclear—was it followed? :

☐**Additional Observations** :

TOOL now displays:

HUMAN [controller name] SAFE control action, PROCESS receives UCA

TOOL displays all of the items that USER selected from this section with open check boxes.

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**Tool displays remaining selected items.**

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**This is repeated until all selected items are accounted for.**

**>>Go to Class 4, Section**

1. **[Section Class 4 CS**
   1. **– Software contoller– Hardware Process]**

**STPA Class 4 Checklist: Hardware Process Acts as if UCA Occurred Without Receiving It. If the process is SOFTWARE go to Section 4.2**

*Use this when the hardware process behaves “as if” it received an Unsafe Control Action (UCA), despite no such command arriving.*

* + 1. **Scenario Identification**

Did the process receive a SAFE control action but still behaved as if it received a UCA?

☐ Yes – **Continue**

☐ No – Skip this section

**Observed Unsafe Behavior (“as if UCA”):** [USER define]

* + 1. **Failure Cases (Hardware Faults)**

☐ **Component internal failure** (e.g. broken actuator, jammed valve)

☐ **Input interface failure** (e.g. sensor stuck, wiring break)

☐ **Power glitch or brown-out** affecting control electronics

☐ **Electromagnetic interference (EMI)** or lightning surge

☐ **Thermal or environmental damage** (overheat, moisture ingress)

* + 1. **Causes Without Hardware Failure**

☐ **Mode-based ignore** – device in fallback or safety mode that “auto-triggers” action

☐ **Out-of-range disturbance** (unexpected external force, shock)

☐ **Hardware limit reached** (e.g. end-stop, torque limit) causing rebound/return motion

☐ **Latency or sluggish response** – timing mismatch between subsystems

☐ **Default or “fail-safe” action** on loss of any input (e.g., valve dumps pressure)

* + 1. **Signal / Control Path Anomalies**

☐ **Signal dropout or corruption** in wiring, bus, or firewall

☐ **Stale buffered command** inadvertently applied later

☐ **Spoofed or phantom signal** injected by malfunctioning sub-system

☐ **Conflicting secondary command** from another controller or safety interlock

* + 1. **Environmental & Time-Dependent Changes**

☐ **Component wear** (backlash, seal wear) causing unintended motion

☐ **Drift or calibration loss** in position or pressure sensors

☐ **Thermal expansion/contraction** altering clearances or stiffness

☐ **Aging-related parameter shift** (e.g., actuator speed slowdown)

☐ No operation (stuck/locked)  
☐ Continuous operation (runaway speed)  
☐ Erratic operation (oscillation, chatter)  
☐ Incorrect direction or sequence  
☐ Unexpected power loss or surge

☐ Consider built-in protections: torque limits, fuses/breakers, E-stops, shields, interlocks

* + 1. **Additional Considerations**

☐ **Unexpected feedback loop** within hardware (oscillation, hunting behavior)   
☐ **Mechanical resonance** excited by unrelated vibrations   
☐ **Power-backup switchover** auto-activating alternate circuits   
☐ **Software/hardware handshake timeout** triggering default action   
☐ **Cross-talk** between adjacent channels (crosstalk on bus)

* + 1. **Threat & Error Management (TEM)**

**Redundancy:** ☐ Dual-channel sensors/actuators could have detected mismatch

**Monitoring:** ☐ Independent health-monitor flagged degradation?

**Alarms:** ☐ Would an alarm on “command ≠ action” mismatch have alerted operators?

**Fallback control:** ☐ Was there a safe-state strategy to prevent “ghost” actions?

TOOL now displays:

Hardware [process name] acts as if it received UCA even though none occurred

TOOL displays all of the items that USER selected from this section with open check boxes.

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**Tool displays remaining selected items.**

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**This is repeated until all selected items are accounted for.**

**COMPLETE FOR THIS CONTROL LOOP – GO TO NEXT CONTROLLER**

* 1. **Section Class 4 CS –**
     1. **Software is the process**

**STPA Class 4 Checklist: *Software* Process Acts as if UCA Occurred Without Receiving It**

*Use this when a software process behaves “as if” it received an Unsafe Control Action (UCA), despite no such command arriving.*

* + 1. **Scenario Identification**

Did the process receive a SAFE control action but still behaved as if it received a UCA?

☐ Yes – **Continue**

☐ No – Skip this section

**Observed Unsafe Behavior (“as if UCA”):** [USER define]

* + 1. **Software Fault Cases**

☐ **Buffer underrun/overrun** – old or out-of-order command in queue executed

☐ **Memory corruption** (bit-flip, heap overflow) causing spurious branch to UCA

☐ **Race condition / concurrency bug** – thread executes default/error handler

☐ **Exception handler default path** triggered (e.g., null-pointer catch)

☐ **Watchdog / timeout fallback** invoking safety routine

* + 1. **Control Path Anomalies**

☐ **Dropped message** on IPC or network, triggering “missing command” logic

☐ **Spoofed or replayed packet** from middleware or external source

☐ **Faulty serialization/deserialization** – mis-parsed control data

☐ **Incorrect debounce or rate-limit logic** causing unintended repeat action

* + 1. **Mode & State Mismanagement**

☐ **State machine mis-transition** to UCA-handling state

☐ **Default/initialization behavior** executed (e.g., on restart)

☐ **Configuration or feature-flag error** enabling unsafe code path

☐ **Failsafe or degraded mode** action mistaken for UCA

* + 1. **Timing & Synchronization Issues**

☐ **Clock skew / drift** causing scheduled UCA routines to fire prematurely

☐ **Timer wrap-around** or underflow triggering default event

☐ **Delayed callback or interrupt** handled as missed command

☐ **Task preemption** leading to execution of stale control data

* + 1. **Configuration & Data Errors**

☐ **Incorrect parameter value** (e.g., threshold) loaded at runtime

☐ **Corrupt or missing configuration file** triggering fallback logic

☐ **Security policy enforcement** (e.g., insufficient privilege) causing safe-mode action

☐ **Misaligned versioning** between control and execution modules

* + 1. **Environmental & Dependency Failures**

☐ **Library or API change** invoked unintended code path

☐ **External service outage** triggering default ‘fail-open’ or ‘fail-closed’ action

☐ **Resource exhaustion** (threads, file handles) leading to error handler path

* + 1. **Threat & Error Management (TEM)**

**Redundancy:** ☐ Was there a secondary validation path that could detect spurious action?

**Monitoring:** ☐ Did health-monitor detect abnormal state transitions?

**Alarms:** ☐ Would an alert on “unexpected action” mismatch have triggered?

**Degradation:** ☐ Was the software’s degraded-mode documented to avoid UCA-like behavior?

SOFTWARE [process name] acts as if it received UCA even though none occurred

TOOL displays all of the items that USER selected from this section with open check boxes.

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**Tool displays remaining selected items.**

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**This is repeated until all selected items are accounted for.**

**COMPLETE FOR THIS CONTROL LOOP – GO TO NEXT CONTROLLER**

* 1. **STPA Class 4 Checklist: Human Acts “As If” They Received an Unsafe Control Action**

**Use this when a human operator performs an unsafe action despite not having received or intended the UCA.**

* + 1. **Scenario Identification**

**Controller:** >TOOL INPUT

**Controlled Process:** >TOOL INPUT

Did the process receive a SAFE control action but still behaved as if it received a UCA?

☐ Yes – **Continue**

☐ No – Skip this section

**Observed Unsafe Behavior (“as if UCA”):** [USER define]

**Unsafe Action Observed (“as if” UCA):** [USER define]

**Context (actual process state):** [USER define]

* + 1. **Failure Cases (Physiological / Mechanical)**

☐ **Sensory impairment** (e.g., hearing loss, poor visibility) triggered unintended action  
  
☐ **Fatigue / microsleep** caused lapse into incorrect action:   
☐ **Medication or illness** impaired judgment or motor control  
    
☐ **Equipment interface malfunction** led operator to perceive a command

* + 1. **Non-Failure Causes (Cognitive / Contextual)**

☐ **Cognitive bias** – operator’s mind filled in missing command (“phantom command”)  
    
☐ **Habituation / expectation** – routine action triggered without prompt

☐ **Conflicting mental models** – misinterpreted situation as requiring UCA

☐ **Authority override** – assumed higher‐level directive without verification  
    
☐ **Social pressure** – peer or cultural norm led to pre‐emptive action

* + 1. **Communication & Organizational Influences**

☐ **Hearing assumed instruction** (e.g., misheard radio call)  
    
☐ **Omitted feedback** from teammates or management  
    
☐ **Power-distance effect** – deference prevented questioning nonexistent command  
    
☐ **Cultural/linguistic barrier** – mis‐interpreted informal cue as order

* + 1. **Training, Procedures & System Design**

☐ **Flawed SOPs** encourage “default” action absent clear input  
    
☐ **Inadequate training** on recognizing when not to act  
    
☐ **“If–then” rule misapplication** (trained “if X then act” but X never occurred)  
  Pattern: If , then \_\_\_\_\_\_; likely ignore \_\_-  
☐ **Conflicting procedures** made non-existent command appear plausible

* + 1. **OODA Loop Breakdown**

**Observe**

☐ **Attention capture** by irrelevant cue (e.g., alarm silence) -  
☐ **Tunnel vision** – missed that no order was given -  
☐ **Mis-recognition** – perceived system state as “command pending” -  
☐ **Sensory ambiguity** – poor signal (e.g., garbled comms) -

**Orient**

☐ **Mental model error** – believed context required UCA -  
☐ **Confirmation bias** – focused on cues that “justified” imagined order -  
☐ **Status quo bias** – “default” routine action felt safest -  
☐ **Macho / anti-authority** (CAST) – acted without checking -

**Decide**

☐ **Rapid‐prime decision** – RPD invoked under time pressure-   
☐ **System 1 haste** – skipped analytical check (Kahneman)-   
☐ **Impulsivity** (CAST) – acted before confirming-   
☐ **Invulnerability** (CAST) – believed action harmless

**Act**

☐ **Muscle memory** – habitual button press, lever movement-   
☐ **Overcorrection** – acted too long / too soon-   
☐ **Action bias** – chose action over verifying inaction-   
☐ **Resignation** (CAST) – assumed nothing else possible-

* + 1. **Threat & Error Management (TEM)**

**Recovered?** ☐ Yes ☐ No — how was phantom action reversed? -

**Detection:** ☐ Immediate self-check ☐ Peer callout ☐ Automated alert -

**Mitigation:** ☐ Safety interlock ☐ Emergency stop ☐ Procedure rollback-

* + 1. **Additional Considerations & Recommendations**

☐ **Means-Ends Hierarchy:** Did operator understand purpose function implementation? -

☐ **Prospect Theory:** Was action driven by loss aversion or risk framing? -

☐ **Organizational Culture:** Are workers encouraged to act without explicit command? -

HUMAN [process name] acts as if it received UCA even though none occurred

TOOL displays all of the items that USER selected from this section with open check boxes.

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**Tool displays remaining selected items.**

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**This is repeated until all selected items are accounted for.**

**COMPLETE FOR THIS CONTROL LOOP – GO TO NEXT CONTROLLER**

* 1. **STPA Class 4 Checklist: Organization Acts “As If” It Received an Unsafe Control Action**

**Use this when an organization takes an unsafe action despite not having received or intended the UCA.**

* + 1. **Scenario Identification**

**Controller:** >TOOL INPUT

**Controlled Process:** >TOOL INPUT

Did the process receive a SAFE control action but still behaved as if it received a UCA?

☐ Yes – **Continue**

☐ No – Skip this section

**Observed Unsafe Behavior (“as if UCA”):** [USER define]

**Unsafe Action Observed (“as if” UCA):** [USER define]

**Context (actual process state):** [USER define]

* + 1. **Failure Cases (Structural/Technical)**

☐ **IT system malfunction** led to false trigger of policy or automated action  
  -  
☐ **Communications infrastructure failure** caused mis-routing of instructions  
  -  
☐ **Physical security breach** (e.g., data center access) triggered emergency response  
  -

* + 1. **Non-Failure Causes (Procedural/Cultural)**

☐ **Policy ambiguity** caused executives to assume authority not granted  
  -  
☐ **Role confusion**—department acted autonomously without directive  
  -  
☐ **Legacy procedures** auto-activated outdated protocol  
  -

☐ **Social or political pressure** prompted pre-emptive organizational action  
  -

* + 1. **Communications & Inter-Unit Influences**

☐ **Mis-interpreted memo** circulated as mandate  
  -  
☐ **Power-distance effect**—junior units assumed unwritten executive directive  
  -  
☐ **Cultural/language barrier** in multinational units led to incorrect conclusion  
  -  
☐ **Latency in reporting**—delayed situational updates led to outdated action  
  -

* + 1. **Training, Procedures & System Design**

☐ **Flawed onboarding**—staff never trained to verify absence of directive  
  -  
☐ **“If–then” rule misapplication**  
  If \_\_\_\_\_\_ then \_\_\_\_\_\_, but \_\_\_\_\_\_ never occurred -

☐ **Conflicting SOPs** implicitly encouraged this action in absence of clear guidance  
  -

* + 1. **OODA Loop Breakdown**
       1. **Observe**

☐ **Data overload**—executives saw alarm dashboard and assumed action-  
☐ **Tunnel vision**—focused on single KPI, missed absence of directive-  
☐ **Mis-recognition**—treated “info request” as “action order” -  
☐ **Feedback missing** from compliance or legal teams-

* + - 1. **Orient**

☐ **Mental model error**—organization believed external mandate existed-  
☐ **Confirmation bias**—sought evidence supporting assumed directive-  
☐ **Status quo bias**—followed ingrained routine rather than verifying change-  
☐ **Authority bias**—assumed C-suite would have communicated needed change-

* + - 1. **Decide**

☐ **Rapid PR-driven response** under media pressure (Recognition-Primed Decision) -  
☐ **System 1 (quick response based on “gut” feeling) over System 2 (careful analytical analysis)**—skipped detailed legal/ethical review-  
☐ **Performance‐metric pressure** led to hasty decision-  
☐ **Overconfidence bias**—believed decision safe without full analysis-

* + - 1. **Act**

☐ **Procedure auto-triggered**—systems executed action despite no new input-  
☐ **Policy enforcement teams** carried out order without validating origin-  
☐ **Action bias**—preferred doing something over waiting-

* + 1. **Threat & Error Management (TEM)**

**Detection:** ☐ Audit flags ☐ Whistleblower report ☐ Automated compliance check -

**Mitigation:** ☐ Emergency rollback ☐ Public statement ☐ Policy update

-

* + 1. **Additional Organizational Considerations**

☐ **Means-Ends Alignment:** Did the action align with organizational mission functions resources? -

☐ **Prospect Theory:** Was the action driven by loss aversion (fear of blame) vs. gain? -

☐ **Inter-Unit Coordination:** Were all relevant departments consulted? -

☐ **Voice Analysis (CAST):** Tone/pause in executive comms may have implied authority-

Organization [process name] PROCESS acts as if it received UCA even though none occurred

TOOL displays all of the items that USER selected from this section with open check boxes.

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**Tool displays remaining selected items.**

USER reviews all of the selected items, checks ones that may be similar and then

Describes scenario: **USER with + sign to add more.**

**This is repeated until all selected items are accounted for.**

**COMPLETE FOR THIS CONTROL LOOP – GO TO NEXT CONTROLLER**

**Once all of the control loops are considered, the final items must be considered:**

**STPA analysis:**

* **DYNAMICS AND CHANGES OVER TIME**

**Dynamics and changes over time – Accidents usually occur after some change. Think of something anything might be implemented, such as a design or procedure, that was initially a good idea but could became a problem because of changes that occur later since that time so the assumptions that the original idea were based on are no longer valid.** Describes scenario: **USER**

* **COMMUNICATION AND COORDINATION AMONG CONTROLLERS AND INTERACTIONS**

**Communications, coordination and interactions among controllers-- This refers to problems when several different controllers are trying to control the same thing and they are not coordinating with each other. A simple example would be two air traffic controllers both not talking to each other and a collision happens. Friendly fire accidents are often similar.**

**Many of these were captured for teams but consider if any others are possible.**

Describes scenario: **USER**

**CAST Analysis**

* **DYNAMICS AND CHANGES OVER TIME**

**Dynamics and changes over time – Accidents usually occur after some change. Was there any change in how the maintenance was performed, or how the aircraft was used? Any other changes in inspection requirements? Are the assumptions that went into the initial design still valid? An example is on the UPS 1354 accident the airport performed maintenance at a time that was historically quiet, but overnight delivery companies started arriving in that window and nobody bothered to change the time that maintenance was performed. Another example was that the initial TCAS assumption did not consider the reduced separation when RVSM was implemented (that, of course, was updated because the requirement was documented and tracked, see the MIT Leveson paper on the topic). Think of something that might have been implemented, such as a design or procedure, that was initially a good idea but became a problem because of changes that occurred since that time so the assumptions that the original idea were based on are no longer valid.** Describes scenario: **USER**

* **COMMUNICATION AND COORDINATION AMONG CONTROLLERS**

**Communications and coordination among controllers-- This refers to problems when two different people (or computers) are trying to control the same thing and they are not coordinating with each other. A simple example would be two air traffic controllers both not talking to each other and a collision happens. Friendly fire accidents are often similar. Imagine even two police officers not communicating and then coming upon one another and firing out of surprise.** Describes scenario: **USER**

* **INDUSTRY AND ORGANIZATIONAL SAFETY CULTURE**

**Industry and organizational safety refers to the design of the safety management system and the safety culture. How well as the SMS designed, was it designed to capture problems? Were there safety culture problems (see page 84 of the CAST handbook).** Describes scenario: **USER**

* **SAFETY INFORMATION SYSTEM**

**Safety information systems – This refers to how safety information is shared among different parts of an organization or agencies. For example, how much was the airline and manufacturer sharing safety concerns that were identified? What about the vendors? What about the regulators? In many cases a lot of events and incidents occurring are not widely shared with people who could use it. There is an ongoing discussion within the Royal Aeronautical Society about how airlines used to post all of the incidents, violations and events to a board so the pilots could read and learn from it. Today that does not happen, or it is very limited and only after it is distilled by various groups in the airline. That is an example of an inadequate safety information system.** Describes scenario: **USER**

**Step 7 Requirements and Mitigations**

* **For STPA Tool lists these are REQUIREMENTS**
* **For CAST Tool list these as MITIGATIONS**

**For each controller present the USER described causal factors for each control in groups based on the Class of scenario.**

**User is then prompted to write a requirement/mitigation to prevent the scenarios. USER can choose to combine several of the scenarios into one requirement.**

**TOOL starts at the bottom of the control structure in same manner as for UCAs and Scenarios**

* **For each controller TOOL INPUT from entered scenarios based on** 
  + **Class 1 List (TOOL INPUT from Step 6): Create Requirement: USER**
  + **Class 2 List (TOOL INPUT from Step 6): Create Requirement: USER**
  + **Class 3 List (TOOL INPUT from Step 6): Create Requirement: USER**
  + **Class 4 List (TOOL INPUT from Step 6): Create Requirement: USER**

**Final Step**

**Tool inputs the user created requirements and the associated scenarios for each controller into a table such as below. User checks box if the scenario will be mitigated from that requirements/mitigation.**

|  |  |  |  |
| --- | --- | --- | --- |
| **REQUIREMENT**  **MITIGATION** | **Scenarios [TOOL IMPUTS]** | | |
| **Scenario 1** | **Scenario 2** | **Scenario…n** |
| **Req/Mit 1** |  |  |  |
| **Req/Mit 2** |  |  |  |
| **Req/Mit 3** |  |  |  |
| **Req/Mit 4** |  |  |  |
| **Req/Mit n** |  |  |  |

**TOOL then should list the REQUIREMENTS/MITIGATIONS in the order of mitigating the MOST causal scenarios.**