33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

I-SHARE:

INCOSE Systems Heuristics Application Repository –

Sharing Systems Engineering Knowhow and Experience

Dov Dori, Dorothy McKinney, Gan Wang, and Scott Jackson

(Best Paper INCOSE IS 2023)

Heuristics: Origin, Defintions

- Ancient Greek: *heuriskein;* Latin: *heurisricus*
- Something found through *discovery* rather than *logic*
- *"**mental shortcuts** that allow people to **solve problems** and make **judgments** quickly and **efficiently** … **rule-of-thumb strategies** [that] shorten **decision-making** time and allow people to function without constantly stopping to think about their next course of action."* (Cherry, 2022)
- *"a mental shortcut that allows an individual to **make a decision**, pass judgment, or **solve** a problem **quickly** and with **minimal mental effort*** (Psychology Today, 2022).
- Context-sensitive; must be applied with judgment to the current or new system (Maier & Rechtin, 2009)

Systems Engineering Heuristics

- *"... a guideline for the **conduct** of **architecting**; lessons learned expressed as a guideline; a natural language abstraction of experience... trusted, nonanalytic guidelines for treating inherently unbounded, ill-structured problems... used as aids to decision-making, value judgments, and assessments"* (Maier & Rechtin, 2009, p. 31).
- A "rule of thumb" or snippet of practical advice, intended to reduce cognitive load.
- Provide guidance for choices or action
- Help solve problems quickly and effectively
- Based on informal experience of professionals who can generalize, abstract
- Provide "words of wisdom" to less experienced professionals.

Heuristics Discovery Cartoon showing engineers doing trial and error, discovering rules of thumb, writing them down as heuristics to share with others

Systems Engineering Heuristics: Maier & Rechtin, 2009, p. 31

architectures from scratch. By much the same process, qualitative heuristics, condensed and codified practical experience, came into being to complement the equations and algorithms of science and engineering in the solving of complex problems. Passed from architect to architect, from system to system, they worked. They helped satisfy a real need.

conjecture, and testing.* At their strongest they are seen as self-evident truths requiring no proof.

The art in architecting lies not in the wisdom of the heuristics, but in the wisdom of knowing which heuristics apply, a priori, to the current project.[2]

The I-SHARE SE Heuristics Project

- Launched at **Fellows Meeting** during the **International Workshop** in Jan. **2020**
  - Following a request from the **UK CAB** (Ian MacTaggart, UK Chapter Past President)
  - Endorsed by **INCOSE leadership**.
- **Requirements:** devise a set of readily available heuristics to be shared among systems engineers that:
  - Change an action that might otherwise, traditionally, occur without the heuristic
  - Be sufficiently **pithy** and **memorable** that they can be **recalled** when the challenge arises
  - Express **abstract phenomena** in a **simple**, **understandable** way
  - Emphasize **usefulness** rather than precision or universality
  - Be **updated** and remain **relevant** and referenced by systems engineers and others
  - **Complement** and **support** the challenges of the *SE Principles*, under INCOSE **Future of Systems Engineering** (FuSE) Program

Motivations for using I-SHARE

- **Reducing the time** needed to make a good decision or a choice, or enabling arrival at an equally good solution more quickly
- **Finding a solution** to a problem, especially a "**wicked**" one
- Seeking **guidance** to **simplify complex** or difficult **problems**
- **Identifying** the most **important factors** to focus on in addressing a complex problem, especially when time is short
- **Improving** the **quality of decisions** made by drawing on best practices

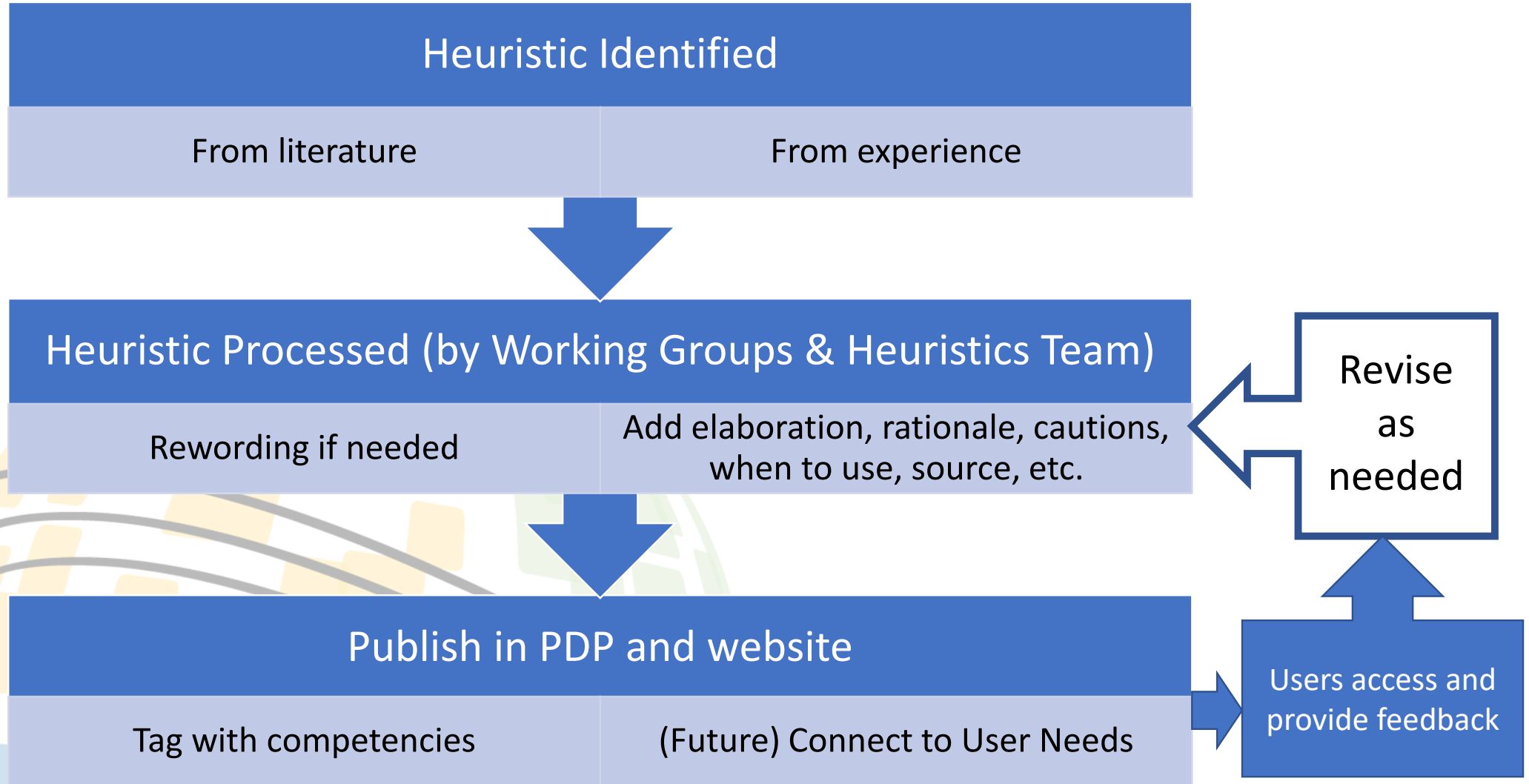
I-SHARE Primary Users

- **Systems engineers**, in performing systems engineering tasks, especially in **design reviews**, and **verification** and **validation** activities
- **Systems thinkers**, in **developing SE methods** and tackling general systems-related problems
- **Managers and decision-makers**, in **evaluating** design **proposals** and performing **decision making**, and for **improved oversight** of systems engineers' work
- **SE training and development programs** to augment organizational training materials with I-SHARE to improve the quality, efficiency and competency of their SE teams.

I-SHARE Secondary Users and Stakeholders

- **Engineers in other professions**, to apply the I-SHARE in their specialties
  - especially **software engineers**, since software systems are systems and software integrates diverse system parts
- **The academic community**, to enrich their SE **curricula** and **course materials** with I-SHARE as an **authoritative source** of teaching material
- **INCOSE**, to enhance its **reputation** and **support** its **mission** by promoting the use of I-SHARE
- Other **international organizations**, such as **IEEE** and **ISO**, to consider the use of I-SHARE heuristics in their standards and guidelines

Selecting I-SHARE Heuristics

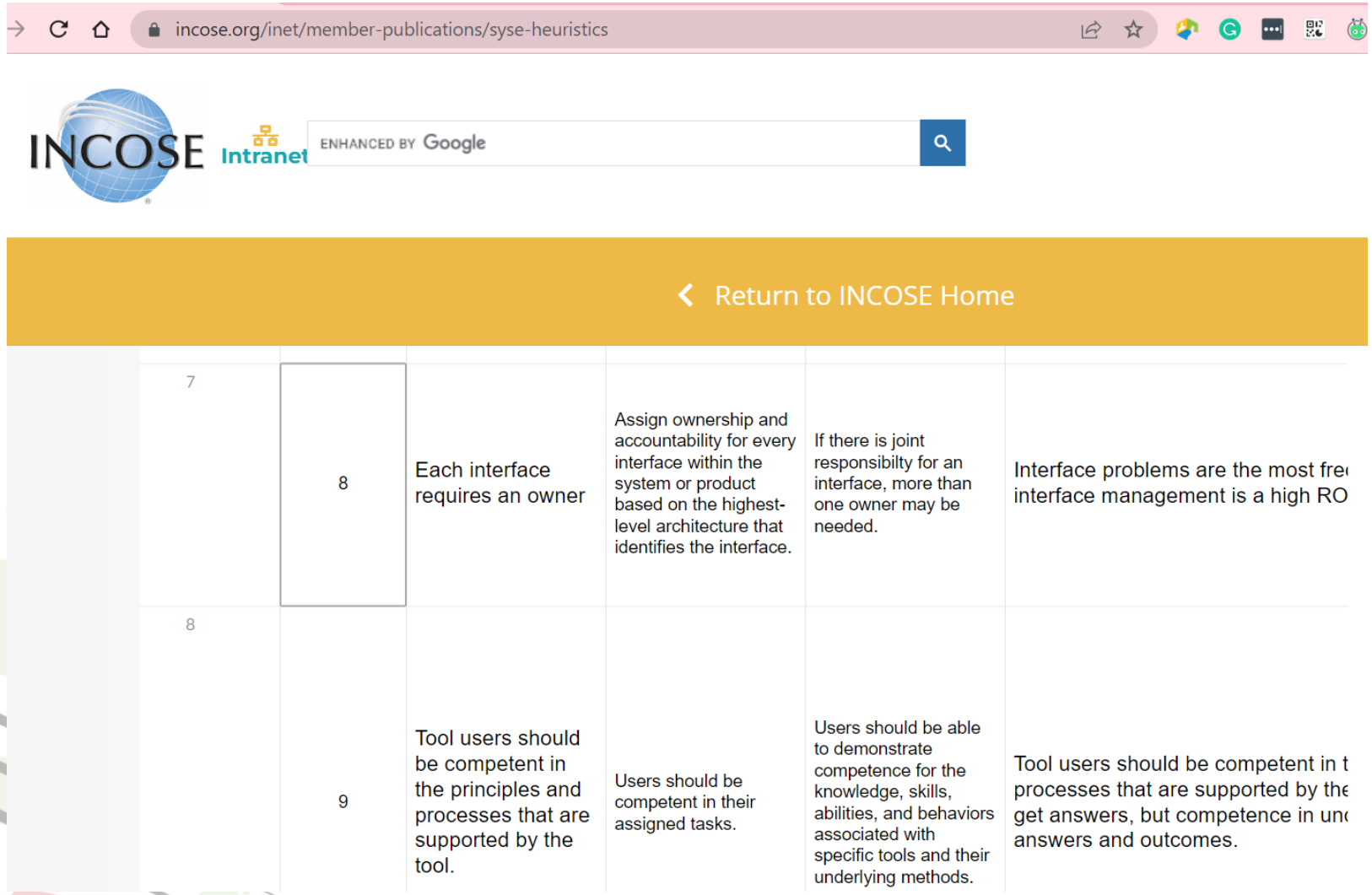


I-SHARE Heuristics selection criteria

No.	Criterion	Rationale	Comments
1	The heuristic must be **focused, clear, and concise.**	Even if the original language is not clear and succinct, we should be able to **rephrase** it in **clear** and **succinct** terms. To this end, we **allow** heuristics **rephrasing**.	The heuristic should be narrower and more focused than merely common sense or “motherhood and apple pie.” They should abstract experience, so lessons learned are passed to less experienced practitioners.
2	The heuristic’s **source** must be **authoritative**.	Ideally, each heuristic should be **published** or **suggested** by a **recognized expert** in the systems engineering field and **supported** by at least two others.	Newly expressed heuristics will probably need to be vetted more thoroughly than ones that have been published and are recommended by one of the I-SHARE team members.
3	The heuristic’s **use** can be **clearly articulated.**	Unless we can state when or how a heuristic can or should be used, it is not likely to serve our intended purpose in promulgating I-SHARE.	Uses could be for a specific of the SE lifecycle phase or a specific type of system, such as complex, socio-technical, cyber-physical, or System-of-Systems.
4	The heuristic must **make sense in its** original **domain** or context	The heuristic should also apply more widely, using heuristics extrapolation (Maier & Rechtin, 2009).	See the explanation on heuristics extrapolation by Maier and Rechtin (2009).
5	The heuristic should be **capable of being applied beyond its original context**.	The heuristic should be **useful** in **solving** or **explaining more than** the **original** problem from which it arose.	The heuristic should be **generalizable**, so it becomes **applicable across multiple domains**.
6	The heuristic can be **easily rationalized** in a few minutes of talk or in one medium-length paragraph.	**Terseness** eases **subsequent publication** and use.	Supporting material may need to be developed before the rationalization can be judged.

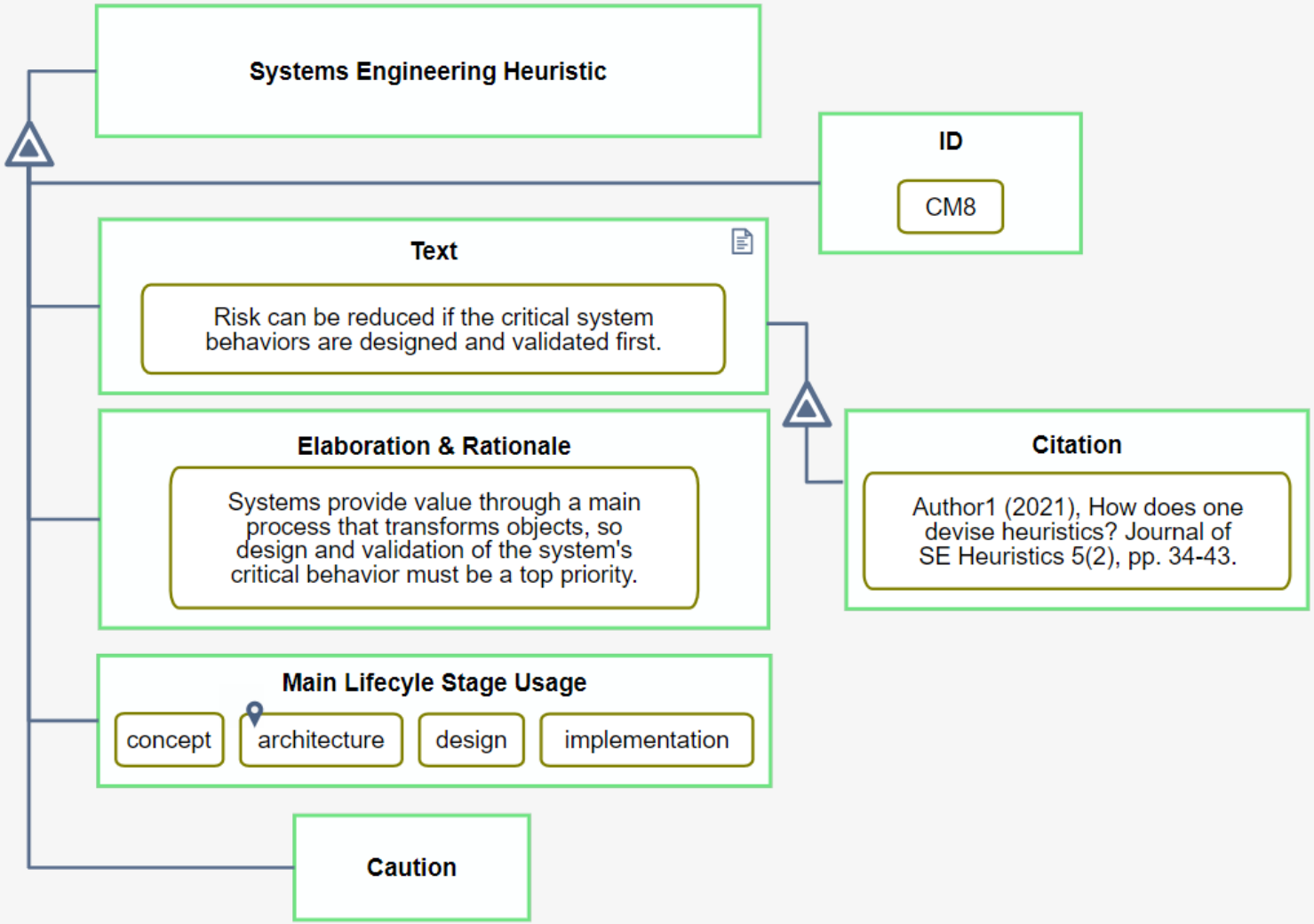
Accessing the I-SHARE Repository

- First log in to www.incose.org (heuristics are only available to INCOSE members)
- In a new window or tab go to https://www.incose.org/inet/member-publications/syse-heuristics
- Now you can search for relevant heuristics by filters, export, and use them for your needs



7	8	Each interface requires an owner	Assign ownership and accountability for every interface within the system or product based on the highest-level architecture that identifies the interface.	If there is joint responsibilty for an interface, more than one owner may be needed.	Interface problems are the most fre interface management is a high RO
8	9	Tool users should be competent in the principles and processes that are supported by the tool.	Users should be competent in their assigned tasks.	Users should be able to demonstrate competence for the knowledge, skills, abilities, and behaviors associated with specific tools and their underlying methods.	Tool users should be competent in t processes that are supported by the get answers, but competence in unc answers and outcomes.

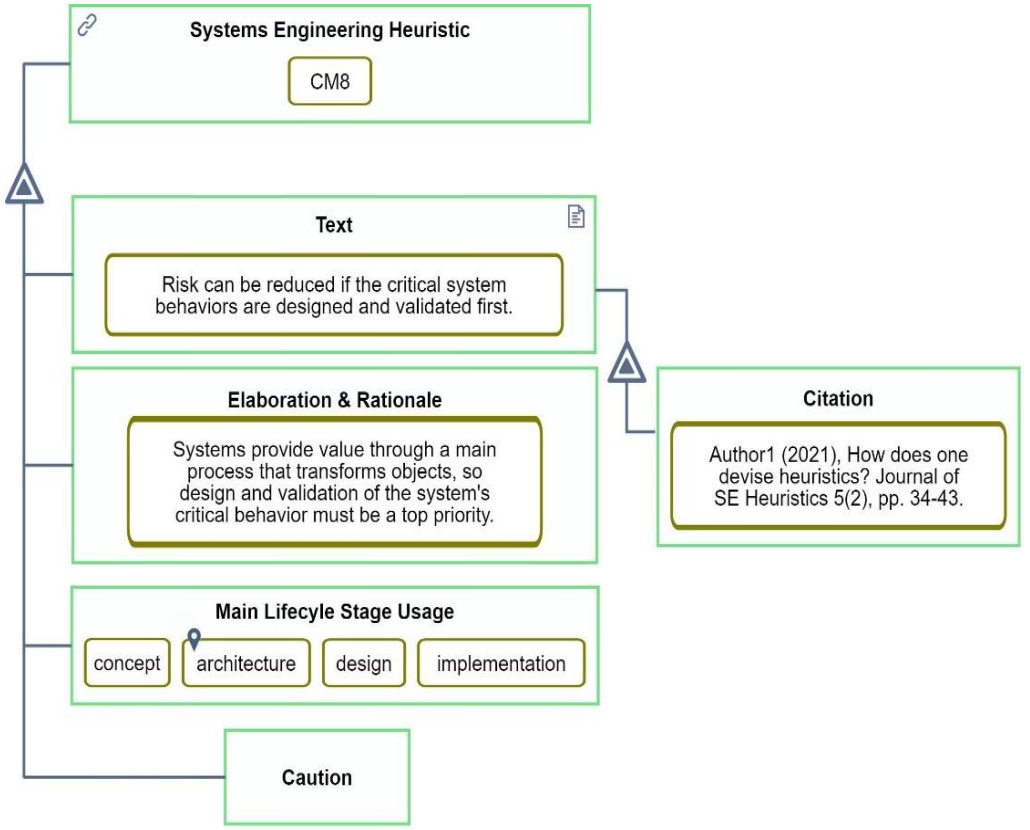
An OPM Heuristic Template populated with CM8



Text of **Systems Engineering Heuristic** is **Risk can be reduced if the critical system behaviours are designed and validated first.**
Elaboration & Rationale of **Systems Engineering Heuristic** is **Systems provide value through a main process that transforms objects, so design and validation of the system's critical behavior must be a top priority.**
Main Lifecyle Stage Usage of **Systems Engineering Heuristic** can be **architecture**, **concept**, **design** or **implementation**.
Main Lifecyle Stage Usage of **Systems Engineering Heuristic** is currently at state **architecture.**
Citation of **Text** is **Author1 (2021), How does one devise heuristics? Journal of SE Heuristics 5(2), pp. 34-43.**
ID of **Systems Engineering Heuristic** is **CM8.**
Systems Engineering Heuristic exhibits **Caution**, **Elaboration & Rationale**, **ID**, **Main Lifecyle Stage Usage** and **Text.**
Text exhibits **Citation.**

ChatGPT interpretation of the OPL of CM8

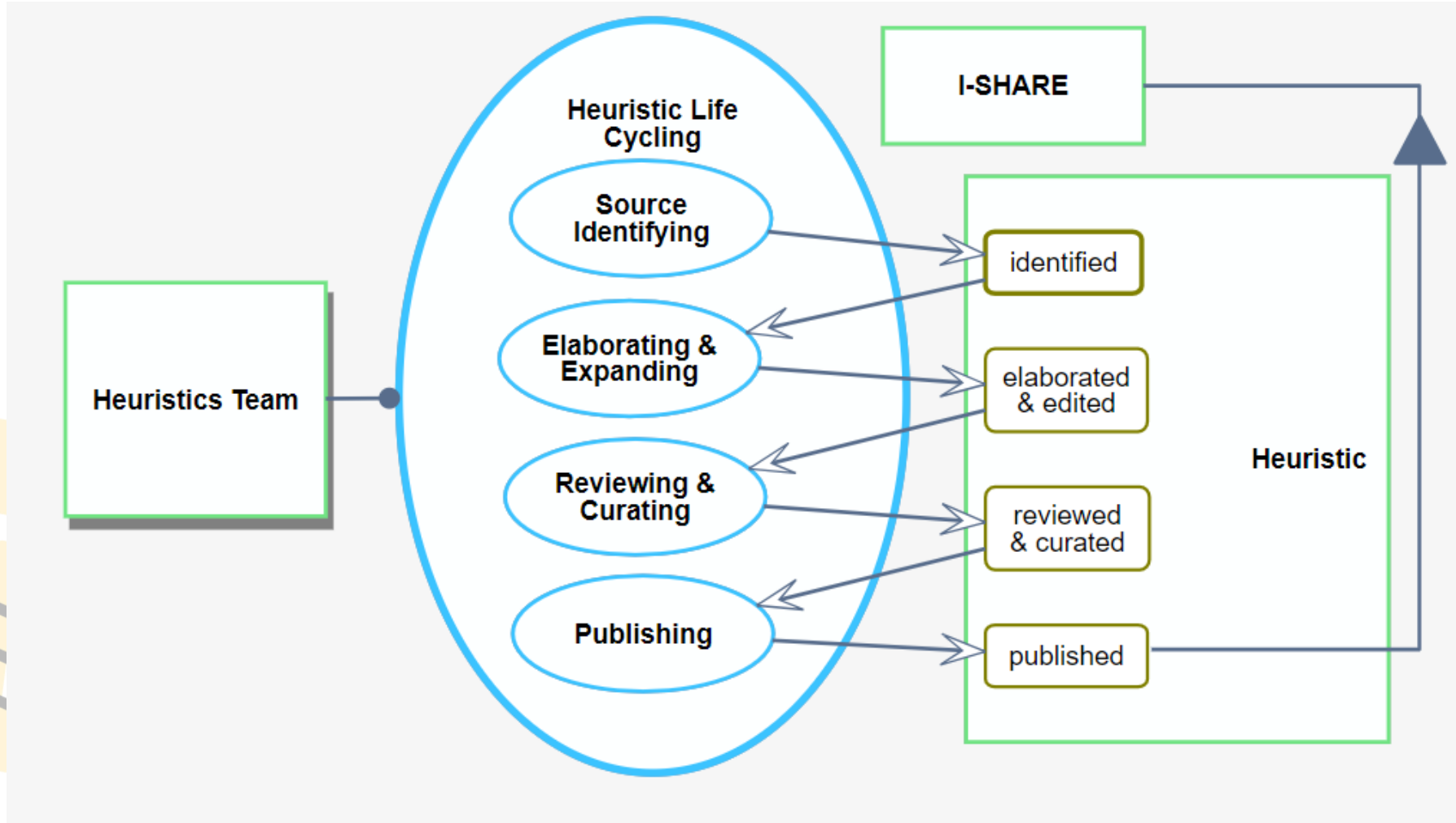
The Systems Engineering Heuristic with the ID CM8 states that "*Risk can be reduced if the critical system behaviors are designed and validated first*." As a systems engineer, this heuristic guides the approach to system development and highlights the importance of **focusing on the essential behaviors of a system during its design and validation stages**. The heuristic suggests that systems provide value through a primary process that involves transforming objects. Therefore, the **design** and **validation** of the critical behaviors of the system should be given **top priority**. By **addressing** these crucial aspects **early in the development lifecycle**, potential **risks can be identified, mitigated, and even eliminated before they impact the overall system performance and reliability**.



Currently, the Systems Engineering Heuristic with the ID CM8 is primarily used in the **architecture stage** of the system's lifecycle. During this stage, the emphasis is on **defining** the system's **structure**, **components**, and **interactions**.

By **considering** the **critical system behaviors upfront** and ensuring their design and validation, potential **issues** and **risks** can be effectively **addressed** at the architectural level, setting a **solid foundation** for **subsequent stages** of development.

An SE Heuristic Lifecycle: Graphic & OPD



An SE Heuristic Lifecycle: OPL

Heuristic Life Cycling from SD zooms in SD1 into **Source Identifying**, **Elaborating & Expanding**, **Reviewing & Curating**, and **Publishing**, which occur in that time sequence.
Heuristic can be **elaborated & edited**, **identified**, **published** or **reviewed & curated**.
State **identified** is initial.
I-SHARE consists of **Heuristic** at state **published**.
Heuristics Team handles **Heuristic Life Cycling**.
Source Identifying yields **Heuristic** at state **identified**.
Elaborating & Expanding changes **Heuristic** from **identified** to **elaborated & edited**.
Reviewing & Curating changes **Heuristic** from **elaborated & edited** to **reviewed & curated**.
Publishing changes **Heuristic** from **reviewed & curated** to **published**.

An SE Heuristic Lifecycle: interpretation of the OPL

The lifecycle of a systems engineering heuristic, as described in the OPL specification, consists of **four stages**: **Source** Identifying, **Elaborating** & **Expanding**, **Reviewing** & **Curating**, and **Publishing**. These stages occur in a specific **sequence**.

1. Source Identifying:
The lifecycle begins in the Source Identifying stage. At this stage, the heuristic is in the initial state, which is **identified**. The goal of this stage is to identify the source of the heuristic. Once the source is identified, the heuristic moves to the next stage.

2. Elaborating & Expanding:
In this stage, the heuristic undergoes elaboration and expansion. It is changed from the identified state to the elaborated & edited state. The Heuristics Team, responsible for managing the lifecycle of the heuristic, works on **refining** and **expanding** the **content** of the heuristic. This stage involves adding more details, examples, or clarifications to enhance the heuristic's effectiveness.

3. Reviewing & Curating:
After the Elaborating & Expanding stage, the heuristic enters the Reviewing & Curating stage. At this point, the heuristic is in the elaborated & edited state. The Heuristics Team reviews and curates the content of the heuristic to **ensure** its **accuracy**, **relevance**, and **quality**. This stage may involve obtaining **feedback** from **domain experts** or **stakeholders** to improve the heuristic further.

4. Publishing:
Once the heuristic has undergone the necessary reviewing and curation, it moves to the final stage, Publishing. In this stage, the heuristic is changed from the reviewed & curated state to the published state. The Heuristics Team makes the heuristic **available** to the **intended audience** or users. At this point, the heuristic becomes part of the I-SHARE (presumably a repository or collection of heuristics) for wider **dissemination** and **usage**.

To summarize the lifecycle stages:
1. Source Identifying: Heuristic is in the identified state.
2. Elaborating & Expanding: Heuristic is changed from identified to elaborated & edited state.
3. Reviewing & Curating: Heuristic is changed from elaborated & edited to reviewed & curated state.
4. Publishing: Heuristic is changed from reviewed & curated to published state and becomes part of I-SHARE.

The Heuristics Team is responsible for managing the lifecycle of the heuristic, guiding it through these stages until it is published and made available for use.

Use Case 1: Supplementing expert review and feedback

- Taylor, a young systems engineer, is tasked with planning a new project to develop a system comprising a component that is based on new, unprecedented technology.
- Having developed the plan, she asks experienced systems engineers to review it and provide feedback.
- As they are all too busy to spend the time needed to carry out a detailed review, they instead ask Janice to screen I-SHARE, select the heuristics she deems applicable to the project, and describe the extent to which her plan aligns with each of them.

Use Case 1: Two relevant heuristics and commensurable provisions taken

Heuristic	Elaboration	Rationale
Early in a high-risk project, both reducing major risks and making progress visible to stakeholders are important.	Activities which reduce technical risks do not necessarily produce visible progress that most stakeholders can see. Stakeholders need confidence they understand that useful progress is being made.	On a project with high technical risks, it is critical to give stakeholders confidence that useful progress is being accomplished and to ensure their continuous support to the project.
Use multiple perspectives to understand the problem	Understanding the Context and implication, Choosing the Right Problem to Solve, framing the Problem, defining the Criteria, understanding the Solution Space and the Possibilities. Understanding Implications of the choices balancing rational and emotional considerations.	Different stakeholders have different perspectives and understandings of what matters. Doing this will improve the quality of your final system solution, and the chances of it being widely adopted and used.

Use Case 2: Provide guidance from lessons learned in systems engineering

Applicable to many scenarios, including:

- **Professors teaching systems engineering** can task students with identifying I-SHARE heuristics that touch upon or **summarize lessons learned** in systems engineering case studies.
- While **teaching introductions to systems engineering**, selected heuristics can illustrate how **experienced systems engineers apply wisdom** acquired from past projects and phrased as I-SHARE heuristics to reduce program or project risks, increasing their success likelihood.
- A **mentor** can ask systems engineers to describe how they would **apply a certain I-SHARE heuristic** in their **plan** for a systems engineering effort, and how they use it as an argument to **convince stakeholders** of their plan's soundness.
- **Links** from the I-SHARE online resource to the ***Systems Engineering Body of Knowledge*** (SEBoK) can offer practitioners interested in honing their systems engineering skills how to **apply** a **heuristic** that is of interest to them in a concrete scenario.

Use Case 3: Providing ways to synthesize and evaluate a systems engineering product

- Extrapolate to turn a descriptive heuristic to a prescriptive one
- Specialize to attend to a given domain or context
- Applicable where there is no suitable modeling approach

Use Case 4: Organizational Culture Improvement

Example: Rolls Royce jet engine organization

1. A small team of expert systems engineers selects from I-SHARE heuristics deemed most relevant to the organization, optionally including “ringers”.
2. SE practitioners in the organization are divided randomly into two groups.
3. From the list, each group is asked to identify heuristics that are:
   a. most applicable to the organization (1st group)
   b. not applied but should be (2nd group).
4. The experts examine the results.

   1st group: Selection of "ringers" indicates how seriously the groups worked

   2nd group: Areas in which training or implementation need to be initiated or improved are identified

INCOSE Working Group Input Example

Resilient Systems Working Group
The system can resist the adversity and its impact and effect.
The system resists the adversity and its impact and effect to a defined extent.
To be resilient, the system is reconfigurable.
To be resilient a system can avoid an adversity by detecting the adversity in advance and making corrective action.
Design the system with functional redundancy or distributed capability.
To be resilient, a system is designed with multiple identical branches.

Summary, Current and Future Work

- SysE heuristics are ideally pithy and memorable, so they are expressed in a few simple sentences.
- Often metaphoric and may include known facts or stories to make a point or deliver a message.
- Structured – have attributes.
- Implementation in e.g., Smartsheet helps users search, query, understand the heuristic meaning and use, and associate it with related heuristics.
- The I-SHARE team diligently examines and discusses each heuristic in the knowledge base to refine and finally approve it.

Conclusion and Future Work

- Work is underway to complete approving many of the heuristics.
- The I-SHARE team continues to collect, review and publish new heuristics and expands the I-SHARE, while piloting the initial database for test uses.
- Collaboration opportunities with INCOSE working groups leverage their specialties and expertise in respective technical domains.
- A topic of future research is the use of model-based systems engineering to help organize the I-SHARE heuristics in a comprehensive OPM model, a small example of which was presented in this work.
- The model shall link various related heuristics, providing for sematic search in the model.

Thanks for Listening!

33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

www.incose.org/symp2023 #INCOSEIS