MTR200040

MITRE TECHNICAL REPORT

MITRE

Cognitive Engineering Toolkit

Cindy Dominguez
Craig Bonaceto
Chris Johnson
Michelle Bryant
Mo Rahman
Erika Darling

March 2020

Sponsor: DARPA/TTO

Contract No.: W56KGU-15-C-0010
Project No.: 0720D070-HA

Distribution Statement A. Approved for Public Release, Distribution Unlimited

Approved for Public Release; Distribution Unlimited. Public Release Case Number 20-2210

Approved By

24 March 2020

Mike Kometer, Project Lead
MITRE DARPA Projects

Date

Abstract

The Defense Advanced Research Projects Agency Hallmark program has applied cognitive engineering with MITRE's support since 2015 to improve Space Enterprise Command and Control decision support. Cognitive engineering methods can be used in DARPA and other DoD programs as a means for ensuring that new technology has mission value. This end-of-program report characterizes the 'why' and 'how' of cognitive engineering, with examples from the Hallmark program. Structured as a 4-part cycle of gather, analyze, design, and evaluate, the cognitive engineering toolkit methods and lessons-learned outlined in this report provide guidance on how to ensure technology innovation effectively supports human cognitive work in high consequence mission environments.

Executive Summary

The Defense Advanced Research Projects Agency (DARPA) Hallmark program has moved through three unique phases toward providing breakthrough capabilities for Space Enterprise Command and Control (SEC2). Hallmark has fostered the development and transition of two performer testbeds, each hosting an array of integrated tools that were evaluated by SEC2 operational teams using future-relevant scenarios in rapid, agile cycles. A unique component of Hallmark has been its incorporation of a cognitive engineering approach that has emphasized understanding, designing for, and explicitly measuring the human cognitive aspects involved in SEC2.This report focuses on the cognitive engineering component of Hallmark, which has been integral to the program since its beginning to advance the state of the art in decision support.

Cognitive engineering methods and processes grew out of major sociotechnical system catastrophes of the late 1980s, and they have enjoyed systematic progress as applied researchers have gotten outside of the lab research environment and moved into the real, messy world to study high consequence decision making in context. More than thirty years of progress in the field has been captured in numerous books and in the *Journal of Cognitive Engineering and Decision Making*. However, even with high interest and investment in new technologies to improve the timeliness and quality of human decision making, cognitive engineering methods have not yet been systematically adopted and employed across Department of Defense (DoD) programs.

Thus, this toolkit's goal is to promote the systematic adoption of cognitive engineering methods within systems engineering processes, from agile to traditional, by providing an accessible collection of cognitive engineering methods and processes along with a use case describing employment of the methods in the Hallmark program. By doing so, we aim to improve the way that technology development projects and programs support human thinking and therefore improve mission outcomes.

When are cognitive engineering methods appropriate? Programs that will benefit from cognitive engineering support are those which involve:

1. Multiple capabilities, technologies, and people working together towards a mission with high-stakes consequences, under conditions of time pressure, high stakes, shifting goals, and ambiguity [5, 6]
2. People who must make decisions, assess situations, plan, understand, adapt and replan, detect problems, and coordinate [4, 4]
3. Organizational, cultural, and economic constraints that tend to push operations towards the boundaries of safe performance [7]

Cognitive engineering process. As the Hallmark program and other cognitive engineering work has evolved recently, we have sought to frame and communicate an integrative model of cognitive engineering that is aligned with systems engineering needs. The model is shown in Figure 1 below. It provides the structure for presenting cognitive engineering methods in this toolkit.

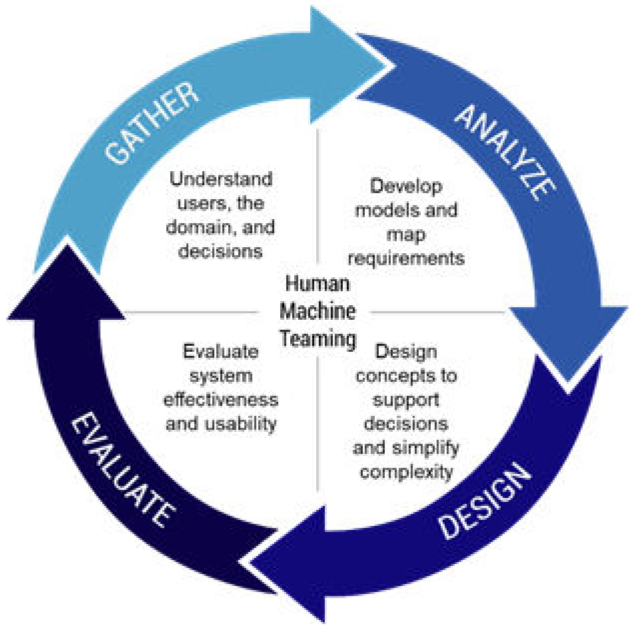


Figure 1. Cognitive engineering lifecycle.

Toolkit methods. The set of cognitive engineering methods presented in Section 2 have been selected to cut across multiple frameworks that have evolved over the past decades. The methods are summarized in Figure 2 below. We have presented each method in a consistent format that provides an introduction, timeframe required for use, when the method should be used, steps to take, tips learned from applying the method, companion methods, and resources and references documenting the method in greater detail.

Gather	Analyze	Design	Evaluate
Task Diagram Interview	Thematic Analysis of Qualitative	Storyboarding	Cognitive Wall Walk
Knowledge Audit	Decision Requirements Table	Participatory Design Events	Heuristic Evaluation
Simulation Interview	Goal Directed Task Analysis	Cognitive Engineering Design Principles	HMT Heuristic Evaluation
HMT Knowledge Audit	Cognitive Work Analysis		Cognitive Performance Indicators Evaluation
Critical Decision Method	Writing Cognitive Work and HMT Requirements and User Stories		Human-in-the-Loop Simulations
Workflow Analysis Interview			Formal Usability Testing
Wagon Wheel Interview			

Figure 2. Cognitive Engineering Toolkit methods

At the end of Section 2, we combine gather-analyze-design-evaluate methods together into "toolchains" that are appropriate for use in short, medium, and long-term projects. The example shown in Figure 3 below is a toolchain for medium-term projects lasting between six months to one year. For a project of this duration aimed at designing a prototype for a new decision support system, a cognitive engineering team might (1) gather data using the Critical Decision Method or Applied Cognitive Task Analysis (ACTA) tools; (2) analyze data using thematic analysis and create a Decision Requirements Table; (3) design using Participatory Design Events to jumpstart ideation for Storyboarding; and (4) evaluate mockup designs using Cognitive Wall Walk and Heuristic Evaluation methods.

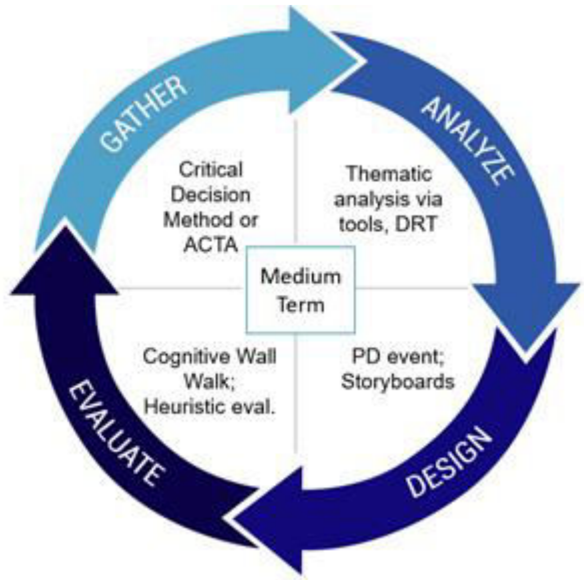


Figure 3. Medium-term Cognitive engineering toolchain, for 6-12 month projects.

Each project and domain has unique goals, timelines, resources, and needs that need to be considered when constructing a cognitive engineering toolchain. These toolchains are provided to inspire cognitive engineering practitioners and project teams to combine methods across the design and development cycle.

Hallmark Exemplar. Section 3 provides an in-depth description of how cognitive engineering was applied in the Hallmark program. It includes a summary of Phases 0, 1, and 2 and describes a trajectory of how MITRE and other team members employed cognitive engineering methods across these phases. Although cognitive evaluation had been the primary cognitive engineering approach used in Hallmark, the program also included visits to several Command and Control facilities, a range of unique analysis activities, and three separate participatory design events in each phase. Key lessons-learned from this long-term and systematic application of cognitive engineering methods across the cycle included:

1. Constructing a baseline operational workflow model early in the program would have helped to inform source selection, performers' understanding of SEC2, training, and tool integration. We share an exemplar and reference of such a model in the cyber domain.

2. Establishing a foundation of system stability and usability, by dedicating early sprints to resolving system and usability features, may have helped avert later issues. The challenging chicken-or-egg conundrum is that early demonstration/evaluation events did not have the later data feeds established, nor scope of tool performance range, that eventually contributed to stability and usability issues downstream. Even so, establishing common look-and-feel and funding the visualization tool provider to provide direction to the other 10-plus tools earlier in Phase 1 may have helped.

3. Enabling knowledge sharing across competitors would improve speed and iteration on evaluation findings. Competition in phase 1 hindered the two different cognitive evaluation providers from sharing their results and findings openly across testbeds and with each other.

4. Ensuring evaluation findings are structured and focused to rapidly inform the next sprint would improve iteration and prevent rehashing known issues during valuable feedback time. We present techniques used and hindsight-based lessons learned regarding how to enable rapid

turnaround of cognitive evaluation analysis into performers' design cycles. Notably, allowing the tool providers to directly participate in evaluation-week After-Action Reviews allowed software developers to gain firsthand feedback was a helpful practice, and focusing cognitive evaluation analysis efforts towards cataloging user stories by tool was appreciated by performers.

Summary. This cognitive engineering toolkit has documented the appropriate use, benefits, methods, and a detailed exemplar showing how a cycle of gather, analyze, design, and evaluate can be applied across the systems engineering lifecycle.

Although each project or program whose goal it is to support complex cognitive work must tailor the methods and how they are applied to their own unique circumstance, we argue that the lack of application of cognitive engineering has resulted in misspent resources and effort as well as user rejection of technology solutions. The systematic application of user-facing gather, analyze, design, and evaluation methods, as appropriate, can result in connecting the mission-driven technology needs of users tightly with the systems that are developed, as validated by evaluation results of each application. This toolkit is intended to be a living document, and we welcome additional feedback to improve and revise the toolkit as we continue to evolve the practice of cognitive engineering.

This page intentionally left blank.

Table of Contents

List of Figures

List of Tables

This page intentionally left blank.

1 Introduction

[Cognitive engineering] is an approach to the design of technology, training, and processes intended to manage cognitive complexity in sociotechnical systems [3].

In Space Enterprise Command and Control (SEC2), as in all operational-level C2 domains, our nation's ultimate goal is to achieve decision superiority. Yet SEC2 is experiencing the same conundrums as other modern C2 domains, including an ever-increasing demand for critical data and analytic capability for supporting decision making. Moreover, the proliferation of disconnected technologies obstructs users as they try to make sense of changing situations, assess risks, and make decisions under uncertainty.

Thus, users and technologists alike are turning to automation, autonomy, and artificial intelligence (AI), including machine learning, to provide decision support. Such decision support must be able to adapt to dynamic contexts, keep pace with the rapid flow of information and activity, and support complex cognitive processes. Decision support that is able to do this, i.e., that is synchronized with the user's decision process and the demands affecting it, is human-centered.

However, challenges abound in ensuring decision support capabilities are implemented in a human-centered way. The pitfalls and failures of software systems conceived with the best of intentions but not informed by operational needs and context are well-documented [1, 2]. One way we can meet these challenges is by applying cognitive engineering methods. These methods apply cognitive psychology principles to the design and development of systems to support the cognitive work of users in time-critical, complex, and high stakes environments such as SEC2.

The cognitive engineering toolkit described in this document is intended to make cognitive engineering methods more accessible to system development professionals. Its primary uses are three-fold:

1. To provide guidance to Defense Advanced Research Projects Agency (DARPA) program managers on incorporating cognitive engineering into their programs. DARPA's Hallmark program is presented as an illustrative case study.
2. To provide guidance to systems engineers on incorporating cognitive engineering across the systems engineering lifecycle.
3. To serve as a resource for human factors and cognitive engineering practitioners on the use of a variety of cognitive engineering methods so they can expand their personal toolkit.

1.1 Characteristics of Programs that Need Cognitive Engineering Support

Cognitive engineering (sometimes called cognitive systems engineering, or CSE) methods are high-impact additions to the research and development of sociotechnical systems for cognitively complex work [3]. This definition is information dense and should be unpacked. First, cognitive engineering methods address the design of technology, training, and processes intended to help people function effectively and accomplish missions with these systems.

The goal of cognitive engineering is to develop systems that are easy to learn, are easy to use, and result in improved human-computer system performance. *[8]*

Second, cognitive complexity refers to activities such as identifying, judging, attending, perceiving, remembering, reasoning, deciding, problem-solving, and planning [4]. Third, sociotechnical systems refer to systems in which multiple capabilities, technologies, and people work together towards a mission with high-stakes consequences, under conditions of time pressure, shifting goals, and ambiguity [5].

In short, when systems have any of the following characteristics, using cognitive engineering methods as early as possible in development will help to improve user acceptance and shorten training timelines:

1. Multiple capabilities, technologies, and people work together towards a mission with high-stakes consequences, under conditions of time pressure, high stakes, shifting goals, and ambiguity [5, 6]
2. People must make decisions, assess situations, plan, ,understand, adapt and re-plan, detect problems, and coordinate [4]
3. There are organizational, cultural, and economic constraints that tend to push operations towards the boundaries of safe performance [7]

SEC2, which enables operational level C2 across the enterprise of military space systems and coordinating government and commercial space systems, is one example of a complex sociotechnical system that requires cognitive engineering to ensure operator acceptance of innovative products. Section 3 describes how cognitive engineering was used to inform the design and development of this system.

1.2 How to Use this Toolkit

This toolkit is targeted to three main audiences: DARPA Program Managers, Systems Engineers, and Human Factors Engineers.

- **DARPA Program Managers:** DARPA Program Managers might benefit most from reviewing Section 3 to assess their program's need for cognitive engineering. Team members should then use the toolkit section (Section 2) to develop and tailor a cognitive engineering support plan.
- **Systems Engineers:** Systems Engineers might begin with Section 3 to assess their systems engineering effort and its need for cognitive engineering support, and then use Section 2 to develop and tailor a cognitive engineering support plan consistent with goals and timelines of the program.
- **Human Factors Engineers or Cognitive Engineering Practitioners:** This reader would benefit from the examples in Sections 3,but might focus on Section 2 to view example cognitive engineering toolchains and obtain guidance on using specific cognitive engineering methods.

This document is structured as follows:

- The remainder of Section 1 provides an overview of cognitive engineering methods and phases and discusses planning cognitive engineering work programs.
- Section 2 describes a set of cognitive engineering methods to address each phase and provides example cognitive engineering toolchains.
- Section 3 describes incorporating cognitive engineering in DARPA programs and presents an illustrative case study on using cognitive engineering in the DARPA Hallmark program.

1.3 Cognitive Engineering Overview

Imagine being given the task of designing a decision support system to aid military commanders in planning troop movements. How would you determine the patterns of information most meaningful to this mission? How would you identify the key software capabilities that would be needed to solve those commanders' most pressing and frequent problems? How would you determine the format in which to display the information to facilitate effective decision making? How would you enable flexibility and transparency across team members (including automation and intelligent agents) to facilitate human-machine teaming? How would you know when you had developed a usable and effective system that yields increased performance? How would you design training materials?

These are the types of system design and evaluation questions that cognitive engineering methods can effectively address [8]. Cognitive engineering encompasses methods and approaches from the fields of human factors, human-computer interaction, cognitive psychology, computer science, artificial intelligence and other related fields. Cognitive engineering methods are useful at establishing a foundational understanding of users, the tasks they perform, task and situational demands that affect how they perform, and the decisions they make as the central drivers for system design. The development and practice of cognitive engineering emerged from high-profile lessons learned in the wake of catastrophes such as Three Mile Island and the USS Vincennes' shootdown of an Iranian commercial airliner in the late 1980s [9] [10] [11].

> ***Experience with the introduction of new technology has shown that increased computerization does not guarantee improved human-machine system performance. Poor use of technology can result in systems that are difficult to learn or use, can create additional workload for system users, or in the extreme, can result in systems that are more likely to lead to catastrophic errors.*** *[8]*

The methods that a cognitive engineer employs are driven by the methodological framework they adopt or operate within. Frameworks include Cognitive Work Analysis, Decision-Centered Design, Situation Awareness-Oriented Design, and Work-Centered Design (see [3] for a detailed description of these frameworks and associated references). At MITRE, we have sought an integrative model of cognitive engineering, aligned with systems engineering objectives, that offers guidance on how to understand users' cognitive work within complex sociotechnical systems. This guidance is framework-agnostic, and it is adapted from the five steps outlined in [12]. The model is shown in Figure 1-1 below.

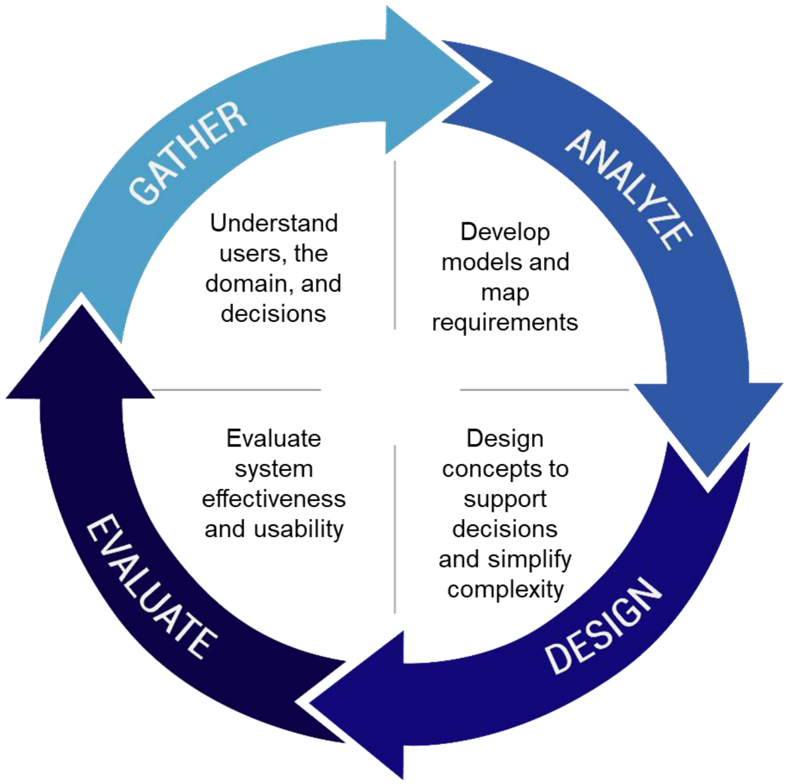


Figure 1-1. The cognitive engineering lifecycle.

The phases of cognitive engineering are gather, analyze, evaluate, and design. These phases are largely parallel to the phases in a systems engineering or software development effort. Figure 1-1 shows how cognitive engineering entails an iterative cycle of gathering data about human work in operational contexts, analyzing those data to understand and characterize cognitive work, designing capabilities to support the cognitive work, and evaluating the capabilities to measure their effectiveness. In projects executed using traditional waterfall systems engineering processes, this cycle might be completed once; in projects using agile processes, the cycle might repeat several times in an iterative fashion. Iterative cycling permits technology and users alike to improve and better adapt to working with one another. When resources do not permit iteration, the technology can still benefit from the cognitive engineering lifecycle approach. Below we provide greater detail about each phase of the cycle.

1.3.1 Gather

The goal of the gather phase is to use knowledge elicitation, observation, and other user-facing techniques to collect information about cognitively challenging tasks, critical decisions, information needs, situation awareness needs, and human-machine teaming needs. Example methods include structured interviewing techniques, observation methods, and process tracing methods (such as think-aloud problem solving).

1.3.2 Analyze

The goal of the analyze phase is to examine and structure the information collected during the gather phase. This includes the creation of cognitive task analysis models, diagrams, and knowledge representation artifacts (such as Decision Requirements Tables, Goal Directed Task Analysis models, and Cognitive Work Analysis models). When called for, requirements and user stories are also written during this phase, with traceability to the data that was collected.

1.3.3 Design

The goal of the design phase is to create systems and displays that support cognitive work and facilitate human-machine teaming. Example methods include design thinking techniques, information visualization techniques, and methods to translate requirements identified in the gather and analysis phases into design concepts. Products from this phase vary in fidelity from paper-based annotated mockups and storyboards to software prototypes and functional software capabilities.

1.3.4 Evaluate

The goal of the evaluate phase is to assess the degree to which design concepts, prototypes, and system capabilities support complex cognitive work and facilitate effective human-machine teaming within the operational context intended for the system.

1.4 Planning a Cognitive Engineering Work Program

Every research and development endeavor is unique, with its own goals, scope, timelines, budget, and priorities. Some focus more on research, while others primarily focus on development with less of a research component. If a program involves research and development of sociotechnical systems for cognitively complex work, there are a range of options for tailoring a cognitive engineering work program. Example options include:

- An extensive data collection and analysis effort across multiple existing domains to form the basis for a first-of-a-kind system, like a handheld device for interacting with an unmanned helicopter at a remote combat outpost. [13]
- A moderate data collection effort with a series visits conducted to elicit user stories and design annotated mockups to inform software engineering, such as MITRE's recent work to create a Multi-Domain Command and Control common operational picture [14].
- Employment of participatory design methods to identify operational needs and seed interface designs, with a series of scenario-based evaluation events and analyses of cognitive performance data collected, such as used in the Hallmark program for Space Enterprise Command and Control (see Section 3.3).

The combination of ways to navigate gather, analyze, design, and evaluate phases is endless; the plan needs to reflect constraints and goals of a particular effort. In the upcoming sections, we describe how cognitive engineering plans have been, and could be, constructed to address specific systems engineering phases and activities, and then finally show how these methods can be combined in toolchains that support varied project time constraints.

2 Cognitive Engineering Toolkit

Sections 2.1– 2.4 below describe a curated collection of core cognitive engineering methods that address each cognitive engineering phase, including: (1) gather information about users and the domain, (2) analyze and represent data collected, (3) design systems and displays, and (4) evaluate systems and displays. Most method descriptions include information about:

- **Timeframe:** The time generally required to conduct the method (e.g., the length of time to conduct an interview or the length of time an analysis requires).
- **Use When:** Guidance on when to use the method. For example, when in the systems engineering process, or under what circumstances it is appropriate to use the method (e.g., if access to expert users is available, a legacy system exists, etc.).
- **Steps:** The high-level steps to perform the method.
- **Tips:** Practical guidance and tips on using the method.
- **Companion Methods:** Any related methods often used in conjunction (e.g., a gather method may link to a suggested analyze method that is appropriate for analyzing and structuring the data produced by the method). Section 2.5 describes related sets of methods that form "cognitive engineering toolchains".
- **Resources and References:** A list of literature citations and resources for conducting the method.

The methods described in the toolkit are depicted in Figure 2-1 below:

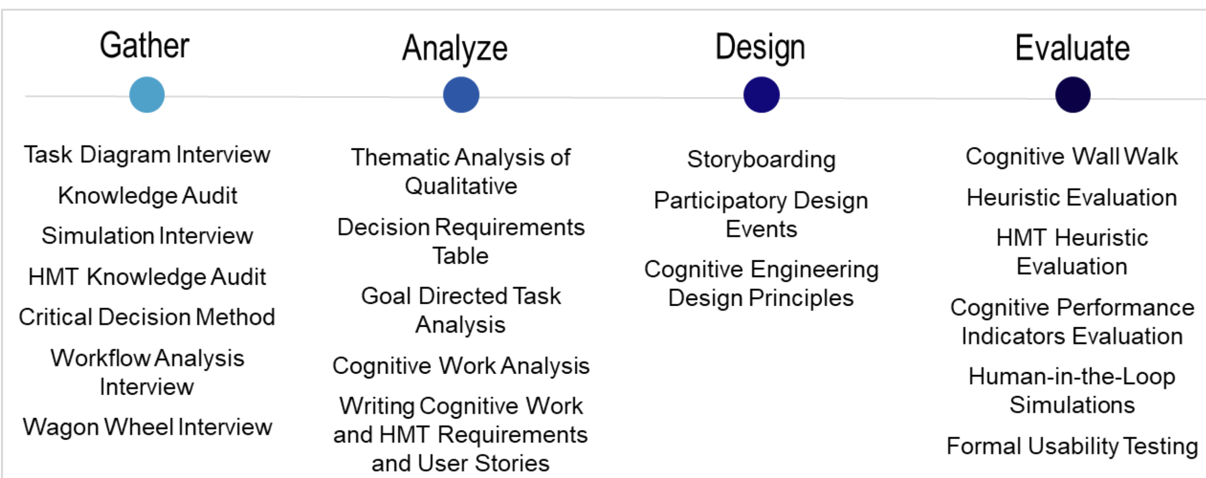


Figure 2-1. Cognitive engineering toolkit methods.

Section 2.5 describes assembling related sets of methods into "cognitive engineering toolchains" given the goals and timeframe of a particular project. A toolchain is a set of related methods where outputs from a prior method are inputs to a subsequent method.

2.1 Gather Methods

Gather methods are used to collect information about cognitively challenging tasks, critical decisions, information needs, situation awareness needs, and human-machine teaming needs. They are used to build a foundational understanding of users and the domain to inform subsequent analysis, design, and evaluation activities. Some gather methods include a built-in analysis and representation component, while others require the selection of additional methods to analyze and represent data gleaned from the method.

2.1.1 Applied Cognitive Task Analysis

In 1998 Militello and Hutton [15] published a seminal Applied Cognitive Task Analysis (ACTA) mini-toolkit consisting of three methods: the Task Diagram Interview, the Knowledge Audit, and the Simulation Interview. These methods can be used separately or mixed and matched with other methods. We have found the Task Diagram Interview to be especially useful in the opening stages of most cognitive engineering efforts, since it elicits what tasks people are responsible for, and how those tasks relate to each other. Each method is described below.

2.1.1.1 Task Diagram Interview

The Task Diagram Interview is used to obtain an overview of the activities of particular task or job and to identify the activities that are most cognitively challenging. It allows the analyst to focus subsequent interview and analysis efforts on the most relevant and challenging aspects of a task or job and to do so in a way that maps to the way domain experts approach and think about their work.

Timeframe: 30 - 60 minutes to conduct the interview, depending on the extent and complexity of the task. If a given task can be performed multiple ways, conduct interviews with three to four domain practitioners in order to represent that variability.

Use When: Domain practitioners are available (experts are not required the achieve the goals of this method), a superficial view of a task or work domain is sought due to time or other resource limitations, or when a deep, detailed understanding of a task or work domain is sought and guidance is needed for structuring and scoping the analysis.

Steps:

- Ask the participant to think through how he or she typically performs the task at hand and to identify the task's primary activities or steps.
- Capture the activities in a flow chart or other type of diagram.
- Ask the participant which of the activities in the diagram require difficult cognitive skills, explaining, e.g., "By cognitive skills I mean judging, assessing, solving problems—thinking skills."
- Circle the task activities identified as requiring difficult cognitive skills.
- For each circled task activity, ask the participant to decompose it into three to six subtasks.

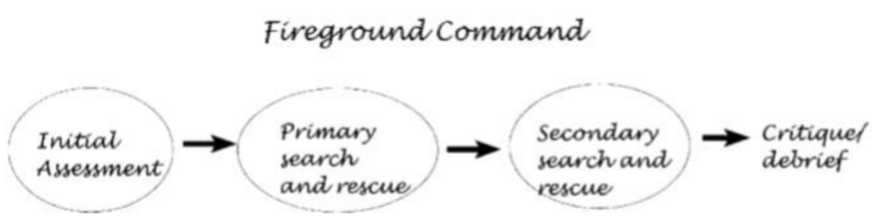


Figure 2-2. Example task diagram (adopted from [15]).

Tips:

- Limit the participant to between three and six steps to keep the assessment focused on the most important steps involved in the task.
- Participants may find it difficult to think about and characterize the cognitive activities of their work. Help them to think about the cognitive side of their work by showing them one or more example task diagrams from other domains.
- Allow the participant to explain activities and steps in the way he or she thinks about and approaches them, and not according to a predefined framework or set of categories.
- Capture and represent the task as described. If the participant wants to describe the work as entirely concurrent, rather than linear, represent the work in that way.
- If the work is classified, ask participants to remain at the unclassified level. In most cases, classified details are not necessary for achieving the goals of this method.

Companion Methods: This method is often used in preparation for a more focused interview or data collection effort. The choice of methods for that follow-on effort should depend at least partly on the nature of the cognitive challenges revealed by the Task Diagram Interview. In the ACTA methodology, the Task Diagram Interview precedes the Knowledge Audit and Simulation Interview.

Resources and References

- [15] L. G. Militello and R. J. B. Hutton, "Applied Cognitive Task Analysis (ACTA): A practitioner's toolkit for understanding cognitive task demands," *Ergonomics, Special Issue: Task Analysis,* vol. 41, no. 11, pp. 1618-1641, 1998.
- [16] L. G. Militello, R. J. B. Hutton, R. M. Pliske, B. J. Knight and G. Klein, "Applied Cognitive Task Analysis (ACTA) Methodology," Klein Associates, Inc., Fairborn, OH, 1997.

2.1.1.2 Knowledge Audit

The Knowledge Audit is designed to elicit and characterize expertise associated with performing cognitively challenging aspects of work. It focuses on the details of cognitive activities used for: Diagnosing and predicting, situation awareness, perceptual skills, developing and knowing when to apply tricks of the trade, improvising, meta-cognition, recognizing anomalies, and compensating for equipment limitations. These cognitive activities are, according to research on expert-novice differences, primary sources of differences in expert versus non-expert task performance.

Timeframe: 60 - 120 minutes per interview depending on the extent and complexity of the task; conduct interviews with at least three to four domain experts in order to obtain a good representation of the ways experts have adapted their cognitive skills to perform effectively across various domain challenges

Use When: Domain experts are available (nonexperts may be interviewed as well in order to reveal expert-nonexpert differences); the goal is to understand cognitive work employed by expert task performers (e.g., for training, assessment, or performance support purposes).

Steps:

- For each task of interest (potentially determined using the Task Diagram method), the analyst asks 7 - 9 probe questions, as follows:

- **Perceptual Skills:** Experts detect cues and patterns and make discriminations that novices typically can't see. Can you think of any examples where that might be true for this task/activity?
- **Recognizing Anomalies:** Experts often notice when something unusual happens. They can quickly detect deviations. They also notice when something that should happen doesn't. Can you think of a time where that was true for you or another expert you know while conducting this task/activity and describe what happened?
- **Past and Future:** Experts can guess how the current situation arose and they can anticipate how the current situation will evolve. Can you think of an instance where you or an expert you know was able to successfully anticipate an evolving situation? Alternatively, can you think of a time when novices were unable to anticipate the evolution of an event?
- **Big Picture:** If you were watching novices, how would you know that they don't have the big picture?
- **Job Smarts:** Are there tricks of the trade that you use? How do these tricks help you complete the task? Are there reasons novices shouldn't use tricks in building their expertise?
- **Improvising or Noticing Opportunities:** Can you recall a situation when you noticed that following the standard procedure wouldn't work? What did you do? Can you think of an example where the procedure would have worked but you saw that you could get more from the situation by taking a different action?

  Self-Monitoring and Adjustment: Experts notice when their performance is sub-par and can often figure out WHY that is happening (e.g.. high workload, fatigue, boredom, distraction) in order to make adjustments. Can you think of any examples where you did this?

Optional Probes

- **Equipment:** Unless you're careful, the equipment can mislead you. Novices usually believe whatever the equipment says. Can you think of examples where you had to rely on experience to avoid being fooled by the equipment?
- **Scenario from Hell:** If you were going to give someone a scenario to teach someone humility—that this is a tough job—what would you put into that scenario? Did you ever have an experience that taught you humility in performing this job?

- For each probe question, the participant provides an example of a situation they have experienced on the job.
- The analyst creates a Knowledge Audit Table and, for each probe question and corresponding situation:
  - Records the example in Column 1 of a Knowledge Audit Table.
  - Asks and records answers in Column 2 : "Why is this task hard for novices or why don't novices know to do that?" or "What errors less-experienced practitioners might tend to make in the situation and why?"

- Asks about and records in Column 3 specific critical cues and decision making strategies ("What cues or strategies do you use in this situation?").

Task of Interest		
Example	Why Difficult	Cues & Strategies
1. **Perceptual Skills:** *Example of perceptual skill* 2. **Anomalies**: *Example of anomaly* 3. **Past & Future:** *Example…*		

Figure 2-3. Knowledge Audit table template.

Tips:

- To optimize efficiency, keep the interview focused on the probe questions and associated areas of expertise.
- As the participant speaks, listen for ambiguous statements, concepts that are unfamiliar, activities that are being glossed over, and other interview content and encourage elaboration.
- Conduct interviews in pairs to improve the likelihood that at least one interviewer will recognize when a statement is ambiguous, a concept unfamiliar, or an activity glossed over.
- Capture as much interview content as possible in the Knowledge Audit table to minimize time spent replaying the interview audio recording (if used).
- Participants may find it difficult to think about and characterize the cognitive activities of their work. Help them to think about the cognitive side of their work by showing them one or more example Knowledge Audit tables for other work domains.

Companion Methods: In the ACTA methodology, the Knowledge Audit follows the Task Diagram Interview and precedes the Simulation Interview. The Task Diagram Interview helps the analyst focus the Knowledge Audit on the cognitively challenging tasks of a work domain. The Simulation Interview adds to and enriches the cognitive-work details elicited by the Knowledge Audit by focusing on judgment and decision making. Many other methods may complement the Knowledge Audit; the choice of methods should depend on the analyst's objectives, resources, and the nature of the cognitive work being investigated.

Resources and References

- [15] L. G. Militello and R. J. B. Hutton, "Applied Cognitive Task Analysis (ACTA): A practitioner's toolkit for understanding cognitive task demands," *Ergonomics, Special Issue: Task Analysis,* vol. 41, no. 11, pp. 1618-1641, 1998.

- [16] L. G. Militello, R. J. B. Hutton, R. M. Pliske, B. J. Knight and G. Klein, "Applied Cognitive Task Analysis (ACTA) Methodology," Klein Associates, Inc., Fairborn, OH, 1997.

2.1.1.3 Simulation Interview

The Simulation Interview is designed to elicit details about the cognitive processes used in making decisions and judgments. It uses a scenario to provide a context the participant can use to ground task or activity descriptions. The participant is asked to recall major events, decisions points, and judgments that occurred during the scenario and, for each, to provide a situation assessment, recommend actions, critical cues, alternative courses of action, and potential errors.

Timeframe: 60 - 120 minutes per interview depending on the extent and complexity of the scenario and users' tasks; conduct interviews with at least three to four domain experts in order to obtain a good representation of the cognitive work involved in the task.

Use When: Domain experts are available (nonexperts may be interviewed as well in order to reveal expert-nonexpert differences); the goal is to understand cognitive work employed by expert task performers (e.g., for training, assessment, or performance support purposes). It is especially helpful when your research benefits from comparing a range of responses to a scenario. The scenario can act as an independent variable, or stimulus, in the research.

Steps:

- Present the participant with a typical scenario via video, paper, computer, etc. (medium does not matter). You might read each scenario segment aloud while sharing a written document for the participant to refer to, including a map or diagram if useful.
- Ask the participant to keep in mind that they will be asked about the decisions and judgments they would have made in this situation. Offer the participant pencil and paper on which to keep notes.
- Divide a whiteboard into 6 columns with headings that match the template below.

Scenario:					
Events/ Decisions/ Judgments	Situation Assessment	Actions	Critical Cues	Alternatives	Potential Errors
Event #1 ***Event #2*** ***Event #3***					

Figure 2-4. Simulation Interview table template.

- After the participant has heard/reviewed the first simulation segment, ask: "Think back over the scenario. Please list the major events/judgments/decision points that occurred during the incident. As you name them, I am going to list them in the left column on the board."
- For each event in the left column, ask the probe questions listed below:

  - ***Situation Assessment*:** What do you think is going on here? What is your assessment of the situation at this point in time?
  - ***Actions*:** What actions, if any, would you take at this point in time?
  - ***Critical Cues*:** What pieces of information led you to this situation assessment/action?
  - ***Alternatives*:** Are there any alternative ways you could interpret this situation? Are there any alternative courses of action that you would consider at this point?
  - ***Potential Errors*:** What errors would an inexperienced person be likely to make? Are there cues they would miss?
- Ask all five questions about a specific segment before moving on to the next segment. Record the answers to each question in the appropriate column.

Tips:

- If possible, have an expert on your research team present each scenario segment; this strategy enables the participant to ask deeper questions and helps establish credibility of your team.
- To optimize efficiency, keep the participant focused on the probe questions.
- As the participant speaks, listen for ambiguous statements, concepts that are unfamiliar, activities that are being glossed over, and other interview content and ask them to elaborate.
- Conduct interviews in pairs to improve the likelihood that at least one interviewer will recognize when a statement is ambiguous, a concept unfamiliar, or an activity glossed over.
- Capture as much interview content as possible in the Simulation Interview table to minimize time spent replaying the interview audio recording (if used).
- Participants may find it difficult to think about and characterize the cognitive activities of their work. Help them to think about the cognitive side of their work by showing them one or more example Simulation Interview tables for other work domains.

Companion Methods: In the ACTA methodology, the Simulation Interview follows the Task Diagram Interview and Knowledge Audit. The Simulation Interview can also be used on its own. It is used to gain additional task performance data focused on decision making and judgments. Many other methods may complement the Simulation Interview; the choice of methods should depend on the analyst’s objectives, resources, and the nature of the cognitive work being investigated.

Resources and References

- [15] L. G. Militello and R. J. B. Hutton, "Applied Cognitive Task Analysis (ACTA): A practitioner’s toolkit for understanding cognitive task demands," *Ergonomics, Special Issue: Task Analysis,* vol. 41, no. 11, pp. 1618-1641, 1998.
- [16] L. G. Militello, R. J. B. Hutton, R. M. Pliske, B. J. Knight and G. Klein, "Applied Cognitive Task Analysis (ACTA) Methodology," Klein Associates, Inc., Fairborn, OH, 1997.

2.1.2 Human-Machine Teaming Knowledge Audit

The Human-Machine Teaming (HMT) Knowledge Audit is an adaptation of the Militello and Hutton Knowledge Audit technique that addresses the challenges of designing autonomous and automated systems to support HMT. Such systems support adaptive, bi-directional team interaction among humans and machines in a way that augments human capabilities for improved mission outcomes. The HMT Knowledge Audit is the core component of an HMT interview session, which also includes introductory probes shown in Figure 2-5 below to paint a picture of envisioned autonomy, identify top challenges, and identify the current state of automation/autonomy. During an HMT Knowledge Audit, participants are asked a series of probe questions drawn from HMT themes, such as Observability, Directability, and Adaptability; probe question categories are also aligned with aspects of expertise in a standard Knowledge Audit. Interview results are used to derive HMT requirements.

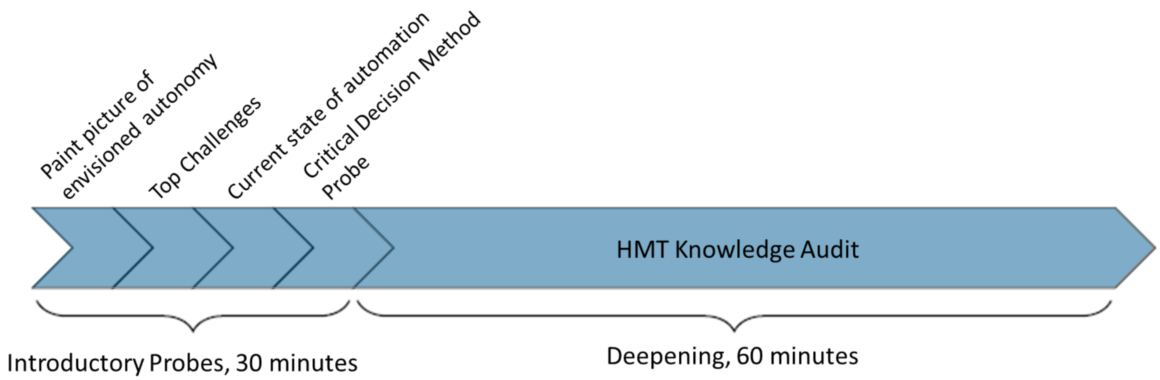


Figure 2-5. Human-machine teaming interview overview.

Timeframe: 90 – 120 minutes to conduct a full HMT Interview; an abbreviated version that includes only the HMT Knowledge Audit may be conducted in 60 minutes. The method may be conducted with both expert and novice domain practitioners, either one-on-one or in small groups.

Use When: Designing systems that will incorporate autonomy, automation, or Artificial Intelligence (AI).

Steps:

- **Paint a Picture of Envisioned Autonomy**: Describe the envisioned system in sufficient detail to allow the participant to understand the concept and imagine how they could use it. This background helps the participant answer subsequent questions about envisioned autonomy.
- **Top Challenges:** Ask about the top 3 – 5 tasks the participant is responsible for and which are the most cognitively challenging. The goal of this step is to understand responsibilities of the job to tailor subsequent interview questions.
- **Current State of Autonomation/Autonomy:** Ask about the current state of automation and how the participant uses automation to accomplish their work. Also ask how current automation falls short and in what ways it is unreliable or challenging.
- **Critical Decision Method Probe (Optional):** Ask about a particularly challenging situation (see CDM method description, Section 2.1.3). The goal of this probe is to quickly elicit examples where the participant used their expertise to deal with a challenging incident, without subsequent sweeps. Such examples reveal challenges of the work and how an autonomous partner could be beneficial.
- **HMT Knowledge Audit:** Ask a series of probe questions that address HMT themes, including Predictability, Exploring the Solution Space, Observability, Calibrated Trust, Directing Attention,

Adaptability, Information Presentation, and Common Ground. Probe questions are provided in [17].

Tips:

- If interview time is constrained, you can eliminate or truncate the current state of autonomy and Critical Decision Method probe portions of the interview.
- The questions in the HMT Knowledge Audit can be asked in any order. If interview time is constrained, you can also prioritize the HMT themes to focus on those that are most relevant. Relevance could be determined during pilot interviews or through discussions with the system developers and program managers.

Companion Methods: The HMT Knowledge Audit is part of a suite of methods to address designing for HMT, including an HMT Heuristic Evaluation and a set of generic, tailorable HMT requirements.

Resources and References:

- [17] P. McDermott, C. Dominguez, N. Kasdaglis, M. Ryan, I. Trahana and A. Nelson, "Human-Machine Teaming Systems Engineering Guide," The MITRE Corporation, Bedford, MA, 2018.

2.1.3 Critical Decision Method

The Critical Decision Method (CDM) is a retrospective interviewing technique for capturing expertise and identifying decision requirements in the context of a challenging incident. It is particularly useful in understanding the key challenges users face and their associated decision-making needs The CDM leverages the fact that domain experts often retain detailed memories of previous cases, especially ones that were unusual, challenging, or somehow involved "critical decisions." A CDM interview involves selecting an incident that the interviewee has personally experienced, and then conducting multiple "sweeps" through the incident with the participant. The sweeps include: (1) Incident Selection and Recall, (2) Timeline Verification and Decision Point Identification, (3) Deepening (where a series of probe questions are asked to better understand the decision points), and (4) "What-if" Queries.

Timeframe: 90 - 120+ minutes to conduct a CDM interview

Use When: Domain experts are available and actual events are infrequent or difficult to observe. The method is very fruitful in identifying "leverage points" – areas where changes in technology or process may have large gains in improving user effectiveness.

Steps:

- **Incident Selection and Recall (Sweep 1)**: Select an incident that meets your goals where the participant played a key role. The incident is typically non-routine, involved the participant in a key decision-making role, and challenged the participant's skills. Once a good incident has been selected, ask the participant to provide a brief account from beginning to end.
- **Timeline Verification and Decision Point Identification (Sweep 2):** Walk through the incident a second time and build a timeline of each critical point (decision point). These are points where understanding shifted, a decision was made, or a key action was taken.
- **Deepening (Sweep 3):** Using the timeline, walk through each segment of the story a third time and use a series of deepening probe questions to elicit additional detail about each critical event/decision point in the incident. For example, probes may elicit details about sources of

information the participant considered, cues they relied upon, or what their goals and priorities were.

- **What-if Queries (Sweep 4):** Pose various hypothetical challenges about the overall incident or at critical points and ask the participant to speculate on how things might have unfolded differently using what if probe questions – for example, "what if you experienced a loss of communication to other team members at this point?" The goal is to determine how key parts of the incident might vary under different circumstances, or to understand options that were considered and rejected.

Tips:

- Sometimes an incident doesn't lend itself to a detailed timeline (e.g., if events are non-linear). Instead, consider just capturing the overall sequence of events, or using a geospatial map to capture critical events.
- Schedule a break after Sweep 2. During the break, clean up the timeline and determine relevant probe questions to use in Sweep 3.
- A CDM interview can be broken out into multiple sessions. In the first session, conduct Sweeps 1 and 2. In the second session, conduct Sweeps 3 and 4.

Companion Methods: A Decision Requirements Table can be used to capture and consolidate interview data.

Resources and References:

- [18] G. Klein, R. Calderwood and D. MacGregor, "Critical decision method of eliciting knowledge," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 19, pp. 462-472, 1998.
- [12] B. Crandall, G. Klein and R. R. Hoffman, Working Minds: A Practitioner's Guide to Cognitive Task Analysis, Cambridge, MA: MIT Press, 2006.
- [19] R. R. Hoffman, B. Crandall, G. Klein, D. G. Jones and M. R. Endsley, "Protocols for Cognitive Task Analysis," Institute for Human and Machine Cognition, Pensacola, FL, 2008.

2.1.4 Workflow Analysis Interview

During a workflow analysis interview, the participant guides the interviewer through the overall workflow they perform to accomplish their goals. As the participant steps through the workflow, the analyst captures information about the tools they use, the information they access, the decisions they make, areas for collaboration, and any pain points they experience. This method is modeled after the Critical Decision Method where the participant guides the interviewer through the workflow in multiple "sweeps".

Task:

Overview

- What is this task all about? What are the goals, purposes, or functions? Why does this have to be done?
- Who usually is responsible for doing this? How is that decided?
- How much time or effort is generally involved in this task?
- When do you accomplish this task in relation to stepping to the jet?
    - What must be accomplished before this task is done?
    - What can be done at the same time this task is being performed?
    - Does this task ever have to be redone or refined if things change?
- Where do you this in training vs. in theater?

Action Sequences

- Let's walk through a typical action sequence to perform this task at a high level. List each step.
- Let's walk through a challenging case (perhaps where there was time pressure or where issues were encountered). List each step.
- What steps are essential and which are likely to be skipped under conditions of time pressure?

We will next sweep through the action sequence in more detail while addressing each of the sections below.

Decisions and Information Needs

- What information or data is needed to perform this task, and where is the data obtained from? List each piece of data or information, the source, and what it's needed for. If you use a checklist to help perform the task, please describe the checklist.
- What are the decisions that must be made?

Collaboration and Coordination Needs

- What type of collaboration or coordination (with other people in the unit or with other organizations) is needed to perform this task?

Tools Used

- What tools do you use to accomplish this task and how do you use them? List each tool and describe what the tool is used for, what is good and useful about the tool, and what problems there are with the tool.

Pain Points and Highlights

- What are the pain points that make this task difficult? For example, are there workarounds to deal with deficiencies in the tools or processes?
- What aspects of the current system are particularly useful in your planning process?
- What have you learned over the years that has made your mission planning process easier or more efficient?

Leverage Points

- Are there any additional tools, features, or aids that might be useful? Are there any process improvements that would help?
- Are there mission planning products that you create outside of JMPS? What are they and which tools do you use?

Figure 2-6. Workflow Analysis Interview template example.

Timeframe: 60 – 120 minutes to conduct the interview. This will vary depending upon the complexity of the workflow. Interviews may be conducted one-on-one or in small groups. Multiple rounds of interviews will enable progressively expanding on workflow sections.

Use When: It is necessary to obtain a broad overview of a set of workflows in the domain.

Steps:

- **Workflow overview:** Ask a series of framing questions to obtain general information about the workflow, including the overall goal of the workflow, the time or effort involved, and who is responsible for accomplishing the workflow (the team composition). Answers to the framing questions may be used to tailor the probe questions that are asked during subsequent sweeps through the workflow.

- **Typical scenario action sequence elicitation:** Ask the participant to describe the high-level sequence of steps that comprise the workflow.
- **Challenging scenario action sequence elicitation:** If time permits, ask the participant to walk through the sequence of steps in a challenging case (e.g., where there was time pressure, issues encountered, or other challenges that made the case difficult or nonroutine).
- **Workflow walkthrough (deepening):** Step through each step in the workflow in more detail and ask a series of probe questions at each step. Probe questions address the decisions made, the information required, any collaboration points, the tools used, pain points, and leverage points.

Tips:

- Ideally, conduct a first round of interviews to establish a starting workflow that is validated and fleshed out in greater detail with subsequent interviews, so that each interview does not need to start from a blank slate.
- If possible, guide the participant through the workflow while they have access to their current software. They will likely be able to recall additional details when in their actual work environment with access to their current tools.
- The mission may entail multiple distinct workflows. Conduct multiple interviews for each major workflow.

Companion Methods: A Task Diagram Interview can be used to establish the range of elements to include in a workflow. Goal Directed Task Analysis (GDTA) can be used to represent the supporting goals that comprise the workflow.

Resources and References:

- [18] G. Klein, R. Calderwood and D. MacGregor, "Critical decision method of eliciting knowledge," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 19, pp. 462-472, 1998.
- [20] S. Trent, R. Hoffman, D. Merritt and S. Smit, "Modelling the cognitive work of cyber protection teams," *The Cyber Defense Review,* vol. 4, no. 1, pp. 125-135, Spring 2019.

2.1.5 Wagon Wheel

The Wagon Wheel method is a cognitive task analysis technique that aims to reveal the nature of team communication. The goal of the method is to identify the primary communication goals, partners, and channels for each position on a team. The method may also be used to investigate the nature of the communications, including typical patterns, means of communication, and communication obstacles. It is useful for dissecting information flows and identifying team roles and functions, information requirements, types of information passed between team members, sources of information, decision and course of action impacts, criticality of information, and the impact of poor information flow. The method can be used with both highly expert and novice participants in distributed and co-located teams, and in both one-on-one and group data collection sessions.

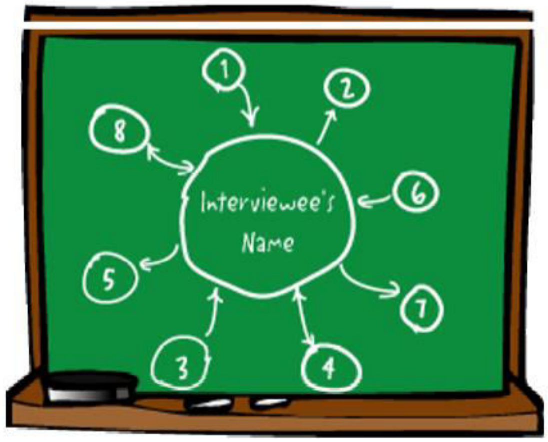


Figure 2-7. Wagon Wheel example.

Timeframe: 30+ minutes to conduct a Wagon Wheel interview with each team member. Interviews are generally conducted one-on-one, but may be conducted in a small group with multiple team members who perform the same or similar role.

Use When: Designing to support tasks that are distributed among team members, or when the research's goal is to understand team coordination and collaboration. The method can be used with both expert and novice participants in distributed and co-located teams.

Steps:

- Place the participant's name in a circle in the center of a large piece of paper or whiteboard.
- Ask who the participant communicates with and place the names/roles as spokes around the central circle.
- Work with the participant to expand and annotate the diagram with:
  - Arrows to show the flow of information.
  - Text to show modes of communication.
  - Weighted lines to show the frequency of communications or importance of a link.
  - Diagram clusters of individuals or organizations who relate to each other in an important way, and show the relationship in your diagram.
- Ask probe questions to understand which links are most critical, typical content of communications, obstacles to communicating, the type of information that is passed, and the decisions information affects.

Tips:

- The Wagon Wheel Method is best applied with one participant at a time. It is necessary to have a whiteboard or some other common point of reference for the interviewer and the participant to record the communication patterns. Blank paper will work.
- If feasible, actual team communication patterns may also be analyzed as was done in Hallmark Phase 0 (see Section 3.3.2.2).

Companion Methods: Concept Maps [21] can be used to represent the team structure and communication patterns captured in a Wagon Wheel interview.

Resources and References:

- [22] D. Klinger and H. B., "Handbook of TEAM CTA," Human Systems Center, Brooks AFB, 2003.
- [23] C. Dominguez, P. Uhlig, J. Brown, O. Gurevich, W. Shumar, G. Stahl, A. Zemel and L. Zipperer, "Studying and supporting collaborative care processes," in Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Santa Monica, CA, 2005.
- [24] R. J. Harder and H. Higley, "Application of thinklets to team cognitive task analysis," in Proceedings of the 37th Annual Hawaii International Conference on System Sciences, Big Island, HI, USA, 2004.
- [25] B. Moon, "Concept maps and wagon wheels: merging methods to improve the understanding of team dynamics," in Proceedings of the First International Conference on Concept Mapping, Pamplona, Spain, 2004.

2.2 Analyze Methods

Analyze methods are used to analyze, structure, and represent the information collected during the gather phase. Analysis and representation of the information collected should align with the goals and purpose of the project. For example, Decision Requirements Tables are effective at capturing critical decisions for new decision support capabilities. When called for, requirements and user stories are also written, with traceability to the data that was collected.

2.2.1 Thematic Analysis of Qualitative Data

A number of the methods in this document involve the collection of verbal data during interviews, work-focused discussions, and work activity. These data can be analyzed to identify themes, frequencies, and patterns. The process below should be tailored to project goals and resources. Some projects do not code every excerpt of content from interview or observation notes, and some do not undertake statistical inter-rater reliability assessment, relying on practice coding and review for consistency.

Timeframe: Transcribing verbal data can take 4 - 6 hours per hour of recorded data. Processing and analyzing interview data requires many days of time from more than one analyst.

Use When: Qualitative data must be analyzed and summarized.

Steps:

- If data were collected using audio or video recordings, transcribe the recordings into written form. Detailed interview notes can suffice if the interview was not recorded.
- Review the transcriptions (or interview notes/data) and break them into separate excerpts. Each transcript excerpt should be sufficiently small to be assigned one or two codes (versus three or four codes in order to represent all that is going on within the element) and sufficiently large for the speaker's point to be made and relevant context to be understood. Excerpts are typically one to three sentences but can be longer if the speaker repeats him or herself. As you identify excerpts, list them in the column of a spreadsheet. Remove extraneous information in the data file, such as introductions and places to go for lunch.
- Develop codes that will be mapped to the excerpts. These codes can be derived from an established theory underlying the research, such as elements of a decision making model like the recognition-primed decision model or other macrocognitive model [6]. The codes can also be derived using a data-driven process whereby analysts review each excerpt, assess the main point or activity it conveys, and either assign a code from the list derived so far or create a new code to represent that main point or activity. A third option is to use a combination of theory- and data-driven codes.
- The set of codes should be documented and each code should be clearly defined.
- The coding process can be streamlined and supported by using qualitative data analysis software; Atlas.ti and NVivo are two well-known products that are available.
- Conduct practice coding. Two to three analysts should independently assign codes to a subset of 20 to 50 excerpts and then compare the choices made and discuss discrepancies. If the number of discrepancies is high, they should tweak the codes and definitions as needed and repeat the process with another 20 to 50 excerpts. Adjustments may be made to the set of codes during this alignment process.

- After all excerpts have been coded, assess inter and intra-rater reliability using the kappa statistic, which takes into account the number of codes and the probability of selecting the same code by chance.
- Compare raters' codes and identify discrepancies. Decide how to resolve discrepancies. Options include consensus, rater discussions, and retaining all assigned codes.
- Group excerpts by code and review grouped excerpts.
- Report code frequencies, write syntheses of excerpts within each code category, assess coded excerpts in terms of support for the theory used to generate the codes, and/or perform more a fine-grained analysis of the excerpts in each category (e.g., conduct another code-based analysis).
- Undertake a team-based review, discussion, and synthesis of content emerging from each code. This step is the real analysis; it can be enhanced by having your team meet to talk about key themes in depth, and to ideate about ways to organize or model findings emerging from the analysis.

Tips:

- Record all steps taken, decisions made (e.g., Can more than one code be assigned to a given excerpt?), and changes made, e.g., to codes, throughout the process.
- Coders should retain the original set of codes assigned to excerpts in addition to the final set of reconciled codes.
- Prior to coding, add a column to the excerpts spreadsheet where notes can be recorded.
- Use practice coding sessions to assess excerpts and make sure they are adequately concise and not too fragmented to interpret.
- Ask a SME to evaluate your assessments and interpretations of the data.
- The steps and tips above are intended to improve the validity and reliability of the analysis. Consider other methods for improving validity, such as comparing your findings with the broader literature or additional theories and approaching your analysis as a test of a null hypothesis. Consider other methods for improving reliability of the analysis, such as frequently checking-in with coding analysts and revisiting code definitions.

Companion Methods: This analysis method is used to analyze data collected using methods such as the Critical Decision Method, the Workflow Analysis Interview, and Think-Aloud methods. It may also be used to evaluate recordings of team communications and work observations. In general, it can be used to analyze and find meaning within any large set of qualitative data.

Resources and References

- [26] V. Braun and V. Clarke, "Using thematic analysis in psychology," Qualitative Research in Psychology, vol. 3, no. 2, pp. 77-101, 2006.
- [27] G. Walker, "Verbal protocol analysis," in *Handbook of Human Factors Methods*, N. Stanton, A. Hedge, K. Brookhuis, E. Salas and H. Hendrick, Eds., Boca Raton, FL, CRC Press, 2004.

2.2.2 Decision Requirements Table

A Decision Requirements Table (DRT) may be used to capture the results of a set of Critical Decision Method interviews, Simulation Interviews, or the results of any interviews or observations in which

Decisions and Cognitive Requirements	Why Difficult?	Current Strategy	Potential Solutions
Determine the weapons loadout for each aircraft.	Iterative process driven by multiple constraints, including the order in which target areas will be visited, the desire to keep loadouts homogenous if possible, and the need to ensure timing will work out.	Done almost entirely on a whiteboard. Loadout stores are allocated to targets and modified repeatedly as the detailed plan is created.	Loadout planning decision aid for 1-ship to 4-ship missions.
Determine transit times between target areas (transit times used to determine whether aircraft has enough time in LAR to release weapons).	Many different possible ways to flow from one target area to the next. May evaluate many different target sorts as the plan is iteratively worked out.	Use a range and bearing tool to plot a range and bearing line between each target area.	After grouping targets, automatically calculate and display distances between each target group.

Figure 2-8. Example Decision Requirements Table.

decision-making activities were uncovered. A DRT provides an in-depth view of each decision. DRTs are well-suited to inform the design of new automation or decision support capabilities.

Timeframe: Most time is spent analyzing data, which will vary with the complexity and amount of data. It takes about 1 - 2 weeks to analyze a set of interview data and create a DRT, if multiple team members are supporting full-time.

Use When: It is helpful to succinctly capture information about each decision or cognitive work element that requires support. DRTs can inform both design recommendations and training intervention recommendations.

Steps:

- **Collect data:** Collect data on decisions and decision making requirements using a gather method such as the Critical Decision Method or Simulation Interview.
- **Identify decisions:** Review the data (e.g. interview data, observations, etc.) to identify each key decision that requires support. Thematic analysis (see 2.2.1) provides an in-depth approach that can be tailored to need.
- **Analyze decisions:** For each decision, identify a set of information that will be captured in the DRT. Such information should align with the goals and purpose of the project but may include:
    - Challenges and difficulties: Information on what makes the decision challenging.
    - Current strategy: Information about the current strategy for making the decision.
    - Potential solution and how it will help: Possible design solutions to support the decision.

  - Cues/factors: Background information known before engaging in the decision (factors) and information from the environment (cues) or known information (factors) that impact decision making and situation understanding.
  - Frequency metrics: Information about the number of times a decision was made in the data set. This may be of use to prioritize decisions for inclusion in a support system.

Tips:

- When eliciting information about decisions during the gather phase, focus on challenging scenarios or situations. Routine incidents can often be handled automatically and don't expose challenging decisions.
- Constructing the DRT on a whiteboard during an interview can be used as a gather technique to directly elicit the table information with participants.

Companion Methods: The Critical Decision Method or Simulation Interview may be used to collect information about decisions in the gather phase.

Resources and References:

- [28] G. Klein, G. L. Kaempf, S. T. M. Wolf and T. Miller, "Applying Decision Requirements to User-Centered Design," International Journal of Human Computer Studies, vol. 46, no. 1, pp. 1-15, 1997.
- [29] D. Thompson, S. L. Wiggins and G. Ho, "Using cognitive task analysis to develop scenario-based training for house-clearing teams (No. DRDC-TORONTO-CR-2007-118)," Klein Associates, Inc., Fairborn, OH, 2006.

2.2.3 Goal Directed Task Analysis

Goal Directed Task Analysis (GDTA) is a cognitive task analysis methodology that identifies the goals users must achieve to accomplish their work, the decisions that must be made to achieve the goals, and the information required for each decision. A GDTA systematically decomposes each top-level goal into a set of supporting sub-goals, decisions, and associated information requirements. Information requirements are categorized using Endsley's situation awareness (SA) framework [30]; that is, information needed to support Level 1 (perception), Level 2 (comprehension), and Level 3 (projection) SA requirements are identified for each decision. The method aims to address the full range of work that must be accomplished in a domain to support user SA and decision-making. Thus, it is quite suitable for design efforts that comprehensively address many aspects of the user's work. The below figure is a MITRE-developed template for conducting GDTA. Additional annotations attached to goals and decisions may describe typical workflow sequences, pain points, tools used, artifacts created, and collaboration needs.

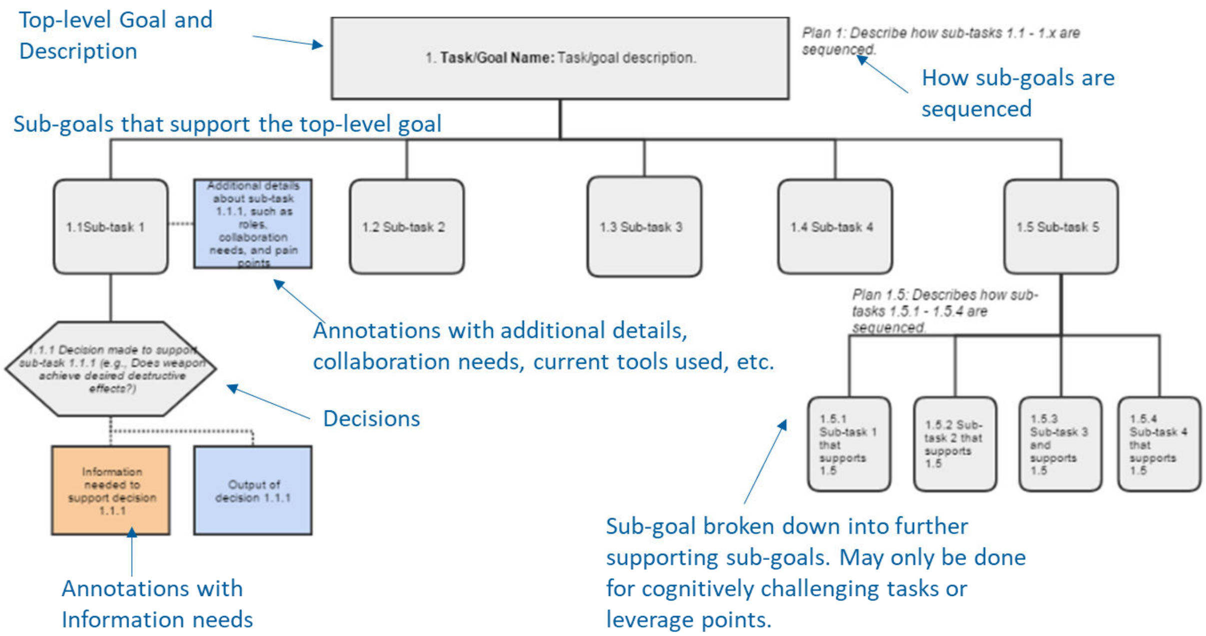


Figure 2-8. Goal Directed Task Analysis template.

Timeframe: About 1 month to create a full GDTA. The timeframe will vary with the complexity of the domain.

Use When: It is important to consider the full range of decisions and information requirements involved in the domain. For example, if a new system is being designed that will support many aspects of the users' work. Even if a small part of the domain is addressed in a design effort (e.g., to develop a "Minimum Viable Product"), it may still be valuable to develop a GDTA as a roadmap for potential future capabilities.

Steps:

- **Identify Goals:** Items that are appropriate for designation as goals are those that require cognitive effort and that are essential to successful task completion. They are higher-level items as opposed to basic information requirements. The goals themselves are not decisions that need to be made, although reaching them will generally require that a set of decisions and a corresponding set of SA requirements be known.
- **Define Decisions:** Define the decisions that are needed to effectively meet each goal in the goal hierarchy. Decisions reflect the need to synthesize information in order to understand how that information will affect the system both now and in the future. They are generally posed in the form of questions (e.g., "*Is it possible to revise routing without impacting weapons release requirements?*")
- **Delineate Information Requirements:** Decisions are posed in the form of questions (e.g., "Is it possible to revise aircraft routing to avoid threats without impacting weapons releases?"), and the associated information requirements provide the information needed to answer those questions. To determine the information requirements, each decision should be analyzed individually to identify all the information the user needs to make that decision. The information

requirements should be listed without reference to a specific technology or how the information is obtained.

Tips:

- It can take some time to create a GDTA that comprehensively addresses the full range of goals. If pressed for time, first outline the top-level goals, then focus on identifying sub-goals, decisions, and information requirements for those top-level goals that will be addressed in the new system.
- Online drawing tools (e.g., Gliffy and draw.io) are effective for creating GDTAs that can be easily updated, shared, collaboratively edited, and linked to supporting pages or other artifacts.

Companion Methods: Since a GDTA comprehensively covers the full range of goals, decision, and information requirements in a domain, conduct multiple Workflow Analysis Interviews in the gather phase that touch upon a variety of scenarios.

Resources and References:

- [31] M. R. Endsley, B. Bolte and D. G. Jones, Designing for Situation Awareness: An Approach to Human-Centered Design, London: Taylor & Francis, 2003.
- [19] R. R. Hoffman, B. Crandall, G. Klein, D. G. Jones and M. R. Endsley, "Protocols for Cognitive Task Analysis," Institute for Human and Machine Cognition, Pensacola, FL, 2008.

2.2.4 Cognitive Work Analysis

Cognitive Work Analysis (CWA) is an approach to comprehensively representing and assessing complex socio-technical work systems in terms of the constraints and resources that shape activity within the system. CWA is a set of detailed analyses pioneered to inform design of process control systems, such as found in the nuclear power, petrochemical, and oil and gas industries. A recent book by Naikar [32] provides case studies for team design and training applications as well. Employing CWA requires a good deal of expertise in the methodology.

Timeframe: A CWA analysis of a complex work domain can take three months to a year with multiple researchers contributing. The duration depends on factors including the depth and extent of the analysis, the number of CWA methods used, the ease with which required data are obtained, and the complexity of the work domain.

Use When: Assessing a complex sociotechnical work domain or sub-elements of that domain. The methodology can be used to identify technology and resource gaps; process improvements; and misalignments between tools, processes, resources, and higher level organizational goals and values. It can provide a useful framework for structuring and developing knowledge management and training systems. The methodology is frequently used for interface design and evaluation

Steps: CWA consists of five work-domain representation techniques, each described below. In practice, the five techniques are infrequently all used together. Before proceeding, the analyst should consider which CWA techniques to use, given analysis resources and objectives.

- **Work Domain Analysis:** The Work Domain Analysis produces a two-dimensional model of the work domain called an abstraction-decomposition space. Along one dimension, the analyst describes the work system in terms of increasingly concrete, i.e., less abstract, constraints. The second dimension is a system-decomposition dimension along which the sociotechnical work system is portrayed as a single unit, then subsystems, then functions, subassemblies, and components (the system elements used for this dimension can be varied). These two dimensions are crossed and the resulting matrix, or abstraction-decomposition space , presents the system in terms ranging from the high-level objective of the entire system (in the upper left corner) to the physical tools used in the smallest system or organizational unit (to the lower right corner). The matrix template is shown below.

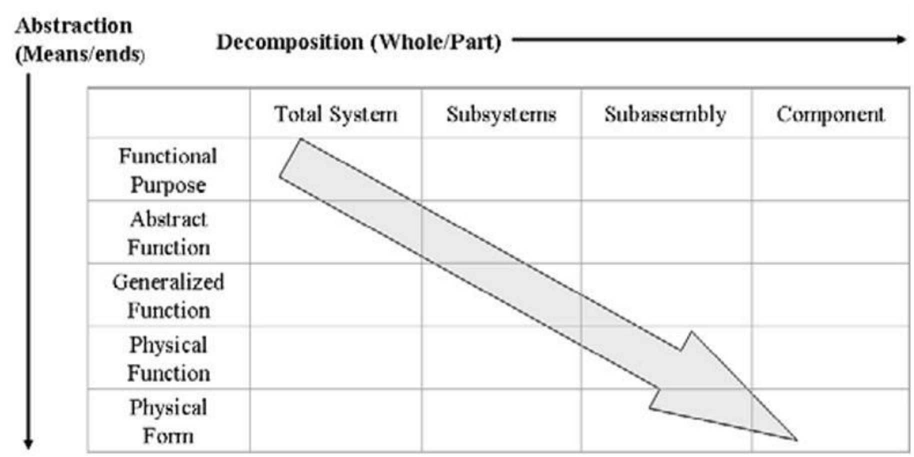


Figure 2-9. The abstraction-decomposition space representation.

- **Control Task Analysis:** The Control Task Analysis involves representing task performance using the Decision Ladder formalism. This analysis technique does not specify the process for understanding how tasks are performed; only for representing task performance. An example Decision Ladder is shown in Figure 2-10 below. It portrays task work as beginning in the lower left with the receipt of information (Activation) that triggers a need to act. The task is depicted as progressing upward from the bottom left corner of the diagram as a series of information collection and assessment activities that culminates at the top of the diagram in the choice of a response goal. The task is next depicted as a downward sequence of activities involved in executing a response to that goal. Ladder rungs are right-facing arrows connecting information collection and assessment activities to response execution activities informed by them.

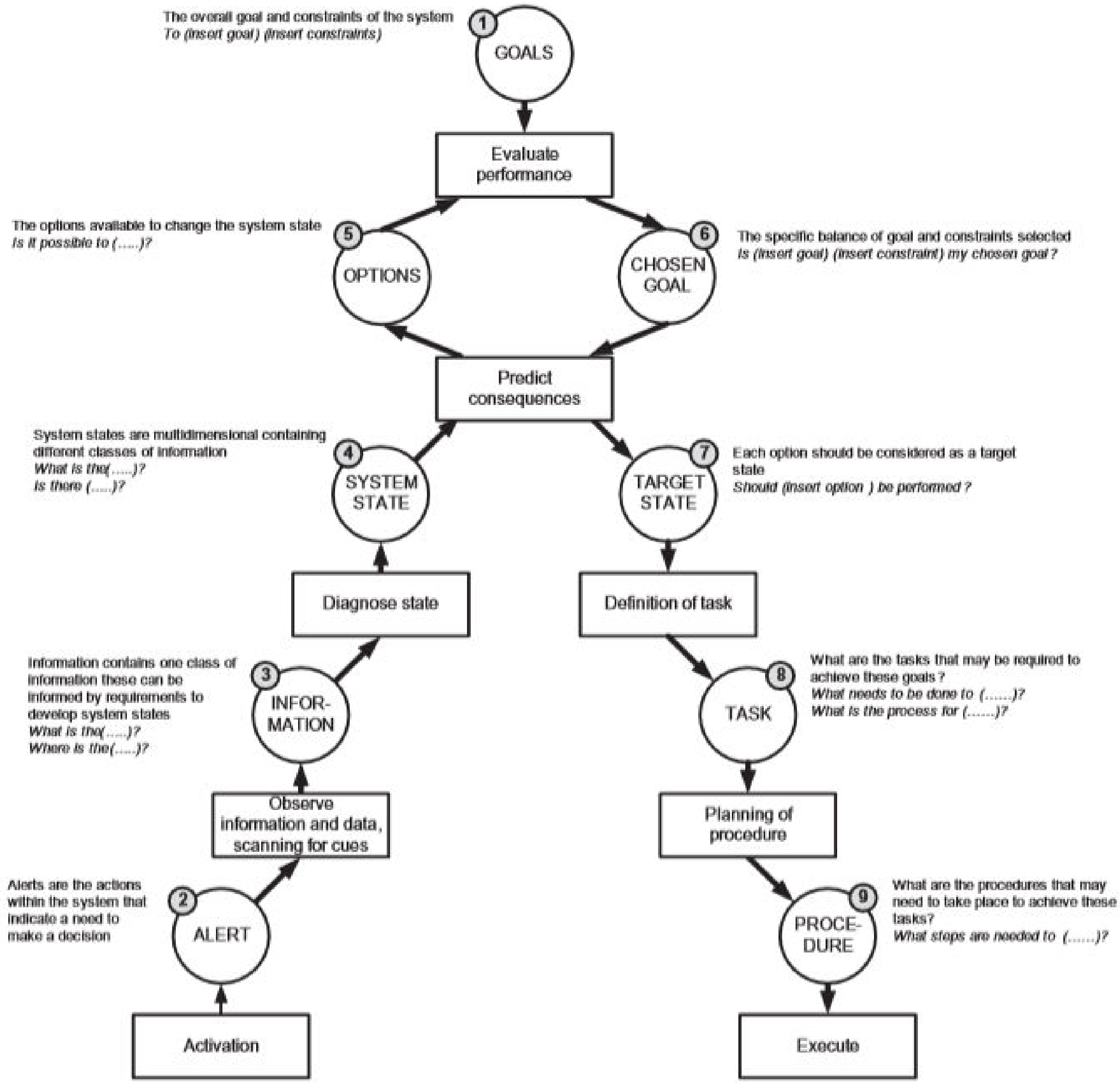


Figure 2-10. The Decision Ladder formalism for representing cognitive work.

- **Strategies Analysis:** Using information flow maps, the analyst represents strategies that system entities (humans, teams, and automation) employ to perform the tasks depicted in the Control Task Analysis.
- **Social Organization and Co-operation Analysis:** In this analysis, organizational and cooperative relationships are mapped to the abstraction-decomposition space, Decision Ladders, and information flow maps.
- **Worker Competencies Analysis:** This analysis technique involves using Rasmussen's Skill, Rule, and Knowledge framework [33] to specify and classify the cognitive activities employed during task performance.

Tips:

- CWA methods are used flexibly. An analyst may use any or all of the methods and in any order, and may adapt the method to suit his or her analysis objectives. For example, the analyst may enter text into the cells of the abstraction-decomposition space or represent cell contents using a means-end analysis format, as shown in the example from [34] in Figure 2-11 below.

	Whole System	Subsystems	Components
Functional Purposes			
Values and Priority Measures	Value and Priority Measure D		
Purpose-related Functions	Purpose-related Function A		
Object-related Processes	Object-related Process B; Object-related Process C		
Physical Objects			

Figure 2-11. An example of a means-end analysis within the abstraction-decomposition space.

- Methods for collecting data are not specified as part of the CWA methodology. It consists only of work domain representation techniques. The data used to develop the work domain representations must be collected using other methods (see Companion Methods).
- To improve the validity and accuracy of your CWA artifacts, develop them iteratively and obtain feedback from one or more SMEs following each iteration.

Companion Methods: A variety of data collection and assessment methods may be used to obtain the data needed for producing CWA representations. These include workplace mapping methods where the analysts uses observation, photography, and *in situ* inquiry, as permitted, to develop a map of a workplace showing physical work activities, flows, and resources. Roles, tasks, needs, knowledge, and other details can be layered onto the map. Interview and walkthrough methods can also be useful, especially when they elicit information about resources and constraints that shape task performance.

Resources and References

- [35] K. Vicente, Cognitive work analysis: Towards safe, productive, and healthy computer-based work, Mahwah, NJ: Lawrence Erlbaum, 1999.
- [32] N. Naikar, Work domain analysis: Concepts, guidelines, and cases, CRC Press, 2013.
- [34] N. Naikar, R. Hopcroft and A. Moylan, "Work domain analysis: Theoretical concepts and methodology (Technical Report No. DSTO-TR-1665)," Defence Science and Technology Organisation Victoria (Australia) Air Operations Division, 2005.
- [36] N. Naikar and P. Sanderson, "Evaluating design proposals for training-system definition with work domain analysis," Journal of the Human Factors and Ergonomics Society, vol. 43, pp. 529-542, 2001.

2.2.5 Writing Cognitive Work and Human-Machine Teaming Requirements and User Stories

The goal of writing requirements or user stories is to provide actionable guidance to system designers and developers. Such guidance is used to ensure that the system will effectively support the nature of the cognitive work that will be performed. While a system may have many technical requirements or user stories pertaining to user interface functions, requirements and user stories written from a cognitive engineering perspective are those that address the cognitive work that must be accomplished by the people using the system. Human-machine teaming requirements are those that explicitly address how users will interact with automation, autonomy, and/or AI. This section provides high-level guidance on steps for writing requirements or user stories to address cognitive work and human-machine teaming.

Timeframe: Varies depending upon the complexity of the system and the number of features it will provide. Upfront work gathering and analyzing information about the domain must be performed before writing requirements or user stories.

Use When: Specifying the characteristics of a new (or revised) system or capability that will support cognitive work.

Steps:

- **Determine whether to write requirements or user stories**: Typically, this choice is made by the overall program management team; user stories are often used in agile development approaches. User stories have the merit of explicitly addressing the value proposition that a feature will provide to a user, and they are often used in agile development efforts. They provide flexibility in the amount of details or context that can be provided. User stories capture the user’s need as a statement of the form “As a <user role>, I want to <what?, to accomplish some goal> so that <why?, some reason>”.

  Requirements have the merit of being precise and testable. They are generally written as “shall” statements that describe specific details a feature will provide, and they may also include measurable performance criteria. For example, “The system shall provide a graphical depiction of threat vulnerability along a route of flight.”
- **Identify the set of features that the system will support:** Review coded interview notes and knowledge representation artifacts (e.g., Goal Directed Task Analysis, Decision Requirements Table, or an abstraction-decomposition space) created during the gather and analyze phases. Look for pain points, leverage points, key goals, key decisions, and positive aspects of the legacy system to preserve. Identify a set of features to address these elements. The selected features should provide a comprehensive suite of capabilities that support the end-to-end cognitive workflow or top level goals of the system. If using agile development, choose a minimum set of core, high priority features that will comprise the “minimum viable product.”
- **Write cognitive work requirements or user stories for each feature:** For each feature, conduct another pass through coded interview notes or knowledge representation artifacts. Identify the information that must be provided to the user to support their decision making and situation awareness needs. Also identify decision making strategies that were identified. Determine the nature of the display design that will support these needs, including how information elements

should be represented on the display. Each feature may have one or more display design requirements, decision requirements, and information requirements.

- **Review cognitive engineering design principles:** Review relevant cognitive engineering design principles given the feature set and nature of cognitive work. Tailor the design principles as additional requirements or user stories.
- **Write human-machine teaming requirements or user stories** [17]**:** If features will incorporate automation, autonomy, and/or AI, write human-machine teaming requirements or user stories. For each relevant human-machine teaming theme (e.g., Directability, Calibrated Trust, etc.), review coded interview notes and extract all relevant quotations. Tailor the general human-machine teaming requirements provided for each theme [17] based on the interview notes.

Tips:

- Write requirements or user stories that explicitly address how the system will support the user's decision making needs. Requirements or user stories should also be sufficiently detailed to address the nature of the display design to support the decisions.
- Requirements and user stories should not be viewed as static. They will evolve as user needs evolve, feedback is collected in the evaluate phase, and as users adapt software for purposes beyond its original envisioned usage. Expect to revise and rewrite requirements or users stories even for implemented capabilities as those capabilities are revised or extended.
- If user stories are written, specific requirements may still be provided for each user story that more precisely describe the required features.

Companion Methods: The HMT Knowledge Audit may be used to identify HMT requirements. Goal Directed Task Analysis, Cognitive Work Analysis, and Decision Requirements Tables are effective in capturing goals and decisions in a comprehensive manner.

Resources and References:

- [37] W. C. Elm, J. W. Gualtieri, B. P. McKenna, J. S. Tittle, J. E. Peffer, S. Szymczak and J. Grossman, "Integrating cognitive systems engineering throughout the systems engineering process," Journal of Cognitive Engineering and Decision Making, vol. 2, no. 3, p. 249–273, 2008.
- [17] P. McDermott, C. Dominguez, N. Kasdaglis, M. Ryan, I. Trahan and A. Nelson, "Human-Machine Teaming Systems Engineering Guide," The MITRE Corporation, Bedford, MA, 2018.
- [38] Anonymous, "User stories," Mountain Goat Software , [Online]. Available: https://www.mountaingoatsoftware.com/agile/user-stories. [Accessed 19 January 2020].
- [39] K. E. Wiegers, "Writing quality requirements," Software Development, pp. 44-48, May 1999.

2.3 Design Methods

This section provides a cognitive engineering filter on how to produce results of the first two phases of the cycle. Design methods are used to create systems and displays that support cognitive work and facilitate human-machine teaming. Design artifacts vary in fidelity from paper-based, annotated mockups and storyboards to software prototypes and functional software capabilities.

2.3.1 Storyboarding

Storyboarding outlines and visually captures a set of conceptual workflows or task procedures during the system design phase. There are many ways to utilize the storyboarding method, but particularly in support of cognitive engineering, this process helps to brainstorm important interactions that support key decision-making and cognitive support needs. The goal or output of storyboarding is to provide a series of steps that depict the user and system interactions using narrative descriptions, low-fidelity user interface mock-ups, and/or drawings providing an initial set of designs for the user experience. An example storyboarding template is shown in Figure 2-12 below. Using the template, user interface visuals would be provided in each box, and captions describing the user actions and intent would be provided beneath each box.

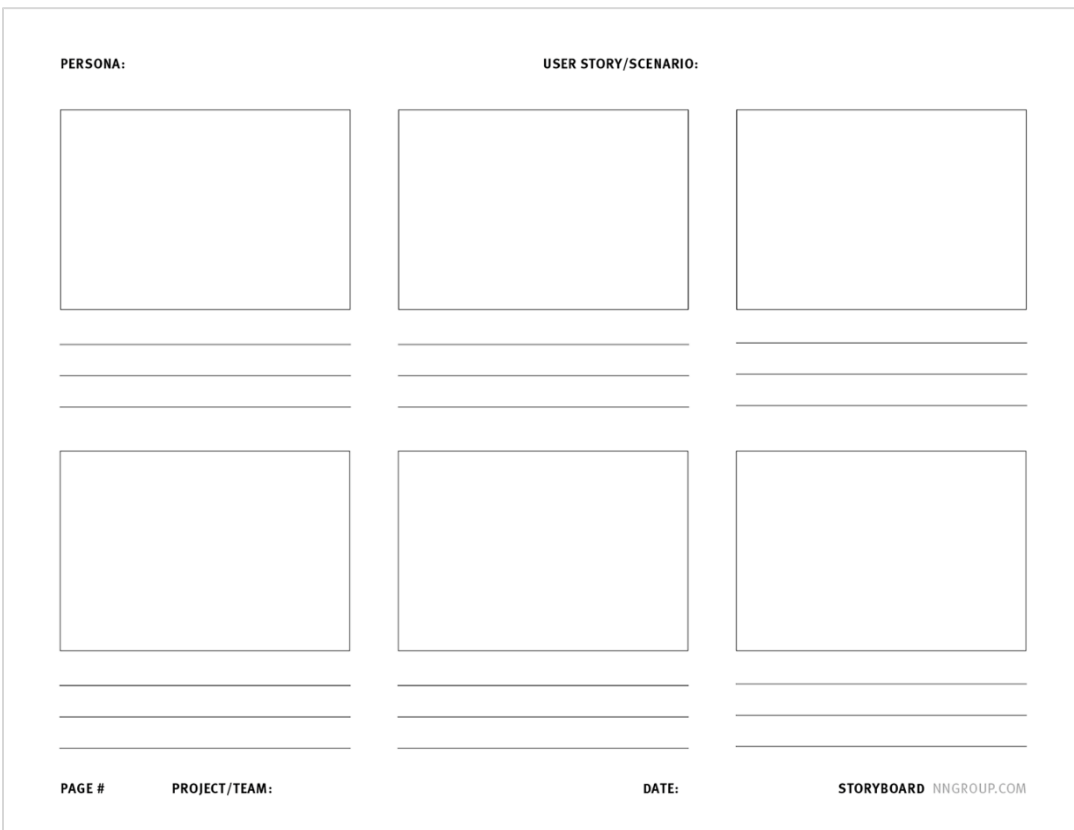


Figure 2-12. Storyboarding template example from [40].

Time Frame: Varies based upon the complexity of the system, number of screens to design, and level of fidelity of the designs. 1 – 2 days to create basic wireframes for a handful of screens to a month or more to create a complex set of storyboards.

Use When: During the early stages of the system design phase. Use iteratively as design concepts are revised and alternative concepts are explored.

Steps:

- **Identify scenario:** Identify at least one set or sets of scenarios to work through the storyboarding process. A scenario describes one or several user goals, for example "A pilot flying solo needs to plan a cross-country flight that will be conducted under visual flight rules". Gather methods such as the Critical Decision Method and Workflow Analysis Interview can be used to identify scenarios.
- **Identify scenario steps:** Break the scenario down into frames or chunks that depict specific actions or interaction behaviors.
- **Design visuals:** Sketch out the type of user interactions that could support the depicted user actions.
- **Add captions:** Use notes to identify how the actions depicted in each storyboard frame supports the overall cognitive workflow (decisions, situation understanding, sensemaking, problem solving, etc.).
- **Validate and iterate:** Once the given scenario has been thoroughly worked through the storyboarding process, circle back to ensure the overall workflow is validated with SMEs.

Companion Methods: Gather methods such as the Critical Decision Method and Workflow Analysis Interview can be used to identify scenarios. A Heuristic Evaluation can be used assess an early design resulting from the storyboarding process.

Resources and References:

- [28] G. Klein, G. L. Kaempf, S. T. M. Wolf and T. Miller, "Applying Decision Requirements to User-Centered Design," International Journal of Human Computer Studies, vol. 46, no. 1, pp. 1-15, 1997.
- [14] J. Klamm, C. Dominguez, B. Yost, P. McDermott and M. Lenox, "Partnering with technology: The importance of human machine teaming in future MDC2 systems," in SPIE Defense and Commercial Sensing Conference, Baltimore, MD, 2019.
- [40] R. Krause, "Storyboards Help Visualize UX Ideas," Nielsen Norman Group, 15 July 2018. [Online]. Available: https://www.nngroup.com/articles/storyboards-visualize-ideas/. [Accessed 18 January 2020].
- [41] Anonymous, "Usability Body of Knowledge: Storyboard," User Experience Professionals' Association, [Online]. Available: https://www.usabilitybok.org/storyboard. [Accessed 27 January 2020].

2.3.2 Participatory Design Events

Participatory design events are multi-day workshops that engage the cognitive engineering team, designers, and developers in a set of activities to co-create new design concepts. They may also be used to identify ways to improve existing design concepts. During the event, participants come to a shared understanding about the nature of the work and priorities. They then ideate on design concepts to support the work.

Timeframe: Several weeks to plan the event and prepare materials, 2 – 3 days to conduct the event, several weeks to create and refine design concepts resulting from the event and write a report documenting the designs and functionality.

Use When: Early in the design process when generating candidate design concepts in partnership with users. Use it to understand how people think about a problem, discipline, or technology and when there is, or could be, an cultural or political disconnect between you and the end user. Use iteratively as designs are revised and alternative design concepts are explored. This method can also be used later in design to make User Experience (UX) improvements, as done with DARPA Hallmark Phase 2 (see Section 3.3.3).

Steps:

- Define goals: Define the goals of the event and the design topics that will be investigated. Frame the goals into 'How might we…' (HMW) questions, such as 'How might we enable planners to rapidly visualize and compare the outcomes of candidate courses of action?'
- Recruit participants: Recruit users or SMEs to participate in the event. Ensure that at least 3 users or SMEs will be present and that there will be designers and developers participating as well as cognitive engineers.
- Plan and prepare materials: Create an agenda for the session and prepare materials that will prompt discussion. A participatory design cycle, addressing one 'how might we…' question, is generally 2-3 hours long, and you might conduct multiple sessions throughout the day. Design sessions work best when conducted in groups with between 3 – 10 people. If many end users are present, divide them across multiple groups. Ensure an end user/SME, a design team member, and a notetaker is assigned to each group.
    - Plan how each HMW cycle will flow time-wise and what team members will a) introduce the HMW question, b) facilitate unbounded ideation, c) prioritize and bin the ideas into groups; d) support creation of designs, and e) facilitate presentation of designs back to the entire group for discussion and voting.
- Conduct the event: Each design session should consist of: discussing the issue or feature for investigation, reviewing the scenario the feature will support, gathering specific design ideas, and wrapping up.
- Synthesize results: Identify the most promising design concepts that resulted from the sessions to further iterate upon and refine.

Tips:

- Create a relaxed environment with a slow conversational pace, food, and an ice-breaking warm-up exercise to spark the idea that everybody is a designer. Creative design is hard work and should not be rushed or forced.
- Spend time ensuring the room understands each HMW question deeply before starting ideation on that question.
- Ensure your room/environment inspires creativity and enables success: post brainstorming rules, examples of designs, and other helpful materials.

- Be sure that each design group involves participants drawing design solutions and not just writing text ideas.
- While the focus of a design event may be creating design solutions, greater understanding of the domain will naturally emerge during the event. Be sure that this understanding is captured along with the design concepts. Have on hand large capture sheets for each team to fill out annotating rationale for design concepts, and describing the cognitive work designs will support and how it will be supported.
- Videotape teams' final presentations of their designs so that their team's verbal rationale is captured.

Companion Methods: Storyboarding may be used during the event to capture and ideate on design concepts.

Resources and References:

- [42] Anonymous, "Participatory Design/Co-design Worksession," OpenInnovation Toolkit, [Online]. Available: https://toolkit.mozilla.org/method/participatory-designco-design-worksession/. [Accessed 19 January 2020].
- [43] Anonymous, "Participatory Design Workshop Guidelines," iTEC, [Online]. Available: http://itec.aalto.fi/participatory-design/workshop-guidelines/. [Accessed 19 January 2020].
- [44] E. a. S. P. J. Sanders, "Co-creation and the new landscapes of design," Co-design, vol. 4, no. 1, pp. 5-18, 2008.

2.3.3 Cognitive Engineering Design Principles

While a multitude of design principles exist in the User Interaction (UI) User Experience (UX) literature (e.g., the Nielsen's Heuristics [45]), this section provides a tailored set of principles that explicitly address designing to support cognitive work and human-machine teaming. It is still necessary to follow UI and UX principles to ensure that basic system usability is maintained, however.

Select those design principles that are appropriate to the features of the system and refer to them throughout the design process. Design principles may also be tailored and codified as formal system requirements or user stories. Below are evidence-based design principles drawn from research on effective human-machine teaming, Joint Cognitive System design, and Situation Awareness Oriented Design:

- **Design to support Observability** [17]**:** Observability provides transparency into what the system is doing relative to task progress. A system is observable when it provides the right level of information so humans understand how it is doing calculations and arriving at recommendations and predictions.
- **Design to support Predictability** [17]**:** With Predictability, future intentions and activities of the system are discernible and understandable. Predictability also means that the system can anticipate changes in situations that aid the user in projecting future states. [17]
- **Design to Direct Attention** [17]**:** The system must be able to direct the attention of the human to critical problem features, cues, indications, and warnings. The system should communicate

proactively when information becomes relevant, such as when obstacles to meeting goals are encountered.

- **Design to support Exploring the Solution Space** [17]**:** Exploring the Solution Space helps users leverage multiple views, knowledge, and solutions to jointly understand the problem space. The system should be able to rapidly generate multiple distinct courses of action and give users ways to rapidly compare those solutions. Both the human and system should be able to broaden or constrict the solution considerations to shift perspectives.
- **Design to support Adaptability** [17]**:** Adaptability enables users to recognize and adapt fluidly to unexpected characteristics of a situation. The system should have multiple options to recognize an unexpected situation and address it.
- **Design to support Calibrated Trust** [17]**:** Calibrated Trust is supported when users have a strong understanding of when and how much to trust an automated partner in context. Users should understand clearly when automation can be relied on, when more oversight is needed, and when performance is unacceptable. The system should provide information sources and the credibility of those sources to help users calibrate trust.
- **Design to achieve Common Ground** [17]**:** Achieving Common Ground means that pertinent beliefs, assumptions, and intentions are shared among team members. Common Ground should be constantly and actively updated and maintained so that team members (humans and systems) can maintain a shared picture of what's happening in the world and engage in backup behavior to support each other.
- **Design for effective Information Presentation** [17]**:** Present information in a manner to support simplicity and understandability. The user should be able to view and interact with information in order to understand the implications of the data.
- **Organize information around goals and decisions** [31]**:** Information should be organized in terms of users' major goals, rather than in a way that is technology-oriented. Combine and presented information elements in a manner that is congruent with the decision making goals of the user.

2.4 Evaluate Methods

Evaluate methods are used to assess the degree to which design concepts, prototypes, and system capabilities support cognitive work and facilitate effective human-machine teaming, within the operational context intended for the system as much as possible. Design should be held as a hypothesis about its effectiveness until it is rejected or accepted in light of empirical evidence. Thus, regularly engaging users and conducting evaluations is an integral part of the cognitive engineering process. While there is no one-size-fits-all approach, an example sequence of evaluation activities may include:

- Conducting regular lightweight formative evaluations of prototypes, designs, and operational software as the system is developed. Iterating and refining the prototypes, designs, or operational software based on the results. Examples of lightweight formative evaluation techniques include the Cognitive Wall Walk, Heuristic Evaluation, and the Cognitive Performance Indicators (CPI) Evaluation
- Conducting a series of summative evaluations using Human-In-The Loop Simulation Event (Evaluation Event) or through Formal Usability Testing after each development increment or milestone. The goal of a summative evaluation is to comprehensively assess the performance of the system.
- Conducting a final summative evaluation of the system to ensure it meets its measurable targets.

The methods and amount of material related to evaluation is overwhelming, and the area is also explained and illustrated in Section 3 of this document. To streamline this section, we have placed important supporting methods (Cognitive Performance Indicators Evaluation, Formal Usability Testing, Defining Cognitive Evaluation Measures, and Developing Decision Centered Test Scenarios) in Appendix B.

2.4.1 Cognitive Wall Walk

The Cognitive Wall Walk is a variant of a usability inspection method called the cognitive walkthrough. During a Cognitive Wall Walk, designs are reviewed by SMEs while walking through one or more realistic scenarios. Scenarios may include routine activities that the system is intended to support as well as tough cases that involve challenging decisions. At various points in the scenarios, participants are asked how the design concepts would support their tasks and decision making objectives.

Timeframe: Will vary based on the number of user interface screens. May be conducted in a half-day or day-long design workshop with multiple participants. May also be conducted in with individual participants. Weeks to months will be needed to develop the scenario and specific storyboards, if not already available, as well as to recruit participants..

Use When: Storyboards or design mockups are available; use early in the design process to validate and improve designs before implementation. Use iteratively after each round of design to continually improve designs.

Steps:

- **Identify the scenarios to be used in the evaluation:** Identify scenarios that address the most important tasks the system was designed to support. If such scenarios do not already exist, identify prior incidents that were “tough cases” and use those in the evaluation. Tough cases may be identified using a gather method such as the Critical Decision Method.

- **Prepare design materials:** Identify the designs that will be used to support each scenario phase. Sequence the designs in order of the scenario phases.
- **Walk through the scenarios:** Walk through each phase of each scenario. Describe the scenario phase and show the design concepts intended to support the phase. Ask participants to describe the steps they would go through and the thoughts they would have if they were to use the design to accomplish each scenario phase.
- **Ask follow-up questions:** After each scenario phase, ask follow-up questions about how the design features supported (or did not support) the scenario phase. Also ask what else participants would like to see in the scenario, including which actions they would need to take that are not presently supported in the designs.
- **Consolidate findings:** Consolidate comments on design changes to incorporate into the next design iteration. Revise the scenarios if necessary based on participant feedback.

Tips:

- It is valuable for both the design and development team members to participate in Cognitive Wall Walk events. This provides designers and developers with an essential understanding of the situations users face that they can carry into subsequent design and development activities, and exposes them to first-hand feedback.

Companion Methods: The Critical Decision Method can be used to identify challenging scenarios that the system must support.

Resources and References:

- [46] C. Dominguez, A. Grome, R. Strouse, B. Crandall, C. Nemeth and M. O'Connor, "Linking Cognitive Data to Design in Navy Command and Control," in INCOSE International Symposium, San Diego, CA, 2010.
- [47] E. L. Papautsky, C. Dominguez, R. Strouse and B. Moon, " Integration of cognitive task analysis and design thinking for autonomous helicopter displays," Journal of Cognitive Engineering and Decision Making, vol. 9, no. 4, pp. 283-294, 2015.

2.4.2 Heuristic Evaluation

A Heuristic Evaluation is a usability inspection method that assesses a system's compliance against a set of usability principles (heuristics). The goal of this method is to identify usability problems with the user interface and to help define the type of problems that exist. This method can be conducted by one or several reviewers, typically usability experts, and provides a means to continually improve system interface designs.

Timeframe: Usually quick to conduct, but will vary with the number of user interface screens. Typically 2 - 3 hours to conduct the actual evaluation (per evaluator), several days on the front and back end of the evaluation to prepare materials and consolidate and prioritize findings.

Use When: An interactive prototype, storyboards, or screenshots of the system to be tested are available; during the initial stages of user interface design. The method should be used iteratively, to ensure that all usability issues are resolved.

Steps:

- Identify key tasks the system is designed to support as a means to conduct the review. Tasks that are frequent and tasks that are complex or challenging allow for a thorough review of a system.
- Identify a set of heuristics to assess the system. There are a few sets used in industry practice and a subset or variation of those principles can be tailored for any specific needs.
- Acting as the intended user, try to accomplish each task, documenting and categorizing the usability issues encountered. Alternatively, interview actual users about how well the system capabilities fulfill each heuristic, eliciting examples.
- After all tasks are complete and all issues have been documented, review each finding and assign a severity rating to each issue. Typically, issues that could prevent a user from accomplishing the overall task goal would be considered more severe, while more minor issues would be categorized as less severe.
- Compile all findings and provide an overall count/summary of issues discovered and where they occurred within a given task flow sequence. A spreadsheet may be used to efficiently compile findings.

Companion Methods: Gather methods such as the Task Diagram Interview can be used to inform key tasks. Heuristic Evaluations can be used to assess Storyboards.

Resources and References:

- [45] J. Nielsen, "10 Usability Heuristics for User Interface Design.," April 1994. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/. [Accessed January 19 2020].

2.4.3 Human-Machine Teaming Heuristic Evaluation

The Human-Machine Teaming (HMT) Heuristic Evaluation is a variation of the Heuristic Evaluation technique that aims to uncover specific information about what autonomy does well and what the shortcomings are. Instead of using a set of usability heuristics, it uses 9 themes that describe effective HMT design (e.g., Observability, Adaptability). An HMT Heuristic Evaluation may be administered as either an interview or questionnaire. During the interview or questionnaire, participants are asked to rate the degree to which the system exhibits each HMT theme; they are then asked follow-up questions on the reasons behind their ratings.

Timeframe: 1 – 2 hours to conduct each evaluation interview or questionnaire, several days to consolidate and prioritize findings.

Use When: A functioning AI, autonomous or automated system or prototype (or a system or prototype with several of these features) to be tested is available. Use iteratively after each round of development to continually improve the system.

Steps:

- Prepare materials: Tailor and print an HMT Heuristic Evaluation interview guide for each participant (available in [17]). The interview guide includes example questions for each HMT theme, along with examples that illustrate what the theme may look like in context.
- Recruit participants: Recruit users who are familiar with the system.

- Conduct the evaluation: Conduct evaluation interviews or administer the evaluation as a questionnaire for each participant.
- Consolidate findings: After completing the individual evaluations, consolidate findings across participants. For each HMT theme, identify successes (areas where the system exhibits the theme) and gaps (areas where the system does not support the theme). For gaps, provide recommended improvements.

Tips:

- The interview format is more effective and timely than the questionnaire format. It allows evaluators to ask follow-up questions on interesting/informative comments.
- This method is primarily intended for actual users of the system. However, design and development team members who understand the system context of use (or have observed usage of the system) may also apply this method.

Companion Methods: Consider also conducting a Heuristic Evaluation to identify usability issues.

Resources and References:

- [17] P. McDermott, C. Dominguez, N. Kasdaglis, M. Ryan, I. Trahana and A. Nelson, "Human-Machine Teaming Systems Engineering Guide," The MITRE Corporation, Bedford, MA, 2018.

2.4.4 Human-In-The-Loop Simulation-Based Testing (Evaluation Event)

In a Human-In-The-Loop (HITL) simulation event, system concepts and working software are evaluated in the context of a realistic, simulated environment. For example, new support concepts to aid pilots may be evaluated in a flight simulator. One or more users participate in such simulations, either individually or in teams (the team structure should be authentic to current or projected operational use). HITL simulations are used to compare systems or candidate design concepts, ensure that systems meet certain objective performance criteria, identify areas for improvement, assess the degree to which changes made to a system improve (or degrade) performance, or to baseline performance with and without certain system capabilities. They are also highly effective as a training tool. This section provides high level guidance on conducting HITL simulation events. The DARPA Hallmark program used this approach as described in Section 3.3Section.

Timeframe: Planning and conducting a HITL simulation event requires considerable time and effort. It often takes months to define the evaluation measures and data collection strategy, recruit and schedule participants, create scenarios and sample data, develop training and operations processes, gain human subjects review approval, configure the simulation and software, test the configuration, conduct the event, and analyze results. This will vary depending upon whether simulation and data collection testbed infrastructure are already in place.

Use When: An interactive prototype and simulation testbed are available. Use after each development increment or milestone to assess the impact of design changes in a rigorous, experimental manner. Can also be useful to train users.

Steps:

- Define the questions: Clearly define the research and/or development questions to be answered, or the experimental design to be conducted, during the HITL simulation. Having clearly stated questions and goals will frame the measures that are collected. For example, a question may be to determine the effectiveness with which a system directs the user's attention to changing conditions.

- Define the cognitive evaluation measures: Determine the qualitative and quantitative measures (dependent variables) that will be assessed to answer the questions and the type of data that will be collected for each measure. For example, you may define a suite of measures that address how well the system supports cognitive work and decision making, the level of mental workload required to perform such cognitive work, the degree of situation awareness the system provides, and the overall usability of the system.
- Specify conditions: Specify the conditions (independent variables) to be compared. For example, you may compare how a single system supports cognitive work under varying conditions, you may compare performance before and after making design changes, you may compare performance with and without a certain system capability, or you may compare performance with a system capability configured in different ways.
- Specify subject participation: The simulation may also be set up using either a within-subjects design, in which every participant is exposed to every level of every independent variable, or as a between-subjects design, in which each participant is exposed to a different scenario. Within subjects designs are generally less complex to execute and require fewer participants, but it can be challenging to recruit the same participants for repeated periodic evaluations.
- Define the test scenarios: Define a realistic operational scenario and the associated set of required data that will be executed in the HITL simulation. The scenario should incorporate challenging cognitive work and decision making, and it should be designed to illuminate the evaluation measures of interest.
- Identify participants: Identify the participants that will be needed to fill each role in the HITL simulation. Certain roles may be filled by participants not directly involved in the evaluation (e.g., air traffic controllers may be required in an evaluation whose focus is pilot performance using a new decision aid).
- Create training materials: Determine what training should be provided to participants and develop it.
- Create other supporting materials as required, including:
    - Develop a pre-test questionnaire to elicit demographic information.
    - Develop a post-evaluation questionnaire, for example, with free text questions; include instruments such as the System Usability Scale (SUS) [48] as appropriate for research goals.
    - Develop a guide for observers and data collectors, so that they know what to look for and capture; including a template for them to fill out while observing.
- Conduct the evaluation: Participants receive training as required and participate in the simulation event. Throughout the event, cognitive engineers, designers, developers, stakeholders, and SMEs should be available to address any issues, answer questions, collect data, and observe how the system is used by operational users in context. For data collection that is not automated, the appropriate data collection team should be in place.

- Analyze results: Analyze and summarize the qualitative and quantitative measures collected during the evaluation. Identify positive elements of the design, evaluation measure outcomes, design issues, key user comments and quotes, usability issues, and prioritized recommendations for improvements.

Tips:

- Dry run the simulation event and extensively test the software and data collection infrastructure ahead of time. Holding a series of planning conferences and telecons was an effective means for the DARPA Hallmark program to coordinate multi-team member progress and readiness, and to surface potential issues and solutions.
- Quantitative metrics need to be presented in context, not just as scores and trends, but analysis of why and what might be done to improve the metric.
- Ensure that both the outcomes of a HITL simulation (e.g., the metrics that were collected) and the reasons behind the outcomes (e.g., the processes that were employed to achieve the outcomes) are evaluated. Understanding the processes that were employed may be achieved through observation and follow-up interviews with participants.
- Have a plan in place to act upon the results of the HITL simulation. Apply appropriate design resources to address feedback and cognitive evaluation results.
- If conducting multiple HITL simulations throughout an iterative development effort, ensure they are spaced far enough apart to allow time to act upon results.

Companion Methods: See sections on Developing Decision-Centered Test Scenarios and Defining Evaluation Measures for guidance on developing scenarios and evaluation measures.

Resources and References:

- [49] B. Crandall, G. Klein and R. R. Hoffman, "Cognitive task analysis for measurement and evaluation," in Working Minds: A Practitioner's Guide to Cognitive Task Analysis, B. Crandall, G. Klein and R. R. Hoffman, Eds., Cambridge, MA, MIT Press, 2006, pp. 229-244.
- [50] V. J. Gawron, Human Performance, Workload, and Situational Awareness Measures Handbook, Boca Raton, FL: CRC Press, 2008.

2.5 Planning a Cognitive Engineering Toolchain

A cognitive engineering "toolchain" is a set of related methods where outputs from a prior method are inputs to a subsequent method. Each project is different and there is no one-size-fits-all approach to cognitive engineering; these example toolchains are provided to help project/program staff and practitioners see how various methods might fit together across a cycle of development, whether it be waterfall or agile. In the case of agile development, the pacing and program goals will drive how cognitive engineering methods are chosen and scheduled in concert with agile cycles that are conducted in an iterative manner.

2.5.1 Short-Term Project Toolchains

These toolchains are appropriate for quick-turn efforts lasting approximately three to six months.

- Recommend changes to an existing system: (1) Gather information about usage scenarios using a Task Diagram Interview combined with a Knowledge Audit or HMT Knowledge Audit; (2) Design using rapid prototyping facilitated by group design sessions involving cognitive engineers and SMEs; (3) Evaluate using a Cognitive Wall Walk, Heuristic Evaluation, HMT Heuristic Evaluation, or Cognitive Performance Indicators Evaluation. The evaluation phase might start the cycle in order to prioritize key elements needing change. Note that this toolchain effectively cuts out an in-depth analysis process, requiring prototype developers to be familiar with the knowledge elicitation data gathered, and have some sort of representation of it as an input.
- Design a new decision support visualization: (1) Gather using the Task Diagram Interview and Knowledge Audit or the HMT Knowledge Audit; (2) Analyze using a Decision Requirements Table co-created during interviews and user story development; (3) Design via holding a Participatory Design Event that feeds into mockup development, either on a whiteboard or using lightweight mockup development software. (4) Evaluate using a Cognitive Wall Walk and/or Heuristic Evaluation.

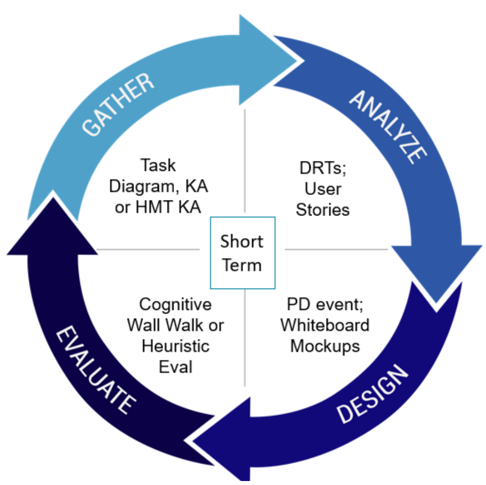


Figure 2-13. Quick-turn cognitive engineering toolchain.

2.5.2 Medium-Term Project Toolchains

These toolchains are appropriate for projects lasting six months to one year.

- Design a new decision support system: (1) Gather data using the Critical Decision Method or Applied Cognitive Task Analysis tools; (2) Analyze using thematic analysis, to the depth possible, combined with creating a Decision Requirements Table; (3) Design using Participatory Design Events to jumpstart ideation for Storyboarding, incorporating Cognitive Engineering Design Principles; (4) Evaluate using the Cognitive Wall Walk and Heuristic Evaluation.

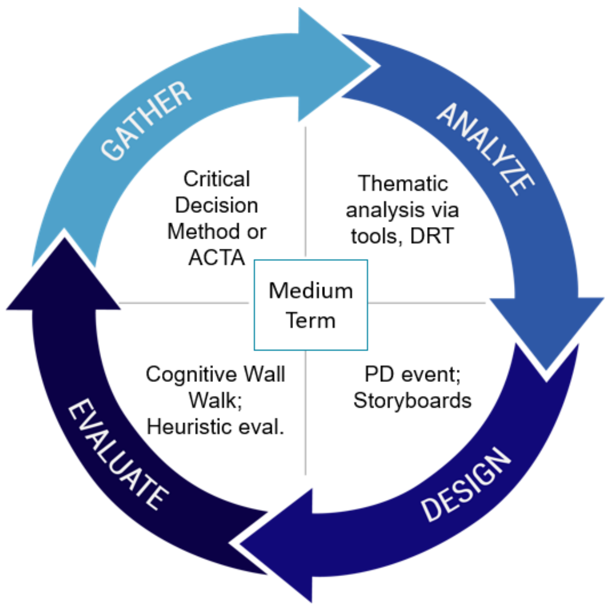


Figure 2-14. Medium-term project cognitive engineering toolchain.

2.5.3 Long-Term Project Toolchains

These toolchains are appropriate for multi-year projects that involve all cognitive engineering phases, including gather, analyze, design, and evaluate, repeated in a manner that conforms to the larger program goals and cycles. Again, the methods chosen will depend on project goals; this toolchain provides a means for completing repeated cognitive engineering cycles that should be tailored accordingly.

- Design a new sociotechnical system: (1) Gather using the Critical Decision Method and other methods fitting the project; (2) Analyze using in-depth thematic analysis, or via Cognitive Work Analysis or Goal Directed Task Analysis if those methods fit project goals; (3) Design using Storyboarding, Participatory Design Events, and Cognitive Engineering Design Principles; (4) Evaluate using Cognitive Wall Walks and Human-In-The-Loop Simulations or Formal Usability Testing.

- Improve an existing sociotechnical system: (1) Gather using Workflow Analysis Interviews, the Critical Decision Method, and Wagon Wheel Interviews; (2) Analyze using Cognitive Work Analysis or Goal Directed Task Analysis; (3) Design using Storyboarding, Participatory Design Events, and Cognitive Engineering Design Principles; (4) Evaluate using Cognitive Wall Walks and/or Human-In-The-Loop Simulations, or Formal Usability Testing, to include applying Cognitive Evaluation Measures.

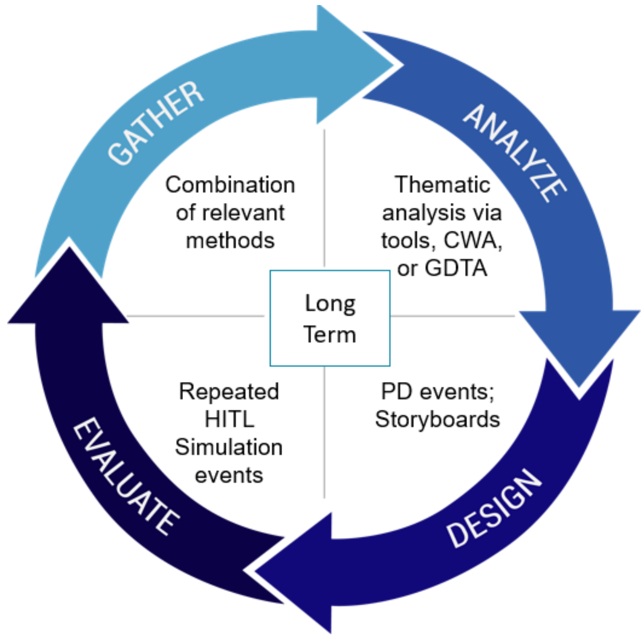


Figure 2-15. Long-term, multi-year project cognitive engineering toolchain.

3 Integrating Cognitive Engineering into a DARPA Program

By charter, DARPA addresses hard problems that the DoD will face in the future. Sometimes those problems focus on technology advances alone, or are basic research challenges that pre-date concerns about implementing technology or innovations in a mission- or systems-based environment. Other DARPA hard problems, such as Hallmark's Space Enterprise Command and Control (SEC2), include processes and systems or tools that might be transitioned in the near future. For those latter problems, meeting the criteria outlined in Section 3.1 below, we offer DARPA program managers experience-based, research-driven guidance on how cognitive engineering can be implemented to improve program impact.

3.1 Characteristics of DARPA Programs Requiring Cognitive Engineering Support

As outlined in Section 1.1 above, programs will benefit from cognitive engineering processes and methods if their systems under development involve:

1. Multiple capabilities, technologies, and people working together towards a mission with high-stakes consequences, under conditions of time pressure, shifting goals, and ambiguity [5]
2. People who must plan, assess, understand, analyze, coordinate, and decide [4]
3. Organizational, cultural, and economic constraints that tend to push operations towards the boundaries of safe performance [7]

Examples include programs developing user facing software to support evolving, threat-based operational missions; programs that need to engage with operational communities and to apply operational knowledge; and in general, programs where front end software is developed, where integration is needed at that front end to support complex planning, situational assessment, and decision making.

3.2 Value of Cognitive Engineering for DARPA

Given DARPA's emphasis on future problem sets, cognitive engineering methods can help project into the future relevant aspects of contextual understanding from current mission sets for first-of-a-kind systems. The Hallmark example provided below illustrates one application for a continuing mission which is in current evolution. However, for the first-of-a-kind systems that many offices of DARPA undertake, gathering and analyzing data about mission needs, the landscape of roles and organizations involved, and challenges that will persist into the future can help inform designs and capability towards user acceptance. The ONR (Office of Naval Research) AACUS program is one example of using cognitive engineering to triangulate across related missions, knowledge required to accomplish mission-related tasks, and user context to inform design [13]. An understanding of the missions and decision spaces, which fundamentally will not change over time, can provide a north star for program and performer staff to anchor on.

3.3 Case Study: DARPA Hallmark

The following section presents a case study of how cognitive engineering methods to gather, analyze, design, and evaluate were employed in the DARPA Hallmark program. With Hallmark, DARPA sought to provide breakthrough capabilities for SEC2. US military capabilities are dependent upon space services for communications, intelligence, navigation, weather data, and more. Physical impact threats combine with growing adversary space control capabilities to present a national security mandate for improving SEC2 capabilities. Command and Control encompasses all aspects of how we organize and carry out our space mission set, but essentially is a human endeavor involving situational awareness, planning, decision making, and assessment. Hallmark took an approach to understand, design for, and explicitly measure the human cognitive aspects of SEC2, including interaction with technology, team processes, and coordination elements.

> ***As we developed the execution strategy for the Hallmark program, the concepts of Zero-Integrator evolved out of our previous acquisition experience, along with inspiration from mobile application development ecosystems, agile development, and cognitive engineering. To begin with, Hallmark aims to develop far future capabilities. How do we build something in the next few years that's still relevant twenty-five years from now? We can emphasize human effectiveness, flexibility, and diversity over set requirements or any specific set of technologies.*** *[61]*

In addition to SEC2 research and development, Hallmark also experimented with a new model of acquisition called Zero-Integrator. Zero-Integrator is an acquisition model where no single performer is appointed program integrator, with measures included to ensure that no single performer dominates integration activities. For Hallmark, these measures included rapid 3 to 4 month evaluation cycles, competition in the form of two independent testbeds and cognitive evaluators, a strong readiness to cancel any performer (or the entire program) at any time, and fixed-schedule, fixed-cost contracts.

Hallmark evolved over three phases. Phase 0 was conducted as a proof-of-concept. A single team of engineers and game designers built an Integrated Software Environment (ISE), both the underlying testbed and the user interface, and a single team of cognitive engineers designed and implemented assessments of how well the ISE supported the cognitive work of SEC2. Hallmark also established its evaluation process of week-long Evaluation Events (EEs), run on a simulated Operations Floor, staffed by actual space operators, conducted every few months.

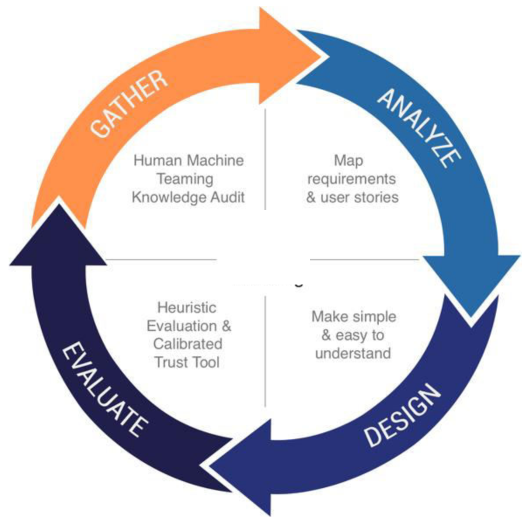
Based on the results of Phase 0, Hallmark was officially approved and performers were selected based on proposals submitted through DARPA's Broad Agency Announcement/Request for Proposals (BAA/RFP) process. While the evaluation process remained the same for Phase 1, the Hallmark software environment was now built from an integration of 10 separate tools. Furthermore, Hallmark selected two performers to build independent testbed environments. Every three-month evaluation cycle now consisted of two back-to-back EEs, one for each testbed. One cognitive evaluation performer was permanently paired with each testbed performer for the entirety of Phase 1, but each tool performer had to successfully run within each testbed environment every three months.

Finally, in Phase 2, additional tools were selected from a new round of proposals and some performers from Phase 1 no longer participated. The new tools had to quickly integrate into the current evolution of the software environment. Both testbeds remained, and each EE was now evaluated by a single cognitive evaluator.

The structure of each phase is important because it influenced what methods were used, when they were used, and how effective they were. Even though the cognitive engineering framework here is presented as an iterative cycle, it's not meant to be a strict script. The framework may be adapted to the specifics of a particular program while still reaping the benefits of each phase.

3.3.1 Gather Methods & Strategies

Although the Gather phase appears to be the obvious place to begin the cognitive engineering cycle, note that there's actually no indicated starting point in the figure. In fact, Phase 0 of Hallmark began in the Design phase, with the engineering team implementing an initial design based on their own previous space experience. This allowed different teams from the program to kick off efforts in parallel and more quickly execute the first evaluation event.



3.3.1.1 Site Visits

In Phase 0, the Hallmark team conducted a number of site visits to relevant space operations floors and other operational command and control organizations, including North American Aerospace Defense Command (NORAD)/United States Northern Command (NORTHCOM), Special Operations Command (SOCOM), and the Joint Space Operations Center (JSpOC). These were mostly observational, but we were also able to conduct a separate visit to the JSpOC in support of a different project and conducted structured cognitive engineering interviews and analysis at that time.

Active: Phase 0

3.3.1.2 SME Interviews

At the start of Phase 1, the Northrop Grumman Corporation (NGC) cognitive evaluation team also conducted a series of initial, foundational SME interviews using three Space Ops SMEs who were themselves NGC employees. The SMEs' experience ranged from U.S. Air Force's Defense Meteorological Satellite Program (DMSP) operations, to GPS constellation operations, to directing multi-mission space operations centers for government agencies [51].

Active: Phase 1

3.3.1.3 Observation

A common problem applying gather methods is substantive access to "real" users and experts, especially in specialized domains. For this reason, Hallmark structured opportunities to collect information in conjunction with a keystone of the evaluation phase, the Evaluation Events (EEs). Each EE provided an entire week of interaction with space operators of varying levels of experience. We observed operators interaction with the ISE over the course of the week, noting behaviors, problems, and comments.

Active: Phase 0, Phase 1, Phase 2

3.3.1.4 After Action Reviews

A critical gather method used across all Hallmark phases was the After Action Review (AAR). About 30 minutes at the end of each day, as well as the entire morning of the final day of the week, was devoted to an open discussion with the operators and the entire Hallmark team about their impressions of the system, the event, what they need to perform their day-to-day jobs, and what they envision they would need to perform their jobs in the future. While AARs did not strictly adhere to any particular gather tool, cognitive engineers employed elements of gather and design methods presented in Section 2 to guide discussion. An example of some questions and methods used include [52]:

- What were major sources of uncertainty today? What was difficult? What was unexpected?
- In terms of communication, what worked and what didn't? Have each role create a Wagon Wheel – "Write your role on a piece of paper and circle it. Draw spokes to your main collaborators and your main sources of information."
- Walk through several of the critical/trigger events in the scenario. Assess how the ISE helped or hindered understanding of what was going on, plans to respond, and briefing upwards to the Decision Maker.

Active: Phase 0, Phase 1, Phase 2

3.3.1.5 Participatory Design

In both Phase 0 and Phase 1, we also brought together operators, other space SMEs, designers and tool developers for one Participatory Design event in addition to the evaluation events. Participatory Design is explained in more detail in Section 2. Even though the events focused on collaboratively designing solutions for specific problems in SEC2, greater understanding of the domain naturally emerges along the way.

Active: Phase 0, Phase 1

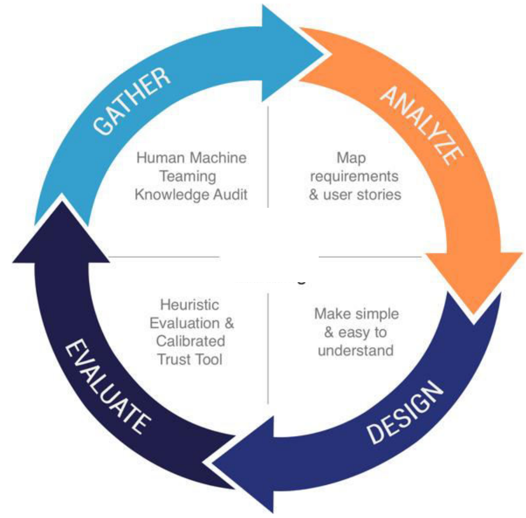
3.3.1.6 Embedded Collaboration

Hallmark benefitted greatly by including space SMEs as members of the core government/contractor team. By being involved in all aspects of the Hallmark, they could provide immediate guidance and domain expertise during planning, design, execution, and evaluation, and act as proxies at times when access to other SMEs was limited.

Active: Phase 0, Phase 1, Phase 2

3.3.2 Analyze Methods & Strategies

With Hallmark's Zero-Integrator approach, Analyze methods had to generate analysis, or at the very least, intermediate findings, very quickly. A major focus of each EE was documenting the cognitive and decision support needs of SEC2 operators and feeding those back to tool developers to improve existing features or add new features.



3.3.2.1 Documenting Operator Capabilities/User Stories

Throughout all phases, cognitive engineering teams compiled observations and feedback during each EE into findings and insights about SEC2 operator needs and how well or how poorly the Hallmark environment currently supported those needs. As the program progressed, more collaborative effort was made to organize these findings into broader categories of SEC2 workflow. For instance, towards the end of Phase 1, NG and General Dynamics worked together to come up with a set of 14 unique "Operator Capabilities" that Hallmark needed to accommodate. Examples include: "Determine orbital change of a satellite based on a maneuver", and "Explore multiple COAs (Course of Action) to respond to anomalous RSO (Resident Space Object) behavior, including time constraints and implications of the different COAs."

In Phase 2, we expanded this into a product we called the Hallmark Usability Catalog (Table 3-1) [53]. The catalog organizes all issues and observations considered unaddressed at the end of Phase 1 and groups them into an even larger set of user stories (Section 2). In theory, these user stories would eventually cover everything that a SEC2 operator needs to do their job. The table below shows an example of a few key columns from the catalog spreadsheet. For instance, the Issue column groups together raw observations and quotes from operators into a common user story (the highlighted sentence that begins each cell). The Functional Area column groups the user stories according to relevant areas of the Hallmark workflow, and the Priority column is a snapshot of how critical addressing the needs illustrated by the user story would contribute to operators' acceptance of the Hallmark system. Because the catalog also tracks each individual source that contributes to a user story, researchers or developers can also see which issues are mentioned most and thus may be most critical to fix.

Table 3-1: Example user stories from the Hallmark Usability Catalog.

Functional Area	Priority	Issue
Data	High	**As an operator, I want insight into the freshness of data input into algorithms/tools**. • Operators were unable to determine how fresh the data was that ReCOAT was using (and whether or not the results are still valid). • Not clear if the TLE data is stale or not. • Operators were not convinced that the information in the Baseball Cards was dynamic and updated automatically (which they wanted it to do). • Intel Lead needs to know what data is being used to create the orbit in Solar... He is concerned that it might be based on old data and therefore he might be making decisions with stale data. • Planner was asking if additional observations were necessary on asset. • It was not clear to the team which TLE was used for the conjunction analysis, given that it had maneuvered recently they were concerned about staleness. • Operators would like to integrate the insights gained since the last time the algorithm was run into a new run. • Space defense has to continually update ReCOAT for changes. • BOBCAT data had all same time stamp and same name
COP	Medium	**As an operator, I want to draw a watch box on the COP to alert me on events in areas of interest**. • Users wanted to be able to put a "defensive bubble" around a particular space object (or, perhaps, a group of them). • Operators were looking for a “watch box” on the Solar COP to tip events that appear in the Astro app. • Confusion on where the impact "location" is going to be
Knowledge Management	Medium	**As an operator, I want to intuitively subscribe to updates about certain Space Objects (RSOs) or other topics of interest**. • It was not obvious on how to subscribe to specific RSO's or topics of interests. During Phase I it was also reported as difficult to subscribe.

Active: Phase 0, Phase 1, Phase 2

3.3.2.2 Documenting Behavior

In Phase 0, Hallmark initially focused on documenting activity and behaviors; for instance, how do operators communicate and collaborate across the Ops Floor? How does shared situational awareness evolve over time, adhere to expected norms, and respond to new technology? Because we were limited

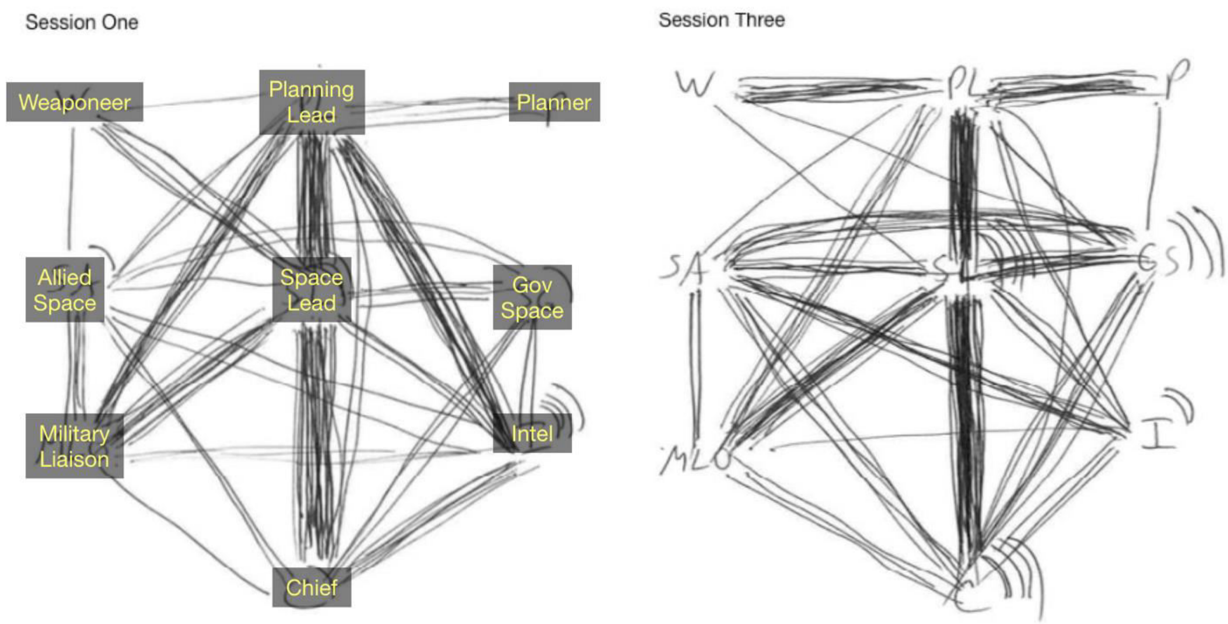


Figure 3-1: Diagram of approximate communication frequency between roles.

in the types of data we could collect, we often had to be flexible and creative in choosing data for our analysis.

For an early EE, one cognitive observer spent all week manually marking on a hand-drawn floor diagram whenever an operator spoke to another or to the group, either on- or off-headset, which showed a general view of communication frequency between roles. These diagrams highlight the consistently strong verbal communications between the chief (C), space lead (SL), and planning lead (PL), indicated via the strong central line. The planner, shown in the upper right of Figure 3-1 as P, is relatively isolated from the rest of the team as far as verbal communications. As another technique, we also asked the operators during the final AAR of the week to draw their own diagram of which team members they communicated with most.

Using Phase 0's ISE event log, we could also look into patterns of behavior regarding individual tools. For example, we generated heatmaps (Figure 3-2) showing use of the ISE's chat tool by role. In the Phase 0 EE2, the heat maps showed that the Space Lead and the Planner rarely used chat to collaborate with the rest of the Ops Floor.

One of the most useful analysis tools we created in Phase 0 was a master timeline of simulation events and observer notes, also using the ISE's event log. After the second EE, for instance, we parsed all events related to communication into a single spreadsheet, along with our own transcribed notes, sorted by simulation time. This effectively overlaid ground truth and operator behavior such that we could quickly follow the progression of key events. We could also use the spreadsheet's built-in sorting and filtering features to look at activity across specific roles and tools. For example, after Phase 0 EE2 we used the Master Timeline to perform a detailed information flow analysis on one 23-minute section of the

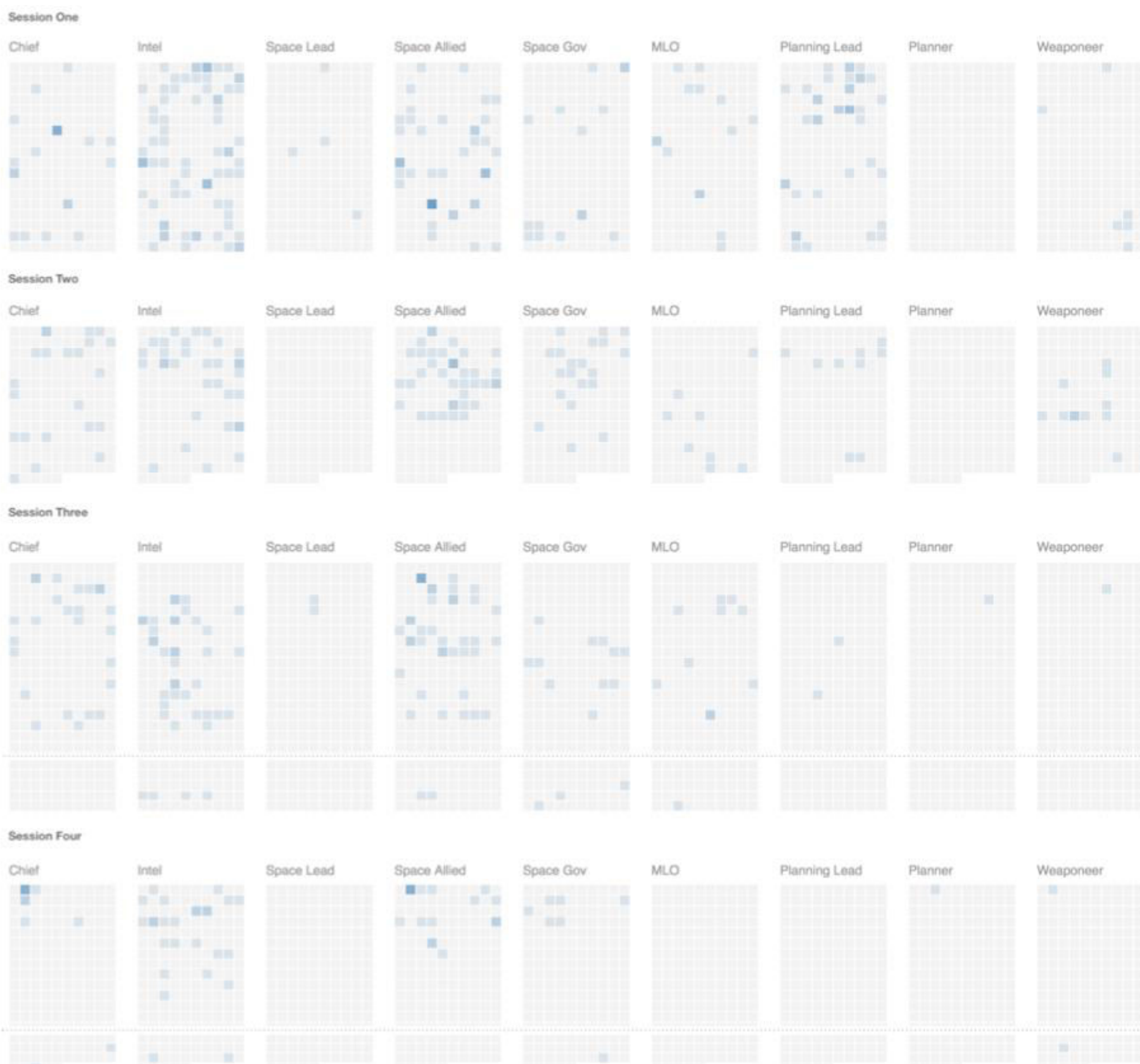


Figure 3-2: Visualization of chat messages sent per role, per session.

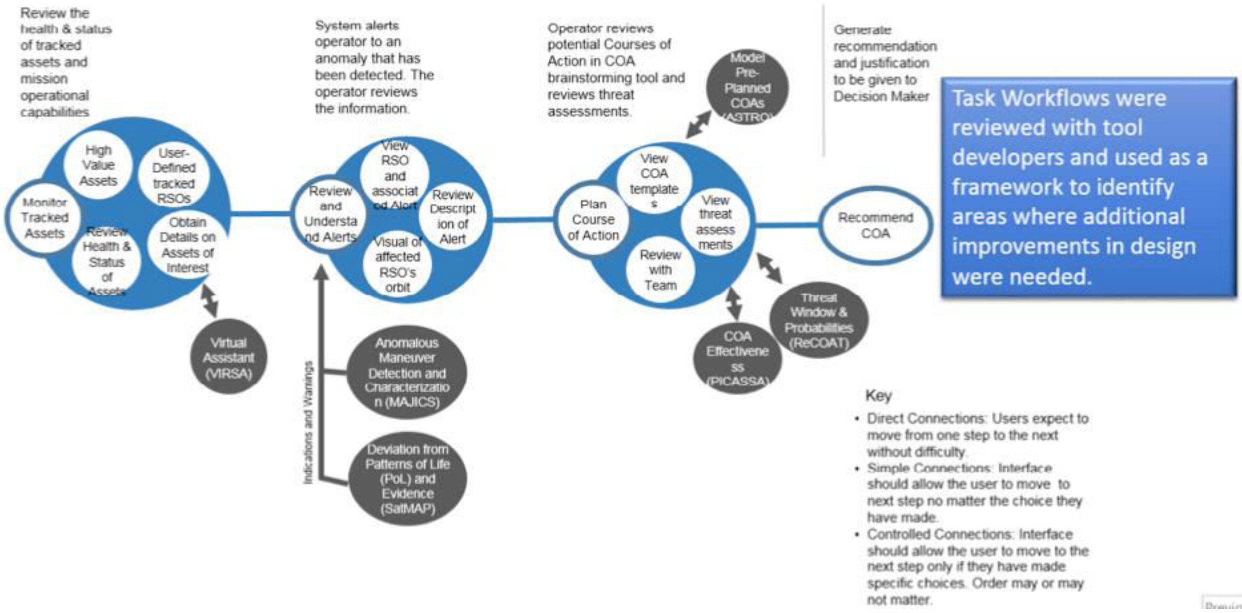


Figure 3-3: NGC's Task workflow for Hallmark's SOLAR tool.

operational scenario, experimenting with a method called CARS-T (which considers Content, Actors/Action, Resources, Situation, and Timeliness [52].

Active: Phase 0, Phase 1

3.3.2.3 Documenting Workflows

After working with the amalgamated Hallmark system of Phase 1, it became apparent that more effort was needed to map out workflows of how operators perform within SEC2. Tool developers were having difficulty fitting their tools together into a seamless environment, and operators were having difficulty discovering what they should do next to perform their jobs when navigating through the system.

Cognitive engineering has created many different techniques for documenting workflows (see Section 2.1). In Phase 1, Hallmark developed 2 levels of workflow models. The first level was a task workflow of a SEC2 operator, including a mapping of individual Hallmark tools onto those tasks (Figure 3-3) NGC constructed and delivered this workflow based on data gathered during operator interviews.

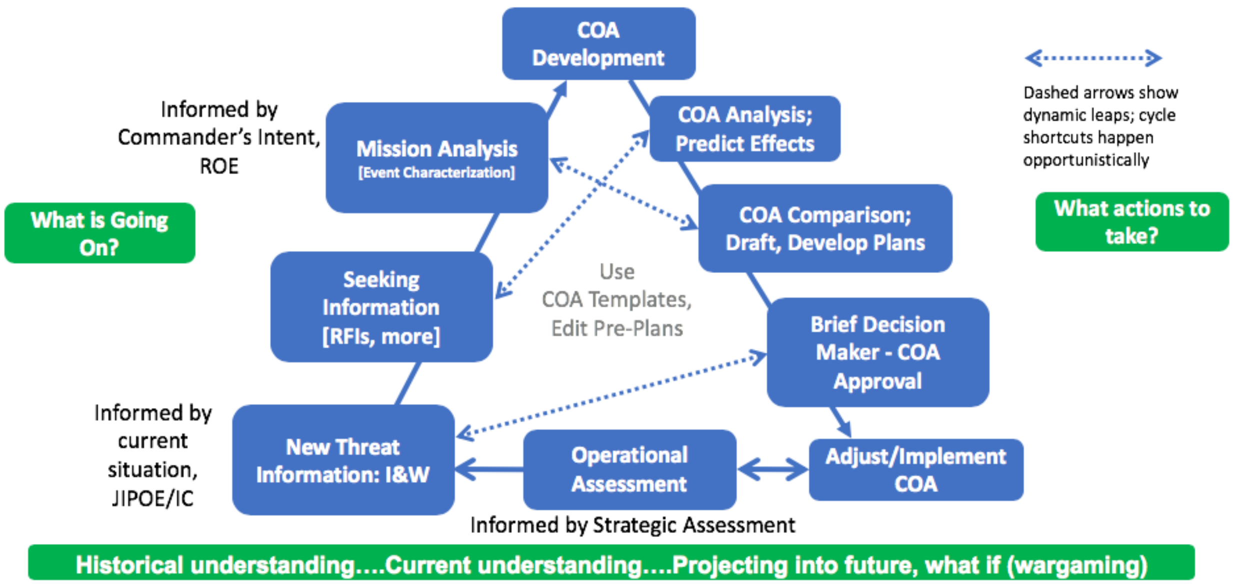
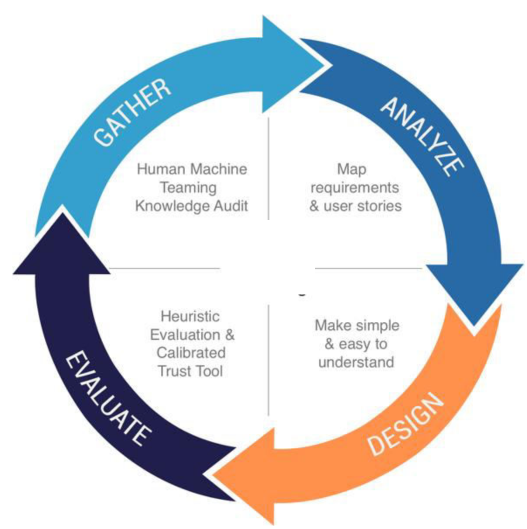


Figure 3-4: Space Enterprise Command and Control cognitive workflow.

The second level was a cognitive workflow modeling the major steps and decisions that make up SEC2, based on the Decision Ladder method [54] (Figure 3-4). Workflow models such as these helped guide discussion about how Hallmark system could best support the operators, including what features needed to be prioritized, which features may be missing, which areas training should focus on, and which areas were the best opportunities for human-machine teaming.

Active: Phase 1

3.3.3 Design Methods & Strategy



As in many programs, the cognitive engineering teams on Hallmark were officially tasked with evaluating effectiveness of operator support, but had no direct control over the design of the tools themselves. In fact, in Phase 2, a usability working group was formed that was tasked to improve usability, but still without a direct hand in design (other than a few members of the working group who were from the tool teams). Other than direct feedback, the main method used to influence the design of Hallmark was through the use of Participatory Design.

3.3.3.1 Participatory Design

In both Phase 0 and Phase 1, MITRE organized participatory design events (Section 2) to help facilitate team collaboration and ideation. The power of participatory design is that it brings together operational experts, cognitive engineers, designers, developers, program staff and systems engineers to work synergistically toward advanced solutions. To use the second participatory design event as an example, eight facilitators from MITRE engaged with seventeen SMEs, twenty-one tool providers, and seven members of the testbed and cognitive evaluation teams across three sites (Bedford, MA, McLean, VA, and Colorado Springs, CO). The design concepts developed in this trio of PD workshops provided insight

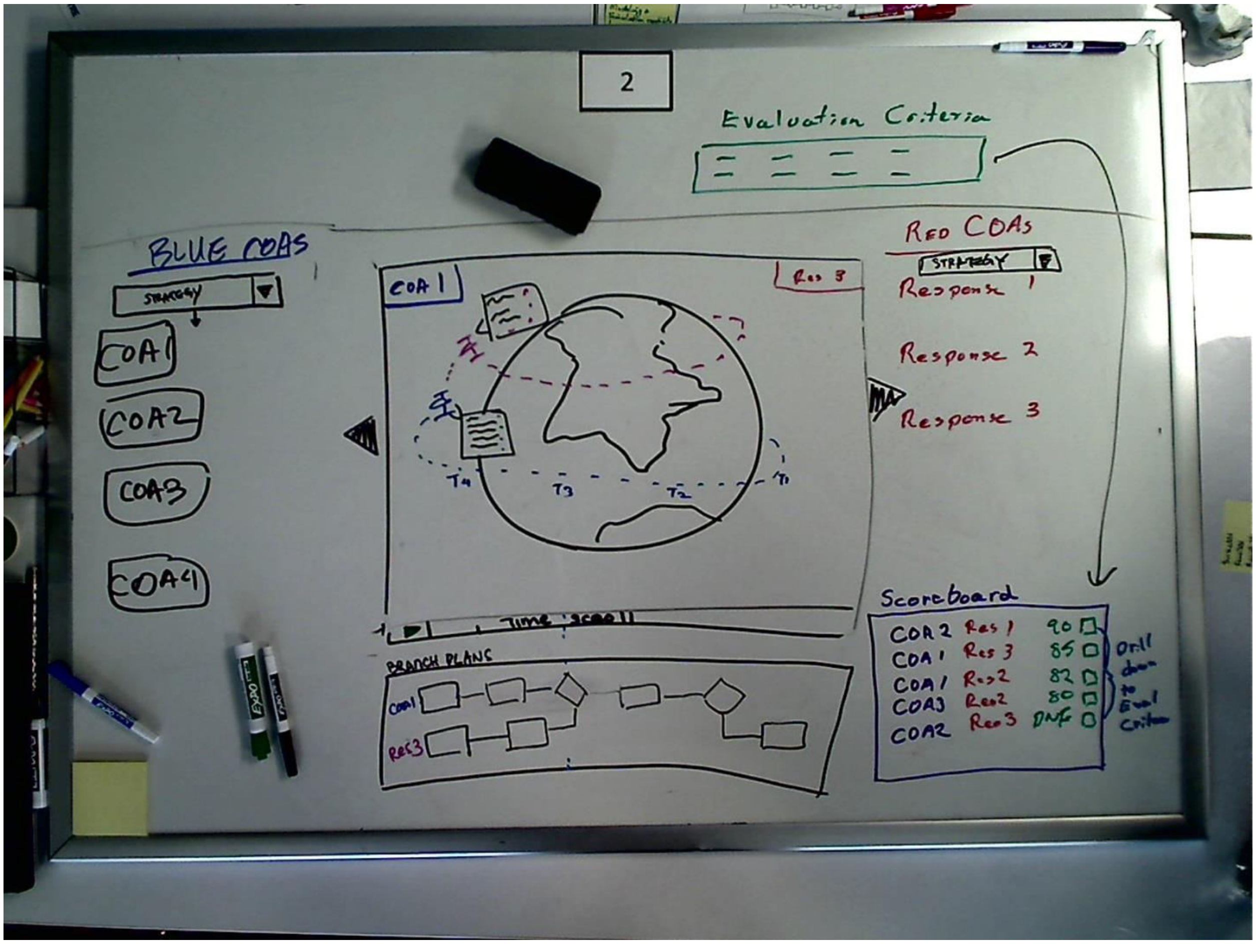


Figure 3-5: Design concept for monitoring courses of action in Space Enterprise Command and Control.

into how to enhance a SEC2 operator's decision-making capabilities. Several themes emerged when considering the full spectrum of designs generated:

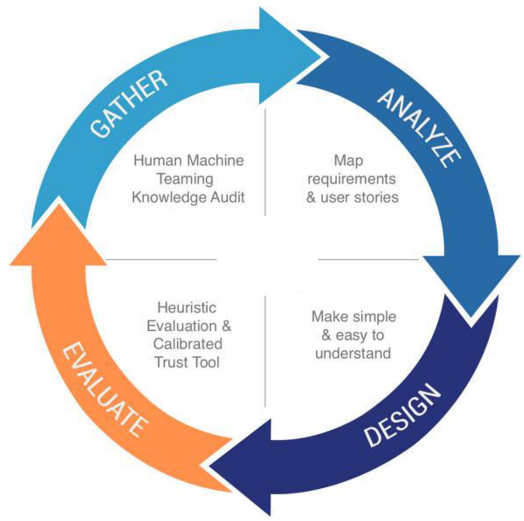
- Present information in context and with the user's perspective in mind. Context can be the event in which planning is occurring, where a task fits in a workflow, the confidence associated with information, or the situation in which an alert occurred.
- Utilize machine learning or advanced modeling and simulation to understand the past and predict future states.
- Transform information to communicate the impacts and implications of actions and events.
- Utilize a drill-down capability to help users manage large amounts of data.

The themes highlighted important functionality needed for SEC2 while the design concepts themselves provided insight into ways that functionality could be operationalized (Figure 3-5). The result was not a "to do" list assigned to organizations, but insights that any developer could use for the betterment of Hallmark tools [55].

As part of our usability working group, we were also able to organize a smaller one-day session during Phase 2. The session brought together Hallmark's internal space SMEs, cognitive evaluation teams, and select tool developers to specifically prioritize ways to improve usability for Hallmark's final evaluation event.

Active: Phase 0, Phase 1, Phase 2

3.3.4 Evaluate Methods & Strategy



Evaluation of system effectiveness often goes hand in hand with analysis. Based on user stories, cognitive, task and workflow models, observations, and experience, cognitive engineers can pinpoint areas where your system fails to support cognitive work, or conversely, supports cognitive work extremely well. However, there are also many options for applying metric-based evaluation. Typical issues cognitive engineering can measure include situation awareness [31], effectiveness and efficiency of task performance, and team collaboration. Based on a literature review in Phase 0, we initially selected Situational Understanding, Decision and Information Confidence, and Timeliness as the areas that would provide insight into Hallmark's effectiveness. After the first evaluation event, effectiveness and efficiency of Information Flow were also added.

3.3.4.1 Survey Methods

A common technique for the evaluate phase is asking users to take surveys designed to measure cognitive performance. Surveys may be applied at any time throughout or at the end of an evaluation. For instance, to test situation awareness, the Situational Awareness Global Assessment Technique (SAGAT [56]) can be administered during a pause in the simulation. Once participants have responded, the simulation is resumed.

In Phases 1 and 2, the NG cognitive evaluation team employed a battery of surveys that directly or indirectly provided evidence for system effectiveness, including:

- Situation Awareness: Situational understanding survey, adding SAGAT [56] in phase 2.
- Mental Workload: NASA Task Load Index (TLX) [57], Mental Resource Availability.
- Trust: Empirically Derived Trust in Automation Scale [58], Modified Madsen & Gregor Human-Computer Trust Scale [59].
- Usability: System Usability Scale (SUS) [60].

Active: Phase 1, Phase 2

3.3.4.2 Activity Tracking Methods

The quantity of different types of user interaction with a system can sometimes be used as evidence of cognitive performance. As discussed above in Section 3.3.2.2, in Phase 0 we looked at tracking activity including the amount of dialogue between operator roles and the number of times the chat window was clicked on. We also examined the quantity of products such as the number of COAs created or the number of RSO analyses requested. Although it is possible to be constrained by what is available in software system logs, ideally developers can code keystroke/user interaction tracking into a system to track the most useful activity.

Active: Phase 0, Phase 1, Phase 2

3.3.4.3 Product Quality Methods

In order to evaluate Situational Understanding, Hallmark used Space SMEs to assess the quality of the products created by operators during the evaluation events. For instance, a major part of SEC2 is briefing decision makers on the Ops Floor analysis of the current situation and recommended COAs for addressing mission goals. We used SMEs to check these briefings and COAs to see if the floor understood and addressed key events in the evaluation scenario, and if their recommended COAs were appropriate.

Active: Phase 0, Phase 1, Phase 2

3.3.4.4 Heuristic Methods

At the start Phase 1 and 2, the NG team applied the UX Product Scorecard heuristic method to evaluate and identify usability issues with the Hallmark system prior to presenting the system to real operators. While usability is not the same as cognitive engineering, we've confirmed in Hallmark that systems needs a basic level of well-designed usability before users will be able to provide feedback and insight into deeper issues.

Active: Phase 1, Phase 2

3.3.4.5 Physiological Methods

In Phase 1, the NG team also captured physiological data from key operator roles to augment self-reported mental workload via surveys. An Empatica E4 wristband recorded pulse and heartrate and a Gazepoint GP3 HD eye tracker recorded gaze duration and frequency, as well as changes in pupil dilation. While there were some interesting differences to pursue further in self-reported workload and algorithmically calculated workload, NG discontinued the capture of eye tracker data in Phase 2 because the methods themselves were still largely supplemental to Hallmark's core goal of agile discovery and addressing key issues to better support SEC2.

Active: Phase 1

3.3.4.6 Experimental Design

While Phase 0 and Phase 1 were designed to reflect the application of cognitive engineering to large-scale simulation evaluations or complex natural environments, in Phase 2 the DARPA program manager also wished to explore a more controlled, experimental approach. The cognitive evaluation team thus designed several tool manipulations that could be controlled and examined within the agile evaluation approach that Hallmark was already employing.

Experiment manipulations included displaying vs. hiding the confidence values provided by analytic tools when they alerted the operator (e.g., "a conjunction between two RSOs is predicted" vs "a conjunction between two RSOs is predicted with medium confidence"), or proactively generating and pushing potential COAs to operators vs operators manually requesting potential COAs when desired. Evaluators would then compare whether the manipulations had any effect on operators' performance in maintaining situational awareness or developing quality COAs.

Even though the evaluation environment remained dynamic and complex, two aspects helped with the experimental design. First, Hallmark had already planned to keep the operational scenarios that the operators work with the same across all evaluation events in Phase 2 (in Phase 0 and Phase 1, Hallmark developed three or four new scenarios for each event). Second, because each event occurred across two weeks with two different testbeds, the cognitive evaluation team could flexibly design manipulations across weeks as well as across individual scenarios.

Active: Phase 2

3.3.5 Institutional Review Boards

When human participants are utilized in any type of research, it is likely to require review and approval from an Institutional Review Board (IRB). The IRB will review the research design for the ethical treatment of participants. Some factors that contribute to their decision include physical, psychological, and professional standards for protecting participants rights. In addition, IRB's review the type of data collected, how those data are analyzed, how the participant's privacy is ensured (e.g., storage of personally identifying information), as well as the duration of the research experiment. Other examples of IRB review criteria can include the amount of nausea or dizziness which might result from a virtual or augmented reality study.

The time it takes for an IRB to review and approve a research study varies based on the complexity of the research design. Be sure to account for this time when embarking on a new study.

3.3.6 Lessons Learned

What follows are a few major lessons learned from applying cognitive engineering to the Zero-Integrator design of the Hallmark program. While the Hallmark environment was unique, individual elements are commonly found in other programs, such as integrating sets of individual tools, competition between performers, and agile software development.

3.3.6.1 Begin with a Baseline Operational Workflow Model

When innovative tools are sought from multiple companies, or, in legacy systems where multiple tools accumulate over time, there are implications for integrating disparate tools and technologies, from both a systems engineering and UX perspective. Hallmark attempted both, mixing new and legacy tools into a single integrated SEC2 system.

Thus, establishing a baseline SEC2 Operational workflow early in Phase 0 that, although evolving throughout, continued across Phase 1 and Phase 2 would have better guided much of the Hallmark program, including source selection, training, tool integration, and cognitive evaluation feedback.

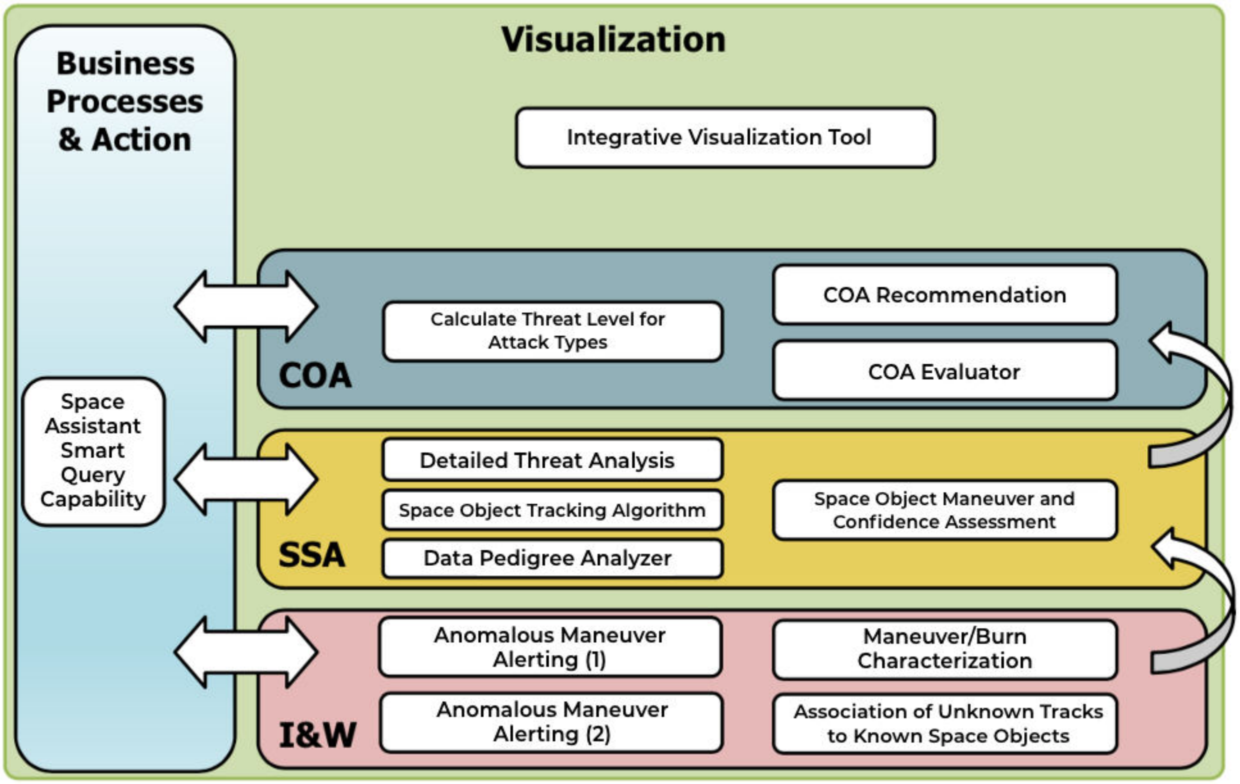


Figure 3-6: Categorization of Phase 2 tools according to Space Enterprise Command and Control support.

Phase 1 began by categorizing selected tools into layers of categories, in a diagram DARPA called the "wedding cake" (Figure 3-6). However, these broad groupings didn't delve deeply enough into operator work. They didn't capture the specific tasks that DARPA needed to apply research and development towards, and, once selected, they didn't inform performers how their tools should support the tasks of an SEC2 Ops Floor. For instance, collaboration, communication and knowledge management are very critical to the SEC2 process, but neither do they suggest advanced space research. No one bid any means for teams to collaborate and these tasks ended up a secondary responsibility of the testbeds using off-the-shelf voice, chat, and wiki tools, which were inadequate for the complexities of shared space situational awareness without tighter integration into the Hallmark system.

Figure 3-7 shows an example of a more detailed cognitive workflow model. Examples of more detailed SEC2 workflow models developed later in Phase 1 are presented in Section 3.3.2.3. As a lesson learned, Hallmark should have used Phase 0 to help develop an SEC2 model closer to this level of analysis, which then could have been used to shape the program BAA (Broad Agency Announcement), identify gaps in task coverage, and so on.

3.3.6.2 Establish a Strong Foundation of System Stability and Usability

Without a strong foundation of stability and usability, getting to the deep cognitive engineering issues of interest can be difficult. With the rapid integration and iteration of Hallmark, we found that basic UX features, e.g., sorting tables or improving system-wide search, would often be postponed to a future sprint (and then another future sprint) in favor of another advanced feature or urgent fix. Operators would then spend a large portion of valuable feedback sessions talking about sorting and search rather than SEC2.

UX is dependent not only on front-end features, but also on the performance and stability of the testbed and individual tools that drive the front-end. It only takes one tool that drags response time or delivers a strange answer to wreck an operator's trust and increase frustration with the entire system.

Dedicate several initial sprints to resolving basic system and usability features before bringing in experienced users for evaluation. Even something as simple as establishing a common look-and-feel can improve the basic usability of an integrated system.

3.3.6.3 Structure Competitive Programs to Enable Knowledge Sharing

Competition was a core component of the Hallmark approach, but it's important that fundamental domain knowledge is shared with all performers, even in a competitive environment. In Phase 1, Hallmark selected two cognitive evaluation performers, each permanently paired with one of two testbed performers. As a result, sharing cognitive findings was hindered. While each cognitive evaluation team shared their results to each tool performer separately, we could not collaborate and share with the entire Hallmark team equally.

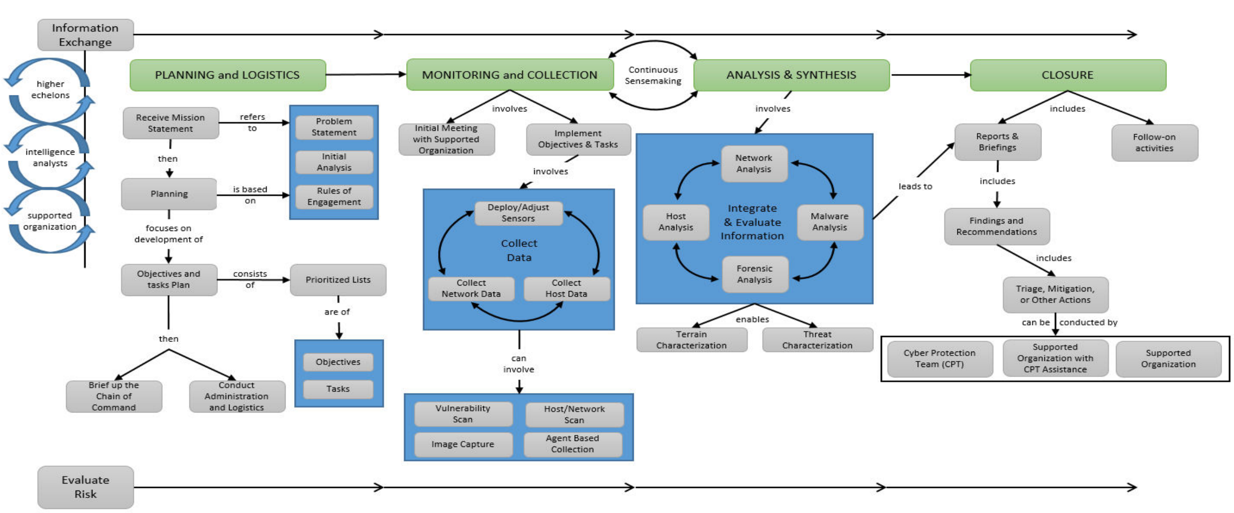


Figure 3-7: A cognitive workflow model of team cyber protection [20].

For instance, the cognitive engineering team that conducted the SME interviews discussed in Section 3.3.1.2 could not share them with the competing testbed. The team that developed the task model shown in Figure 3-3 could not compare, discuss, or debate the model with the space SMEs from the other cognitive engineering team.

3.3.6.4 Ensure Evaluation Findings Inform the Next Sprint

For agile projects, it's critical to scope processes of data collection, analysis, and feedback such that cognitive evaluation findings directly inform the next sprint. Otherwise momentum is lost and operators find themselves rehashing known issues.

Hallmark struggled with these timelines within our three- to four-month evaluation cycles. Even though the cognitive evaluation teams gamely set up admirable processes for turning around feedback to tool providers quickly, the truth is, that pace is really hard. As a member of one of the cognitive evaluation teams responded when asked What has been difficult or challenging about the Zero-Integrator model?: "I wouldn't want to support 90-day cycles for an entire five-year program" [61].

One technique that was particularly helpful throughout the program was allowing the tool providers to directly participate in the final end-of-the-evaluation-week After-Action Reviews (Section 3.3.1.4). This allowed software developers to hear firsthand what operators thought about the tools and experience the cognitive engineers' process for eliciting feedback and encouraging design thinking that helped influence sprint schedule and prioritization.

As another recommendation, build in an "analysis freeze" at the beginning of the sprints to allow for extra time to allow cognitive engineers to analyze data and discuss priorities before tool providers establish their own development path. Working in parallel efforts can slow the iterative process and runs the risk of each team taking divergent paths.

In addition to improving timing, feedback also needs to be at a level that is actionable by the developers. Quantitative metrics like SUS or SAGAT surveys (Section 3.3.4.1) need to be presented in context, not just scores and trends, but an analysis of why and what might be done to fix or improve. Feedback should not dictate a particular design, but provide flexibility for developers to come up with new and creative solutions to the underlying issue. We believe that the user stories compiled in the Hallmark Usability Catalog (Table 3-1) represent one appropriate level of feedback. Not only do they enable flexible solutions, they also remain relevant in guiding the development of other SEC2 systems in the future.

At the same time, focusing purely on cognitive evaluation without also applying the appropriate UX resources to follow up on feedback will not lead to ideal solutions. Possible resolutions include dedicating a performer exclusively to system UX, along with some authority to direct design and feature prioritization, or requiring that performers include experienced UX or cognitive engineers as part of their teams.

4 Summary

This cognitive engineering toolkit has documented the appropriate use, benefits, methods, and a detailed exemplar showing how a cycle of gather, analyze, design, and evaluate methods can be applied across the systems engineering lifecycle.

Although each project or program whose goal it is to support complex cognitive work must tailor the methods selected and how they are applied to their own unique circumstance, we argue that the lack of application of these techniques has resulted in misspent resources and effort as well as user rejection of technology solutions. The systematic application of user-facing gather, analyze, design, and evaluation methods, as appropriate, can result in connecting the mission-driven technology needs of users tightly with the systems developed, as validated by evaluation results of each application. This toolkit is intended to be a living document, and we welcome additional feedback to improve and revise the toolkit as we continue to evolve the practice of cognitive engineering.

5 References

[1] DoD Defense Science Board, Design and acquisition of software for defense systems, Washington, DC: Office of the Under Secretary of Defense for Research and Engineering, 2018.

[2] E. Tenner, Why things bite back: Technology and the revenge of unintended consequences, New York: Alfred A. Knopf, 1996.

[3] L. G. Militello, C. O. Dominguez, G. Lintern and G. Klein, "The role of cognitive systems engineering in the systems engineering design process," *Systems Engineering,* vol. 13, pp. 261-273, 2009.

[4] G. Klein, K. Ross, B. Moon, D. Klein, R. Hoffman and E. Hollnagel, "Macrocognition," *IEEE Intelligent Systems,* vol. 18, no. 3, pp. 81-85, 2003.

[5] J. Orasanu and T. Connolly, "The reinvention of decision making," in *Decision making in Action: Models and Methods*, G. A. Klein, J. Orasanu, R. Calderwood and C. E. Zsambok, Eds., Westport, CT, Ablex Publishing, 1993.

[6] G. A. Klein, J. Orasanu, R. Calderwood and C. E. Zsambok, Decision making in action: Models and methods, Norwood, NJ: Ablex, 1993.

[7] J. Rasmussen, A. Pejtersen and L. Goodstein, Cognitive Systems Engineering, New York: Wiley-Interscience, 1994.

[8] E. M. Roth, E. S. Patterson and R. J. Mumaw, "Cognitive Engineering: Issues in User-Centered System Design," in *Encyclopedia of Software Engineering, 2nd Edition*, New York, Wiley-Interscience, John Wiley & Sons, 2002, p. 163 – 179.

[9] N. J. Cooke and F. Durso, Stories of Modern Technology Failures and Cognitive Engineering Successes, Boca Raton, FL: CRC Press, 2007.

[10] C. S. C. and G. S. Malecki, "Tactical decision making under stress: History and overview," in *Making Decisions Under Stress: Implications for Individual and Team Training*, J. A. Cannon-Bowers and E. Salas, Eds., Washington, D. C., American Psychological Association, pp. 3-15.

[11] J. H. Johnston, J. A. Cannon-Bowers and E. Salas, ""Tactical Decision Making Under Stress (TADMUS): Mapping a program of research to a real world incident—the USS Vincennes," in *TO HFM Symposium on Collaborative Crew Performance in Complex Operational Systems*, Edinburgh, United Kingdom, 1998.

[12] B. Crandall, G. Klein and R. R. Hoffman, Working Minds: A Practitioner's Guide to Cognitive Task Analysis, Cambridge, MA: MIT Press, 2006.

[13] C. Dominguez, R. Strouse, L. Papautsky and B. Moon, "Cognitive Design of an Application Enabling Remote Bases to Receive Unmanned Helicopter Resupply," *Journal of Human-Robot Interaction,* vol. 4, no. 2, pp. 50-60, 2015.

[14] J. Klamm, C. Dominguez, B. Yost, P. McDermott and M. Lenox, "Partnering with technology: The importance of human machine teaming in future MDC2 systems," in *SPIE Defense and Commercial Sensing Conference*, Baltimore, MD, 2019.

[15] L. G. Militello and R. J. B. Hutton, "Applied Cognitive Task Analysis (ACTA): A practitioner's toolkit for understanding cognitive task demands," *Ergonomics, Special Issue: Task Analysis,* vol. 41, no. 11, pp. 1618-1641, 1998.

[16] L. G. Militello, R. J. B. Hutton, R. M. Pliske, B. J. Knight and G. Klein, "Applied Cognitive Task Analysis (ACTA) Methodology," Klein Associates, Inc., Fairborn, OH, 1997.

[17] P. McDermott, C. Dominguez, N. Kasdaglis, M. Ryan, I. Trahana and A. Nelson, "Human-Machine Teaming Systems Engineering Guide," The MITRE Corporation, Bedford, MA, 2018.

[18] G. Klein, R. Calderwood and D. MacGregor, "Critical decision method of eliciting knowledge," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 19, pp. 462-472, 1998.

[19] R. R. Hoffman, B. Crandall, G. Klein, D. G. Jones and M. R. Endsley, "Protocols for Cognitive Task Analysis," Institute for Human and Machine Cognition, Pensacola, FL, 2008.

[20] S. Trent, R. Hoffman, D. Merritt and S. Smit, "Modelling the cognitive work of cyber protection teams," *The Cyber Defense Review,* vol. 4, no. 1, pp. 125-135, Spring 2019.

[21] B. M. Moon, R. R. Hoffman, J. D. Novak and A. J. Canas, Applied Concept Mapping: Capturing, Analyzing, and Organizing Knowledge, Boca Raton, FL: CRC Press, 2011.

[22] D. Klinger and H. B., "Handbook of TEAM CTA," Human Systems Center, Brooks AFB, 2003.

[23] C. Dominguez, P. Uhlig, J. Brown, O. Gurevich, W. Shumar, G. Stahl, A. Zemel and L. Zipperer, "Studying and supporting collaborative care processes," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Santa Monica, CA, 2005.

[24] R. J. Harder and H. Higley, "Application of thinklets to team cognitive task analysis," in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, Big Island, HI, USA, 2004.

[25] B. Moon, "Concept maps and wagon wheels: merging methods to improve the understanding of team dynamics," in *Proceedings of the First International Conference on Concept Mapping*, Pamplona, Spain, 2004.

[26] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology,* vol. 3, no. 2, pp. 77-101, 2006.

[27] G. Walker, "Verbal protocol analysis," in *Handbook of Human Factors Methods*, N. Stanton, A. Hedge, K. Brookhuis, E. Salas and H. Hendrick, Eds., Boca Raton, FL, CRC Press, 2004.

[28] G. Klein, G. L. Kaempf, S. T. M. Wolf and T. Miller, "Applying Decision Requirements to User-Centered Design," *International Journal of Human Computer Studies,* vol. 46, no. 1, pp. 1-15, 1997.

[29] D. Thompson, S. L. Wiggins and G. Ho, "Using cognitive task analysis to develop scenario-based training for house-clearing teams (No. DRDC-TORONTO-CR-2007-118)," Klein Associates, Inc., Fairborn, OH, 2006.

[30] M. R. Endsley, "The Role of Situation Awarenesss in Naturalistic Decision Making," in *Naturalistic Decision Making*, C. E. Zsambok and G. Klein, Eds., Mahwah, NJ, Lawrnece Erlbaum Associates, 1997.

[31] M. R. Endsley, B. Bolte and D. G. Jones, Designing for Situation Awareness: An Approach to Human-Centered Design, London: Taylor & Francis, 2003.

[32] N. Naikar, Work domain analysis: Concepts, guidelines, and cases, CRC Press, 2013.

[33] J. Rasmussen, "Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models," *IEEE Transactions on Systems, Man, and Cybernetics,* Vols. SMC-13, no. 3, pp. 257-266, 1983.

[34] N. Naikar, R. Hopcroft and A. Moylan, "Work domain analysis: Theoretical concepts and methodology (Technical Report No. DSTO-TR-1665)," Defence Science and Technology Organisation Victoria (Australia) Air Operations Division, 2005.

[35] K. Vicente, Cognitive work analysis: Towards safe, productive, and healthy computer-based work, Mahwah, NJ: Lawrence Erlbaum, 1999.

[36] N. Naikar and P. Sanderson, "Evaluating design proposals for training-system definition with work domain analysis," *Journal of the Human Factors and Ergonomics Society,* vol. 43, pp. 529-542, 2001.

[37] W. C. Elm, J. W. Gualtieri, B. P. McKenna, J. S. Tittle, J. E. Peffer, S. Szymczak and J. Grossman, "Integrating cognitive systems engineering throughout the systems engineering process," *Journal of Cognitive Engineering and Decision Making,* vol. 2, no. 3, p. 249–273, 2008.

[38] Anonymous, "User stories," Mountain Goat Software , [Online]. Available: https://www.mountaingoatsoftware.com/agile/user-stories. [Accessed 19 January 2020].

[39] K. E. Wiegers, "Writing quality requirements," *Software Development,* pp. 44-48, May 1999.

[40] R. Krause, "Storyboards Help Visualize UX Ideas," Nielsen Norman Group, 15 July 2018. [Online]. Available: https://www.nngroup.com/articles/storyboards-visualize-ideas/. [Accessed 18 January 2020].

[41] Anonymous, "Usability Body of Knowledge: Storyboard," User Experience Professionals' Association, [Online]. Available: https://www.usabilitybok.org/storyboard. [Accessed 27 January 2020].

[42] Anonymous, "Participatory Design/Co-design Worksession," OpenInnovation Toolkit, [Online]. Available: https://toolkit.mozilla.org/method/participatory-designco-design-worksession/. [Accessed 19 January 2020].

[43] Anonymous, "Participatory Design Workshop Guidelines," iTEC, [Online]. Available: http://itec.aalto.fi/participatory-design/workshop-guidelines/. [Accessed 19 January 2020].

[44] E. a. S. P. J. Sanders, "Co-creation and the new landscapes of design," *Co-design,* vol. 4, no. 1, pp. 5-18, 2008.

[45] J. Nielsen, "10 Usability Heuristics for User Interface Design.," April 1994. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/. [Accessed January 19 2020].

[46] C. Dominguez, A. Grome, R. Strouse, B. Crandall, C. Nemeth and M. O'Connor, "Linking Cognitive Data to Design in Navy Command and Control," in *INCOSE International Symposium*, San Diego, CA, 2010.

[47] E. L. Papautsky, C. Dominguez, R. Strouse and B. Moon, " Integration of cognitive task analysis and design thinking for autonomous helicopter displays," *Journal of Cognitive Engineering and Decision Making,* vol. 9, no. 4, pp. 283-294, 2015.

[48] J. Brook, "SUS – A quick and dirty usability scale," in *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, B. A. Weerdmeester and A. L. McClelland, Eds., New York, Taylor & Francis, 1993.

[49] B. Crandall, G. Klein and R. R. Hoffman, "Cognitive task analysis for measurement and evaluation," in *Working Minds: A Practitioner's Guide to Cognitive Task Analysis*, B. Crandall, G. Klein and R. R. Hoffman, Eds., Cambridge, MA, MIT Press, 2006, pp. 229-244.

[50] V. J. Gawron, Human Performance, Workload, and Situational Awareness Measures Handbook, Boca Raton, FL: CRC Press, 2008.

[51] N. Ezer, "DARPA Hallmark phase 1: knowledge elicitation process and results in preparation for evaluation event 2," Northrop Grumman Systems Corporation, Falls Church, VA, 2018.

[52] C. Dominguez, "DARPA Hallmark phase zero demonstration 2.0 cognitive engineering report," The MITRE Corporation, McLean, VA, 2017.

[53] C. Johnson, "DARPA Hallmark usability catalog," The MITRE Corporation, McLean, VA, 2019.

[54] R. Hoffman and M. McCloskey, "The macrocognitive decision ladder," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, San Diego, CA, 2013.

[55] P. McDermott, "DARPA Hallmark phase 1 participatory design concepts," The MITRE Corporation, McLean, VA, 2018.

[56] M. Endsley, "Situation awareness global assessment technique (SAGAT)," in *Proceedings of the IEEE 1988 National Aerospace and Electronics Conference*, Dayton, OH, 1988.

[57] S. Hart, "NASA-Task Load Index (NASA-TLX): 20 years later," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, San Francisco, CA, 2006.

[58] J. Jian, A. Bisantz and C. Drury, "Foundations for an empirically determined scale of trust in automated systems," *International Journal of Cognitive Ergonomics,* vol. 4, no. 1, pp. 53-71, 2000.

[59] M. Madsen and S. Gregor, "Measuring human-computer trust," in *Proceedings of the 11th Australasian Conference on Information Systems*, Melbourne, 2006.

[60] J. B. Brooke, "SUS: a retrospective," *Journal of Usability Studies,* vol. 8, no. 2, pp. 29-40, 2013.

[61] J. Raley, F. Barlos and C. Johnson, "Faster, freer, flexible? The zero-integrator experiment," *Defense Acquisition Magazine,* vol. XLVII, no. 4, pp. 33-38, July/August 2019.

[62] H. Beyer and K. Holtzblatt, Contextual Design: Defining Customer-Centered Systems, San Francisco, CA: Morgan Kaufmann Publishers, 1998.

[63] S. L. Wiggins and D. A. Cox, "System evaluation using the cognitive performance indicators," in *Macrocognition Metrics and Scenarios: Design and Evaluation for Real-World Teams*, J. E. Miller and E. S. Patterson, Eds., Boca Raton, FL, CRC Press, pp. 285 - 302.

[64] D. Mayhew, The Usability Engineering Lifecycle, Burlington, MA: Morgan-Kaufmann, 1999.

[65] R. W. Pew and A. S. Mavor, Human-System Integration in the System Development Process, Washington, D. C.: National Academies Press, 2007.

[66] T. Tullis and W. Albert, Measuring the User Experience, 2nd Ed., Burlington, MA: Morgan-Kaufmann, 2013.

[67] P. McDermott, "Calibrated Trust Evaluation Toolkit," The MITRE Corporation, 20 August 2019. [Online]. Available: https://comm.mitre.org/calibrated-trust-toolkit/. [Accessed 9 January 2020].

[68] S. J. Selcon and R. M. Taylor, "Evaluation of the situational awareness rating technique (SART) as a tool for aircrew system design," in *AGARD Conference Proceedings No. 478*, Neuilly-sur-Seine, France, 1989.

[69] ISO, "Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability," ISO 9241-11, 1998. [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=16883. [Accessed 18 December 2019].

[70] J. Miller and E. Patterson, Eds., Macrocognition Metrics and Scenarios: Design and evaluation for real-world teams., London and New York: Routledge, Taylor and Francis Group, 2016.

[71] P. McDermott, C. Dominguez, N. Kasdaglis, M. Ryan, I. Trahana and A. Nelson, "Human-Machine Teaming Systems Engineering Guide," The MITRE Corporation, Bedford, MA, 2018.

[72] S. S. Potter, W. C. Elm and J. S. Tittle, "Evaluating the resilience of a human-computer decision-making team: a methodology for decision-centered testing," in *Proceedings of the 8th International Conference on Naturalistic Decision Making*, Pacific Grove, CA, 2006.

[73] R. Rousseau, J. Easter, W. Elm and S. S. Potter, "Decision-Centered Testing (DCT): Evaluating Joint Computer Cognitive Work," in *Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting*, Orlando, FL, 2005.

[74] M. E. Jobidon and R. Breton, "Initial validation of the decision centered testing methodology," Defence R&D Canada -Valcartier Technical Report No. TR 2008-302, Valcartier, Quebec, 2009.

[75] N. Kasdaglis, "DARPA Hallmark cognitive evaluation transition report (MTR 170233)," The MITRE Corporation, McLean, VA, 2017.

[76] N. Ezer, "DARPA Hallmark phase 1: final technical report," Northrop Grumman Systems Corporation, Falls Church, VA, 2018.

[77] H. Plattner, " An Introduction to Design Thinking PROCESS GUIDE," Stanford d.school, 2010. [Online]. Available: https://dschool-

old.stanford.edu/sandbox/groups/designresources/wiki/36873/attachments/74b3d/ModeGuideBOOTCAMP2010L.pdf. [Accessed 9 January 2020].

This page intentionally left blank.

Appendix A Abbreviations and Acronyms

AAR	After Action Review
ACTA	Applied Cognitive Task Analysis
ADS	Abstraction Decomposition Space
AI	Artificial Intelligence
ATO	Air Tasking Order
BAA	Broad Agency Announcement
C2	Command and Control
CDM	Critical Decision Method
CE	Cognitive Engineering
COA	Course of Action
CONOPS	Concept of Operations
CPI	Cognitive Performance Indicators
CSE	Cognitive Systems Engineering
CWA	Cognitive Work Analysis
DARPA	Defense Advanced Research Projects Agency
DCT	Decision-Centered Testing
DoD	Department of Defense
DRT	Decision Requirements Table
EE	Evaluation Event
GDTA	Goal Directed Task Analysis
GPS	Global Positioning System
HE	Heuristic Evaluation
HITL	Human-In-The-Loop
HMT	Human-Machine Teaming
HMW	How Might We
IRB	Institutional Review Board
ISE	Integrated Software Environment
JSpOC	Joint Space Operations Center
NASA-TLX	NASA Task Load Index
NDM	Naturalistic Decision Making
NGC	Northrop Grumman Corporation

NORAD	North American Aerospace Defense Command
NORTHCOM	Northern Command
ONR	Office of Naval Research
R&D	Research & Development
RFP	Request for Proposals
RSO	Resident Space Object
SA	Situation Awareness
SAGAT	Situation Awareness Global Assessment Technique
SEC2	Space Enterprise Command and Control
SME	Subject Matter Expert
SOCOM	Special Operations Command
SRK	Skills, Rules, Knowledge
SUS	System Usability Scale
UI	User Interface
UX	User Experience
WDA	Work Domain Analysis

Appendix B Additional Toolkit Methods

B.1 Cognitive Performance Indicators Evaluation

A Cognitive Performance Indicators (CPI) evaluation is similar to a Heuristic Evaluation, but it is conducted against a set of 11 cognitive engineering design heuristics called CPIs. The goal is to quickly determine whether a design follows the CPIs and to recommend improvements. It typically involves 3 – 5 reviewers (e.g., SMEs, cognitive engineers, human factors engineers) each individually comparing screens intended to support cognitive work against the CPIs. Findings are then combined across reviewers.

Timeframe: Similar to a Heuristic Evaluation, usually quick to conduct, but will vary with the number and complexity of user interface screens. 2 - 3 hours to conduct the actual evaluation (per evaluator), several days on the front and back end of the evaluation to prepare materials and consolidate and prioritize findings.

Use When: An interactive prototype, storyboards, or screenshots of the system to be tested are available. Use iteratively after each round of development to continually improve the system.

Steps:

- Develop an evaluation plan: Identify the number of evaluators and the time available to complete the evaluation. If the system is large or complex, assign different system components to different evaluators.
- Review background materials: Review background materials (e.g., Concept of Operations (CONOPs) documents, requirements, etc.) to understand the nature of the system, its users, and the envisioned context of use. If time permits, develop a user profile and identify the work users will accomplish with the system.
- Identify the scenarios to be used in the evaluation: Identify scenarios that encompass the most important tasks the system was designed to support. If such scenarios do not already exist, identify prior incidents that were "tough cases" and use those in the evaluation. Tough cases may be identified using a gather technique such as the Critical Decision Method.
- Get a demonstration of the system from developers or users. Use the demonstration to collect information about the intended users of the system, tasks the system is supposed to support, and to answer "what if" questions about the system (e.g., If you only had half the time for this task, what would you do differently?"). Review the background information and demonstration information to select the set of CPIs to use for the evaluation.
- Familiarize evaluators with background, system, and scenarios: Conduct an initial walkthrough of the system with each evaluator (either individually or as a group) with the CPIs in hand. Evaluators should note areas that require further investigation and ask clarifying questions.
- Conduct evaluations individually using the CPIs: In addition to the familiarization pass, each evaluator makes another pass through the system while evaluating it against the CPIs.

Evaluators should work through the scenario if one is available and record potential problems as well as positive aspects of the design.

- Regroup and examine findings: After completing the individual evaluations, regroup and compare and contrast findings. The goal is to generate a list of ways in which the system supports cognitive work and ways in which it hinders it. Use the affinity process [62] or similar method to create a coherent set of findings.
- Determine where to focus resources for subsequent design efforts: Prioritize the list of issues from the prior step based on those that are most disruptive to cognitive work.

Tips:

- If you have 2 – 3 hours to conduct the evaluation, use the full set of CPIs. If you have less than 2 hours to conduct the evaluation, consider using a subset of the CPIs that are most relevant.

Companion Methods: Gather methods such as the Task Diagram Interview can be used to quickly collect information on the cognitive work the users of the system will engage in. The Critical Decision Method can be used to identify challenging scenarios that the system must support. Consider also conducting a Heuristic Evaluation to identify usability issues.

Resources and References:

- [62] H. Beyer and K. Holtzblatt, Contextual Design: Defining Customer-Centered Systems, San Francisco, CA: Morgan Kaufmann Publishers, 1998.
- [63] S. L. Wiggins and D. A. Cox, "System evaluation using the cognitive performance indicators," in Macrocognition Metrics and Scenarios: Design and Evaluation for Real-World Teams, J. E. Miller and E. S. Patterson, Eds., Boca Raton, FL, CRC Press, pp. 285 - 302.

B.2 Formal Usability Testing

In a formal usability test, operational users guide the researcher through a design, functional prototype, or working system and attempt to complete a set of realistic test scenarios, typically with a minimal amount of assistance. A series of formative usability tests may be conducted at different stages of development to identify and fix usability problems, obtain a better understanding of user needs, refine requirements, and ensure that the final system will meet criteria for efficiency, effectiveness, and satisfaction. For example, an iterative process of repeated formative evaluations of prototypes can be used to improve the design to better match user needs. Summative usability tests are conducted to measure the degree to which a system meets measurable requirements for effectiveness, efficiency, and satisfaction. Summative evaluations are more comprehensive and quantitative in nature than formative evaluations, and they are typically conducted before a software release. From a cognitive engineering standpoint, usability testing must also assess the effectiveness of the system supporting cognitive work and decision making.

Timeframe: Conduct one or more formative usability tests throughout development conduct a summative test at the end of development or at the end of each product increment. It generally takes 2

– 4 weeks to prepare materials, conduct testing, analyze results, and summarize and prioritize improvements.

Use When: Assessing the degree to which a system meets measurable performance criteria for the intended user population.

Steps:

- Create test scenarios: Develop a set of one or more test scenarios with identifiable outcomes that exercise the goals and decisions the system is intended to support.
- Determine data to collect and collection methods: Determine the evaluation measures that will be collected and how they will be collected. Evaluation measures are used to assess:
    - Effectiveness: Whether users can use the system to achieve the goals it is intended to support. Example effectiveness measures include the unassisted scenario completion rate (i.e., user ability to accomplish the scenario without assistance), the assisted scenario completion rate, and the number of assists. Incorporate measures that address the ability of the system to support cognitive work and decision making.
    - Efficiency: Once the user has learned the system, how quickly they can accomplish their tasks using the system. Example metrics include the time to complete each scenario, or time to complete tasks correctly (i.e. build a course of action or a briefing using given capabilities).
    - Satisfaction: The degree to which users like using the system, typically measured using the System Usability Scale (SUS) [48].
- Create training materials: Determine and create the training that participants will need, either ahead of time or during the evaluation.
- Create other test materials as required, including:
    - Develop a pre-test questionnaire to elicit demographic information and request users' preferences regarding potential contact after the test (e.g., are they willing to be contacted at a later date for follow-up questions?).
    - Author a script so that each test participant receives the same information prior to and during the testing sessions.
    - Prepare informed consent forms if required by the Institutional Review Board (IRB).
    - Create an exercise for users to practice "thinking aloud" while working, such as finding a document on their computer.
    - Develop a post-evaluation questionnaire, for example, with free text questions and the SUS.
    - Develop a guide for observers, so that they know what to look for and capture; including a template for them to fill out while observing.

- Recruit participants: Try to obtain a representative sample of users with various backgrounds and experience levels (e.g., novice, intermediate, expert).
- Conduct the evaluation: Guide participants through each scenario in one-on-one sessions.
- Report findings: Consolidate findings across participants. Identify positive elements of the design, evaluation measure outcomes, design issues, key user comments and quotes, usability issues, and prioritized recommendations for improvements.

Tips:

- Conduct an expert review of the software (e.g., a heuristic evaluation) to resolve issues before testing with users.
- Obtain IRB approval and informed consent from participants if required (see Section 3.3.5).
- Dry-run the test with SMEs or other proxy users to ensure that the tasks are reasonable and the supporting materials are clear.
- Inform participants that the software is being tested, not their performance.
- Ensure that any stakeholders who are not conducting the test do not interrupt the participants or offer to help them.
- Combine observer notes immediately after the fact, and code them to identify users' requests for feature enhancement, negative and positive comments about the interface, etc.
- Analyze the combined notes to determine urgent needs for design changes, including consulting with software developers to determine the cost-benefit ratio for making changes.

Companion Methods: See sections on Developing Decision-Centered Test Scenarios and Defining Evaluation Measures for guidance on developing scenarios and measures to support the usability testing.

Resources and References:

- [64] D. Mayhew, The Usability Engineering Lifecycle, Burlington, MA: Morgan-Kaufmann, 1999.
- [65] R. W. Pew and A. S. Mavor, Human-System Integration in the System Development Process, Washington, D. C.: National Academies Press, 2007.
- [66] T. Tullis and W. Albert, Measuring the User Experience, 2nd Ed., Burlington, MA: Morgan-Kaufmann, 2013.

B.3 Defining Cognitive Evaluation Measures

The purpose of cognitive evaluation measures is to assess the degree to which a system supports cognitive work and enables proficient performance. Evaluation measures can be used to compare systems or candidate design concepts, ensure that a system meets certain objective performance criteria, or assess the degree to which changes made to a system improve (or degrade) performance. They may be assessed in Human-In-The Loop Simulation Events (Evaluation Events) and in Formal

Usability Tests. In this section, we describe the creation of measures to assess cognitive work, human-machine teaming, workload, situation awareness, and usability.

Steps:

- Identify measures based on cognitive work requirements: Cognitive work requirements are the cognitively challenging aspects of a task (e.g., that involve planning, re-planning, decision making sensemaking, etc.), along with the reasons for the difficulty. They are measures specific to the nature of the cognitive work and decision making that the system is intended to support. Such measures stem from research conducted in the gather and analyze phases to understand the cognitive requirements of the work, and they also stem from knowledge about the macrocognitive functions and processes the system is intended to support [4]. You might leverage existing cognitive work requirements or user stories developed in the analyze phase as the basis for such metrics. For example, such metrics might address the ability of the system to support a set of decisions, make sense of ambiguous situations, re-plan after conditions change, coordinate with team members to maintain common ground, or address other macrocognitive challenges. Measures should describe aspects of cognitive performance that need to be supported and how (e.g., how much, what kinds, when, and for how long).
- Identify measures based on human-machine teaming support requirements: These are measures that address the degree to which the system exhibits attributes that any system must have in order to effectively support cognitive work and facilitate human-machine teaming. They are rooted in evidence-based human-machine teaming principles, including Observability, Predictability, Directing Attention, Exploring the Solution Space, Adaptability, Calibrated Trust, and Common Ground [17]. Some principles, such as Calibrated Trust, have established evaluation techniques [67]. Others, such as Observability, will require the creation of tailored measures that describe how the system being evaluated may exhibit the principle given the nature of the cognitive work requirements. For example, measuring Observability may require the creation of metrics to assess the degree to which the operator understood the basis behind a system decision recommendation. Create measures for those principles that are most important for the system.
- Identify mental workload measures: Mental workload is the effort expended to achieve a certain level of performance (e.g., the effort expended to make a decision or perform a task using the system). There are many established techniques to subjectively assess mental workload; widely used measures including the NASA Task Load Index (NASA-TLX) [57], the Subjective Workload Assessment Technique (SWAT), and the Overall Workload Scale [50].
- Identify situation awareness measures: Situation awareness describes the possession of knowledge relevant to the task being performed. Established techniques to measure situation awareness include the Situation Awareness Global Assessment Technique (SAGAT) [56] and the Situation Awareness Rating Technique (SART) [68].
- Identify usability measures: Usability is the "extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" [69]. Usability measures include [66]:

- Effectiveness: Whether users can use the system to achieve the goals it is intended to support. Effectiveness measures should largely be covered by measures to assess cognitive work requirements described above.
- Efficiency: Once learned, efficiency is how quickly users can accomplish their tasks using the system. For example, it may be measured by the time to complete a task.
- Satisfaction: The degree to which users like using the system, often measured using the System Usability Scale (SUS) [48].

- Identify the data collection strategy for each measure: Identify the specific data that will be collected for each measure and how the data will be collected.

Tips:

- Defined measures are not necessary to address all of the items described above. Focus on developing measures based on cognitive work requirements – these are the most important to assess.

Companion Methods: Cognitive Evaluation Measures may be assessed in Human-In-The Loop Simulation Events (Evaluation Events) and in Formal Usability Tests.

Resources and References:

- [48] J. Brook, "SUS – A quick and dirty usability scale," in Usability Evaluation in Industry, P. W. Jordan, B. Thomas, B. A. Weerdmeester and A. L. McClelland, Eds., New York, Taylor & Francis, 1993.
- [49] B. Crandall, G. Klein and R. R. Hoffman, "Cognitive task analysis for measurement and evaluation," in Working Minds: A Practitioner's Guide to Cognitive Task Analysis, B. Crandall, G. Klein and R. R. Hoffman, Eds., Cambridge, MA, MIT Press, 2006, pp. 229-244.
- [56] M. Endsley, "Situation awareness global assessment technique (SAGAT)," in Proceedings of the IEEE 1988 National Aerospace and Electronics Conference, Dayton, OH, 1988.
- [68] S. J. Selcon and R. M. Taylor, "Evaluation of the situational awareness rating technique (SART) as a tool for aircrew system design," in AGARD Conference Proceedings No. 478, Neuilly-sur-Seine, France, 1989.
- [57] S. Hart, "NASA-Task Load Index (NASA-TLX): 20 years later," in Proceedings of the Human Factors and Ergonomics Society Annual Meeting, San Francisco, CA, 2006.
- [4] G. Klein, K. Ross, B. Moon, D. Klein, R. Hoffman and E. Hollnagel, "Macrocognition," IEEE Intelligent Systems, vol. 18, no. 3, pp. 81-85, 2003.
- [67] P. McDermott, "Calibrated Trust Evaluation Toolkit," The MITRE Corporation, 20 August 2019. [Online]. Available: https://comm.mitre.org/calibrated-trust-toolkit/. [Accessed 9 January 2020].
- [17] P. McDermott, C. Dominguez, N. Kasdaglis, M. Ryan, I. Trahana and A. Nelson, "Human-Machine Teaming Systems Engineering Guide," The MITRE Corporation, Bedford, MA, 2018.

- [50] V. J. Gawron, Human Performance, Workload, and Situational Awareness Measures Handbook, Boca Raton, FL: CRC Press, 2008.
- [66] T. Tullis and W. Albert, Measuring the User Experience, 2nd Ed., Burlington, MA: Morgan-Kaufmann, 2013.
- [69] ISO, "Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability," ISO 9241-11, 1998. [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=16883. [Accessed 18 December 2019].
- [70] J. Miller and E. Patterson, Eds., Macrocognition Metrics and Scenarios: Design and evaluation for real-world teams., London and New York: Routledge, Taylor and Francis Group, 2016.

B.4 Developing Decision-Centered Test Scenarios

Operationally realistic and relevant test scenarios underpin a variety of evaluate methods. Without the proper scenarios, the cognitive evaluation measures described above are useless. From a cognitive engineering standpoint, it is essential that test scenarios illuminate challenging aspects of cognitive work and decision-making that a system is designed to support. This section outlines a method to construct test scenarios that are based on principles of effective cognitive support; it also outlines the steps to use such scenarios in a Decision-Centered Testing (DCT) methodology.

Timeframe: Most time will be spent in the gather and analyze phases, determining the nature of cognitive work and decision making. Constructing the actual scenarios may take on the order of a few days to several weeks.

Use When: Developing test scenarios that will be used to evaluate systems that support cognitive work. Employ DCT in an iterative fashion to identify design strengths, weakness, and improvements.

Steps:

- **Define cognitive support requirement focus:** Determine which of the cognitive support requirements will be the focus of the test scenario. Cognitive support requirements describe the attributes that any system must exhibit in order to effectively support human-machine teaming, such as Observability and Directing Attention [71]. For example, testing for Directing Attention might involve assessing the ability of the system to help shift focus from one problem to another.
- **Define the cognitive work and decision making activities:** Define the specific cognitive work or decision making activities to be assessed (e.g., “determine an engagement sequence against a set of incoming missiles”). Also define the variables that make each cognitive work activity or decision making event difficult.
- **Define test conditions:** For each cognitive work and decision making activity, determine how test conditions will be varied with respect to the difficulty of the task.

- **Define scenarios:** Finally, define a set of test scenarios that describe the specific series of events that will bring about the desired level of difficulty for each cognitive work and decision making activity while also illuminating the cognitive support requirements that are the focus of the test.

Companion Methods: Use gather methods such as the Critical Decision Method, Task Diagram, and/or Knowledge Audit, and/or analyze methods such as Goal Directed Task Analysis or Cognitive Work Analysis to identify key goals and decisions. DCT scenarios are used in a variety of evaluate methods, including the Cognitive Wall Walk, Cognitive Performance Indicators Evaluation, and Formal Usability Testing.

Resources and References:

- [72] S. S. Potter, W. C. Elm and J. S. Tittle, "Evaluating the resilience of a human-computer decision-making team: a methodology for decision-centered testing," in Proceedings of the 8th International Conference on Naturalistic Decision Making, Pacific Grove, CA, 2006.
- [73] R. Rousseau, J. Easter, W. Elm and S. S. Potter, "Decision-Centered Testing (DCT): Evaluating Joint Computer Cognitive Work," in Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting, Orlando, FL, 2005.
- [74] M. E. Jobidon and R. Breton, "Initial validation of the decision centered testing methodology," Defence R&D Canada -Valcartier Technical Report No. TR 2008-302, Valcartier, Quebec, 2009.