Unlocking User-Centered Design Methods for Building Cyber Security Visualizations

Sean McKenna*  Diane Staheli  Miriah Meyer
University of Utah  MIT Lincoln Laboratory  University of Utah

Abstract—User-centered design can aid visualization designers to build better, more practical tools that meet the needs of cyber security users. The cyber security visualization research community can adopt a variety of design methods to more efficiently and effectively build tools. We demonstrate how previous cyber visualization research has omitted a discussion of effectiveness and process in the explanation of design methods. In this paper, we discuss three design methods and illustrate how each method informed two real-world cyber security visualization projects which resulted in successful deployments to users.

1 Introduction

The practice of user-centered design incorporates careful consideration of users’ needs, wants, and limitations throughout the design process [6], which helps in evaluating both the effectiveness and appropriateness of tools [27]. In a survey of the Visualization for Cyber Security (VizSec) proceedings from the past 5 years, roughly 40% of the 51 papers included evaluation with users, mirroring the findings of a recent survey looking back a full 10 years [33]. Only 7 of these 51 papers discuss iterative evaluation with users to improve the design of a tool, with the more common case being evaluation with users only after the design of a tool is complete. Thus, there is an opportunity within the VizSec community to improve the efficacy of visualization tools by using evaluation and user-centered design methods throughout the entire design process, which includes gathering user needs, design opportunities, and ideas before even building a tool; we found only 1 instance of a VizSec paper which did so in the past 5 years [38].

Introducing users into the design process often requires significant time commitments on their part. Cyber security analysts have very limited time due to the fast-paced nature of their jobs, leaving visualization designers with limited access to these domain experts. Simultaneously, analysts and visualization designers are challenged by the variations of cyber security data, the uniqueness of different computer networks, and the complexities of the threat analysis process [1]. Adopting user-centered approaches within the design of cyber security visualization tools thus requires methods that are not only effective, but also efficient.

In this paper we discuss three design methods — qualitative coding, personas, and data sketches — and frame their use specifically for designing visualizations of cyber security data from a user-centered perspective. We ground the discussions of these methods in two different cyber security visualization design projects, and we illustrate how each design method was both efficient and effective for this design space. For each method, we present outcomes from the design projects, as well as practical usage recommendations, which we believe will be useful for future cyber security projects.

We begin the paper by discussing related work in Section 2, followed by a brief introduction to the design process model used by both example design projects in Section 3. Then, we discuss three efficient and effective design methods for cyber security visualization design in Section 4 and wrap-up in Section 5.

2 Related Work

By focusing on the needs, wants, and limitations of users, user-centered design enables users to achieve their goals more effectively, efficiently, and with increased satisfaction, thus providing benefits such as increased productivity, better accessibility, reduced stress and risk of harm, and an improved user well-being [6]. Within the cyber security visualization literature, a number of user-centered design methods have been utilized. Komlodi et al. performed iterative usability studies on visualization prototypes to improve upon their glyph design [19], while Hao et al. focus their discussion on justifying a web visualization framework [15]. Furthermore, Paul et al. present a design-first approach for finding innovative visualization solutions that emphasizes visual concepts before user requirements [29]. The limitation of these works is that they do not address the usefulness of design methods early in the design process to obtain requirements from users.

Several cyber security papers have discussed user-centered design methods during the early phases of the visualization design process, but these papers have rarely linked these methods to a final, deployed tool. Goodall et al. interviewed analysts to derive requirements for a network security tool [14], while Stoll et al. explain the personas design method [34]; however, neither of these methods were validated for their efficacy or efficiency. The cyber command gauge cluster by Erbacher utilized a human-in-the-loop process with real users [7], but the project only described a prototype state. Wagner et al. conducted several different design methods to uncover user needs [38], while Best et al. identified user needs and domain challenges by building a prototype visualization system with users [1]. A co-creation approach is described by Landstorfer for building pixel carpets [21], but as with previous work they only describe a prototype, not a system deployed to users.

One common methodology in cyber security to understand user needs is cognitive task analysis (CTA) [2, 4, 5, 9, 11, 13, 23, 42], which focuses on the unobservable, cognitive activities of users [40]. While CTA can produce a rich analysis of users’ cognitive processes, CTA methods require significant time from users for study. In addition to analysts’ time constraints, researchers may have difficulty gaining access to organizations and confidential data [41, 42]. Since a CTA methodology contains many possible methods, the expertise required, training time, and performance time can vary considerably [40]. However, the wealth of information available from published CTAs for cyber security is valuable information for visualization designers, and we advocate utilizing design methods that build off of this knowledge whenever possible.

3 Design Activity Framework

The work presented in this paper utilizes the design activity framework, a design process model that focuses on the steps a designer takes during the course of building a visualization [26]. Many design models exist within visualization literature for the purposes of both structuring and explaining design projects [17, 32, 37, 39]. We advocate for the use of a design process model that incorporates users throughout the development life cycle of a visualization tool to increase the chances
for a successful project that is both effective and efficient.

The design activity framework consists of four different design activities: **understand**, ideate, make, and deploy. Each of these activities consists of a **motivation**, design **outcomes**, and **methods**. As shown in Figure 1, the motivation places the designer within a specific activity in the framework, with the goal of establishing a specific set of outcomes for that activity. Outcomes are achieved using one or more design methods, both for generative or evaluative purposes. The design activity framework supports iterative, user-centered visualization design, and provides guidance about effective methods for reaching a range of design goals. While the focus of this paper is on three specific design methods, we ground these methods within the design activity framework to provide guidance about how and when these methods can be used most effectively.

We can use the framing of a design activity to find effective methods for cyber security visualization design. We define effectiveness here as a reflection in two parts: short-term and long-term. In the short-term, an effective design method must successfully achieve the desired outcome for the design activity — we argue that this completed outcome is one way to validate a design method. The long-term effectiveness of a method can be established when the method is used within the development of a deployed visualization tool: one that is evaluated with, and given to, real end users. Thus, we can determine if a design method was effective within a project by reflecting on these two questions:

1. Did you achieve your desired outcomes?
2. Did you deploy a tool to users as a result of this method?

We will return to these questions in Section 4 to discuss the effectiveness of the three methods presented in this paper for enabling successful designs of cyber security visualizations.

### 4 Design Methods for Cybersecurity Visualization

As discussed in Section 2, a number of user-centered design methods have been discussed in the cyber security visualization literature, such as interviews, observations, usability testing, focus groups, and workshops. A few methods were discussed in the context of a larger design process, but none of these methods were validated in the context of contribution to a completed, deployed visualization tool. Furthermore, there are many other user-centered design methods that have yet to be demonstrated for cyber security visualization design. For example, an extensive list of 100 different methods was discussed in the context of the design activity framework [26]. Thus, there is an opportunity to introduce and validate these methods in real-world, cyber security visualization projects.

Here, we discuss three design methods that were validated in the context of two cyber security visualization projects. The first project was the redesign of a cyber security firm’s large software tool [26], and the second project was the design of a web dashboard for a network operations center. By situating these methods within our design process of these projects, we are able to reflect on their efficacy and provide guidance for their use. The three methods we discuss are qualitatived coding, personas, and data sketches. The qualitative coding method played a key role in the understand activity of the software redesign project, while the personas and data sketches methods both played instrumental roles in the understand and ideate activities of our web dashboard design project.

For each method, we first discuss our motivation to place that method in the context of the larger design process. Then, we highlight the outcomes achieved, followed by results and implications of what we learned and a discussion of the method’s efficiency, effectiveness, and limitations. Lastly, we present recommendations for using each method for cyber security visualization design.

#### 4.1 Qualitative Coding

When tasked with redesigning a large cyber security tool, our design team had limited access to end users. Despite the fact that a fully deployed tool already existed, we were taking a step back to find users’ needs in the first design activity: **understand**. Our motivation in this activity was to better understand the needs and design opportunities for network security analyst to redesign the firm’s tool. But how do we identify user needs without direct access to end users? Many researchers have studied users in this domain from a variety of perspectives, particularly with cognitive task analyses. For this project, we built off of this rich existing body of knowledge through qualitative coding of three cognitive task analyses.

We took inspiration from the social sciences [35] to help structure our analysis by performing an open coding on several key CTA papers from the field. Qualitative researchers often use coding as a method to organize, structure, and consolidate information into a structured framework. Open coding is a subset of qualitative coding, which focuses on the original content to form the codes you make, as opposed to axial coding, which incorporates existing categories to tag onto the source material [35]. This method has been utilized by visualization researchers to perform various post-hoc analyses [16, 20, 31, 33], but we had not seen this method used in the understand activity to pinpoint user needs for cyber security.

After half a month of extensive literature review, four members of our design team identified and performed a deep reading on three cognitive task analysis papers [4, 8, 11], pulling out key quotes, paraphrases, and models. Each piece of data corresponds to rows of our coding table, and we met several times over a month to better organize, iterate on, and consistently tag this information across all three papers. These meetings and iterative coding process were crucial to allow the design team to come to an agreement on our final codes. After a month of open coding the three papers, we consolidated all of our data together in a final meeting.

#### Outcomes

We present a sample outcome of our coding method in Figure 2; a more complete table of all the data is included in Supplemental Materials. Each piece of information is organized across one or more papers and into a hierarchy of categories. At the top-most level, we identified categories such as data, design guidelines, phases, roles, responsibilities, tasks, terminology, tools, and workflows. Additionally, we tagged information with sub-categories on a finer scale.

Focusing the data from these three CTAs enabled us to identify user needs without the user, as we had limited access to cyber security analysts. Over the course of a few weeks, our design team synthesized the codes into a set of distinguishable design opportunities, such as provenance, scalability, usability, desirability, data type handling, and data hierarchy continuity. We used our knowledge from the qualitative coding method to prioritize this list and distinguish opportunities with the most potential to impact cyber security analysts. This produced our final thematic design opportunities for improvements to the
Fig. 2. A sample of qualitative codes pulled from three cognitive task analyses papers. For more details, please see Supplemental Materials.1

existing tool: usability, workflow improvements, desirability, and temporal data representation.

Results and Implications

After identifying key design opportunities, our design team iterated on a series of ideas for the company to improve their tool. We sketched out and detailed a more usable welcome screen, added a widget for sharing messages among analysts, highlighted recent user activity to promote sharing, visually clarified distinctions between vulnerabilities and alerts, and created a new overview timeline visualization to coordinate all views. A software developer incorporated these changes, and the updated tool was tested with Department of Defense analysts using an A/B evaluation method. The result of this evaluation was that the redesigned tool was more usable and effective than the previous design.

Lastly, the qualitative coding method enabled us to identify extensions to a well-known data hierarchy model for cyber security situational awareness [4] — we present this extension in Figure 3. The original data model describes how analysts process, filter, sort, and select data, as it transfers from raw data into situational awareness. Our extensions highlights the data feedback loop, clearly shows the outputs from this feedback loop, and provides identification of tasks for filtering the data across levels.

Discussion

The qualitative coding method was efficient as compared to more complex methods, such as a multiple-analyst cognitive task analysis; we conducted the qualitative coding in under two months. As for the effectiveness of this method, we were able to focus our user needs into a set of concrete design opportunities to produce the desired outcome: understanding of user needs without direct access to users. These design opportunities led to the final redesign of a deployed tool that analysts found more usable and effective than before. The complete table of our coding results1 can be utilized by others to identify, categorize, and prioritize different user needs in future cyber security design projects. A limitation to this work is that it focuses on the details for only three papers; these results may be extended by coding additional papers from this field. Furthermore, one caveat to this approach is that published research may not reflect all the nuances of an operational environment. Thus, this method should not simply be used to replace access to real users.

---

1. For more details, please see Supplemental Materials.
Recommendations

- Start your coding method on a few papers to develop an initial set of codes; select papers from appropriate venues:
  - e.g. VizSec, VIS, CHI, HFES, Behavior & Information Technology, Computers & Security, FIRST, HST, AMCIS, SAM, CyCon, FloCon, CogSIMA, DHS CATCH, HCI HAS, CTS SECTORS.
- On the first pass, highlight and tag key pieces of information; we suggest starting with the categories we identified.1
- Limit the time and scope on your first pass of coding; spend more time to meet as a team and agree on codes.
- Once you reach a consensus on codes, expand to more papers and divide up the work, allowing some overlap in coverage for consistency.

4.2 Persons

The next design method we present was utilized during our second project: designing a cyber security dashboard for communication of cyber information. We have included sample images of the dashboard in Figure 4 to show how a design method can iteratively improve upon the design of a final deployed tool. We began this project with a broad, and fuzzy, goal, requiring us to take a step back and identify the needs of the users; again, we started in the understand/design activity. But who were the real users for a dashboard? With the task of communication, we surmised that more than one type of user was meant to utilize the dashboard. We could not find much research discussing users beyond network analysts, so our motivation was to uncover information on a range of users for cyber security to help form the design opportunities for this project. This motivation is an ideal fit for the personas design method.

The personas method is often utilized within the user-experience, design, and HCI communities [3, 10, 24, 25, 30]. Personas are documents meant to foster communication within a design team as archetypes of users, their behaviors, and their knowledge [24]. Within the cyber security domain, Stoll et al. describe a specific methodology for using personas, highlighting their benefits for cyber security visualization design [34]. Here, we further this work in three ways. First, we describe how personas benefit the communication within a design team. Second, we add visual elements to our personas to promote fast visual comparison of multiple user profiles and highlight interactions between personas. Third, we tailor our personas to the field of cyber security by incorporating key aspects of cyber situational awareness.

We developed the personas based on a dozen semi-structured interviews conducted over six weeks with various stakeholders: network analysts, managers, researchers embedded in cyber operations, and various other cyber security and business-focused users. Reflecting on the data gathered during these interviews and existing literature, we produced personas for four different kinds of users: an analyst, manager, director, and CEO. Once we identified four different kinds of users for our project, we narrowed the project’s focus to specifically design our dashboard for only two of the personas: analysts and managers. By isolating these two types of users, we were able to keep our focus consistent throughout the rest of the design process; from development to evaluation, these two user archetypes became the key motivation to justify and balance all our decisions as a design team.

Outcomes

We present the resulting personas from our project in Figure 5 and provide them in Supplemental Materials.3 The four personas are: a cyber analyst, a network operations center (NOC) manager, a director of information technology (IT), and a chief executive officer (CEO). For each persona, we pinpointed the goal or domain-specific task for each archetypal user and visually illustrated the user’s cyber knowledge and situational awareness (SA) focus. We also considered the range or window of temporal data that each user requested, illustrating how to represent visualization-specific needs within a persona. Next, we highlighted each user’s key cyber SA questions, pulling from an existing question taxonomy as a basis [28]. Lastly, we identified the general flow of both decisions (downward) and information (upwards) between these personas to characterize interactions taking place between them.

Results and Implications

Personas played a critical role in helping us to decide which users and needs to target in our design process. Narrowing the focus of our dashboard project early was crucial due to the time limitations of our project. We decided that the dashboard should not be too high level for just CEO’s nor just another tool for analysts. We targeted our dashboard to both cyber analysts and managers by combining features for analysts to quickly explore the data with standard visualizations for managers to quickly comprehend the details of the data; see Figure 4(a) for the first prototype of our design using these two personas. Furthermore, the narrowed design focus uncovered several key user needs for our project. By brainstorming off these needs, we were able to ideate upon various dashboard designs and compare how they worked for different users based on the personas we created. Some examples of these user needs include: intuitive and easy-to-use, communication and presentation, ability to provide details-on-demand, simplification and aggregation of data, adaptability, and promotion of collaboration between users. The personas continued to aid our design team in both communicating and evaluating the dashboard, up until its deployment.

Discussion

The personas presented in Figure 5 can be used as a starting point or tailored by others in future visualization design projects for cyber security. Furthermore, these personas can be modified for different project motivations and user needs; it is common for personas to alter and become more refined over time [3]. The personas design method took less than three months’ time including the interview process, and this method resulted in the design of a deployed dashboard. Thus, the personas method can be both efficient and effective for cyber security visualization design. Additionally, the personas method can be data-driven, where personas are built and evaluated against data directly captured from users [25].

Recommendations

- Use personas to target the right users for a design or to evaluate a design with your users in mind.
- Talk with real users to build personas; if you cannot, use existing research or qualitative coding of the literature.
- Pinpoint user goals, knowledge, behaviors, and activities, focusing on both similarities and differences across users.
- Incorporate visual encodings when appropriate to enable easier and faster comparison across personas.
- Use and adapt personas over time; keep them as a living document to fuel multiple design projects.

4.3 Data Sketches

As originally pioneered, data sketches allow a designer to “quickly and flexibly produce transient and uncertain visual representations of domain data by scavenging existing applications for functionality that allow data, interactions, and functionality to be combined” [22]. In other words, a data sketch is a visualization developed using available software tools. We incorporated data sketches into our design of the cyber security dashboard during our understand/ideate design activities in order to establish a more complete data and task abstraction for the communication of cyber information. Our motivation was to better understand an analyst’s needs, and to ideate further on the potential design options; we also sought recommendations for cyber security dashboard design. We reached out to a network security analyst at the University of Utah to obtain real-world data for the data sketches, and followed-up with this analyst to get feedback on the sketches.

We obtained a network flow dataset from our collaborator containing over 2.3 million network flows, which captured over 0.4 TB throughput on the university’s network. This dataset captured a five-minute snapshot of the network traffic. In developing data sketches of this flow dataset our focus was not on the scale or optimization of
the data, but how to best represent the data. The question we wished to answer was this: if this is the raw data we have and given our technical network security analyst user, what views are appropriate, or inappropriate, to use in a dashboard?

We spent a month sketching with this data. We utilized Python to simplify, aggregate, and parse the data in various ways, and used Tableau, Gephi, and D3.js to produce a variety of visualizations. Even with these powerful visualization tools, it was still challenging to explore this relatively small cyber security dataset. To supplement our own sketches, we also included images from existing literature of less common and more complex visual representations that made use of real-world cyber security data [12, 18, 36].

Outcomes

We present an overview of the twenty data sketches we produced in Figure 6; please see Supplemental Materials1 for a full-page version of each data sketch. We categorized each of the data sketches into four high-level groupings — network graphs, maps, aggregated charts, and time — which helped guide our discussion with our network analyst. We performed a free-form, informal evaluation session with our analyst for three hours to see which visual representations were easily understood and potentially most useful. These data sketches can be repurposed in future projects for further brainstorming.

Results and Implications

We showed each data sketch to our analyst; here we summarize the analyst’s feedback for each kind of data sketch.

- **Network Graphs**: The analyst was un convinced that the graphs could show meaningful insights at scale with each node representing a single IP address. Furthermore, the layout algorithm confused the analyst since it positioned each IP address at a location that was not meaningful to the analyst.

- **Maps**: In contrast to the network graph sketches, the map representations garnered positive feedback from the analyst, in particular the cartograms due to their novelty.

- **Aggregated Charts**: These charts concerned the analyst because the finest level of detail was not available. We also included one data sketch to show a 3D data chart, which seemed to entice the analyst despite our continued warnings about the usability challenges of 3D for cyber security visualization [19]. More unique kinds of visualization, such as parallel coordinates and treemaps, confused the analyst on first glance and required further explanation. After explanation, the analyst commented that parallel coordinates seemed promising for exploring multidimensional data, while the treemaps, which showed the IP address hierarchy, seemed less useful.

- **Time**: These sketches were discussed in less detail; however, the analyst stated that the timestamp was one of the least important data fields to him.
Fig. 6. An overview of the twenty data sketches we evaluated with a cyber security analyst; this feedback was critical to our redesign of a cyber security dashboard in Figure 4(b). We categorized each sketch into four groups: network graphs, maps, aggregated charts, and time. Several data sketches we pulled from existing literature [12, 18, 36]. We provide a full-page version of each data sketch in Supplemental Materials.
• Avoid visual representations that require significant explanation, such as parallel coordinates or treemaps.
• Precise details on the time scale may not be immediately vital.
• Summary views for communication can use aggregation.
• Aggregation of data should be immediately obvious.
• A map-based view could aid the discovery of patterns.

With these considerations in mind, we revisited our initial dashboard design and performed another iteration on the ideate and make design activities to produce the final dashboard design shown in Figure 4(b).

The major change made in the final design is the type of encoding, using a map view with aggregation over time. This change was, in part, driven by the results of the data sketches method, which showed the potential of aggregation and map-based views for discovering and communicating cyber data.

Discussion
We found that data sketches were very time efficient; the entire process took about two months to set up, perform, evaluate, and analyze. Furthermore, these data sketches were effective in our design process for producing a set of recommendations for dashboard design, and for pinpointing certain representations of the data as promising. Furthermore, this method provided some key insights for our redesign of the dashboard, which is currently deployed to users. These data sketches and the feedback we received can be used by others to inspire and evaluate their own visualization design projects for cyber security.

There were several limitations to our approach. First, several of the sketches presented were taken from images in the literature, and thus were not based on our collaborator’s data. Unfortunately, many of the tools in visualization papers, particularly for cyber security, tend not to be publicly available or provide a consistent data format for others to easily and readily use the tools for such an exercise. This meant we either had to not include these more unique and interesting visualizations in our set, or compromise by showing alternative data; we opted for the latter and included a brief description of the data being used for each encoding. The second limitation was that we only received feedback on the data sketches from one analyst. While additional analyst feedback would be preferable, the feedback we did receive was helpful for allowing us to pull out potential design ideas and focus on a smaller subset of ideas quickly.

Recommendations
• Incorporate real data whenever possible; if you cannot, use realistic datasets like the VAST challenge datasets.
• Repurpose the tools you know, and experiment with new ones (e.g. Python, Tableau, Gephi, D3.js, Processing, Excel, Spotfire, Arcsight, Splunk).
• Utilize real-data examples of visualization tools if a tool is unavailable or requires excessive time to input your data.
• Explore both interaction and animation in your data sketches.
• During evaluations, provide users with tasks or prompts if your goals require focusing the user feedback.
• Users may provide initial positive feedback on sketches because they are novel; consider re-evaluating at a later time.
• Introducing many data sketches at once can overload users; consider introducing sketches in multiple sessions.

5 Conclusion
In this paper we demonstrate that user-centered design methods are both efficient and effective for cyber security visualization design. We utilize the design activity framework to describe our design process and to validate the effectiveness of three design methods: qualitative coding of CTAs, personas, and data sketching. Through two real-world project examples, we highlight our motivations, outcomes, and results using these methods. Furthermore, we explain our insights and provide practical recommendations for using these methods in cyber security projects.

User-centered design methods can help a designer establish user needs, uncover design opportunities, and evaluate ideas. The three design methods discussed in this paper can incorporate any varied number of users. We encourage future cyber security visualization projects to broaden the methodologies, methods, and techniques at their disposal in order to more completely explore this design space. Ultimately, embracing user-centered design methods and the importance of the design process will help us as a community to be more efficient at building effective visualization tools for the cyber security community.

Acknowledgments
The authors wish to thank Jonzy, Dan Bowden, and Tamara Denning for the data sketches method, staff members at MIT Lincoln Laboratory for the personas method, Dominika Mazur, Matthew Parkin, and James Agutter for the qualitative coding method, and the Visualization Design Lab at the University of Utah for their feedback on this work. This work is sponsored in part by the Air Force Research Laboratory, the DARPA XDATA program, and by the U.S. Army Research Office under a prime contract issued to Intelligent Automation, Inc. The Lincoln Laboratory portion of this work was sponsored by the Assistant Secretary of Defense for Research & Engineering under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government or Intelligent Automation, Inc.

References


