Exploring design and product development data in high-tech companies using data visualization

Master's Thesis in Human-Computer Interaction
Uppsala Universitet: Department of Informatics and Media

Valjean Clark
Supervisor: Else Nygren
29.05.2013
Abstract

In high-tech companies, user experience has become an important part of data-driven product development, but unfortunately the artifacts created by user experience work are often stored in disparate documents and databases, and it is difficult to get a big picture of all the artifacts that have been created and how they relate to each other and to other product development data. Data visualization is one way to approach solving these issues, but how can this data be presented to users in a meaningful way such that both the artifacts and the relationships between them are understood? Three hierarchical data visualizations - partition, sunburst, and zoomable treemap - were built around some early product development data. These visualizations were tested in a comparative exploratory usability test with six users to find how well users understood the data and the relationships between the data. The most significant results were that most users did not know how to interact with the partition and sunburst visualizations until prompted to do so, users had a difference in understanding the data between the sunburst and partition visualization, and some techniques worked very well to keep users oriented while others did not. Future work should first focus on adding a creative element to the tool, where data can be added and relationships can be built in the visualization itself, and then focus on testing the tool again with a more specific audience of strategic planners, UX designers, and requirements engineers.
# Table of Contents

1 Introduction  
  1.1 Problem Statement  
  1.2 Initial Data Gathering  
  1.3 Relevant users  
  1.4 Context of use  
  1.5 Vocabulary and definitions  

2 Theory and related work  

3 Development of prototype  
  3.1 Choosing visualization software  
  3.2 Data model  
  3.3 Choosing visualizations to create  
  3.4 Building additional features onto visualizations  
  3.5 Technology requirements  
  3.6 Limitations of prototype  

4 User testing methodology  
  4.1 Research questions  
  4.2 Participant characteristics  
  4.3 Test design  
  4.4 Task list  
  4.5 Environment, equipment, and logistics  
  4.6 Data collected and evaluation measures  
  4.7 Limitations of user testing  

5 Results and discussion  
  5.1 Partition  
  5.2 Sunburst  
  5.3 Zoomable treemap  
  5.4 Results relevant to all visualizations  
  5.5 Summary of results  

6 Conclusion and recommendations  

Acknowledgements  

References  

Appendices  
  Appendix A: Introduction  
  Appendix B: UX Landing Zone task  
  Appendix C: Artifact definitions for test users
1 Introduction

In high tech companies, product development is structured in product development life cycles. Product development life cycles are characterized by a series of phases, typically following a definition - development - deployment structure. The structure itself is often shown as a funnel, working from more loosely defined concepts to concretely defined prototypes to final products. Deliverables and activities are specified within this structure.

User experience (UX) design has also gained traction in recent years in high-tech companies. Models have been developed for implementing and integrating UX practices (or a related body of practices, like user-centered design) with product development [24]. Explicitly or implicitly, changes to the activities and deliverables in product development are necessary to adapt to the activities and goals of UX design.

As companies become increasingly knowledge-driven, and as processes change and adapt, the amount of data needed to develop products and collaborate with multiple teams has grown in size and fragmentation, resulting in large amounts of unstructured data in myriad formats. Structured data has many benefits, and numerous tools exist in which to store and standardize structured data; however, the mere existence or even attempted enforcement of the use of such tools has not been sufficient to change the behavior of users to store data in a particular way.

This thesis is composed of three phases: gathering background data with stakeholders, building a prototype of the tool, and performing an evaluation of the tool. Background information was gathered using unstructured interviews, and the data gathered informed the design and development of the prototype. The process of prototyping the tool employed rapid web-based front end development to arrive at three different designs. These designs were then tested using a qualitative, exploratory comparison methodology to gather insights for future designs.

Note that all data shown in this paper is a substitute for the company data used in the actual visualizations. All properties and features of the data visualization itself are identical.

1.1 Problem Statement

Using the theory of design rationale, the problem can be framed as devising a course of actions to change the current situation into an improved preferred situation [8, 13]. Using this framing, the current situation can be stated as follows: People do not make use of the tools at hand to structure and store data in product development, particularly in the space of early design to the beginning of development. This has several negative implications: data from current and previous product development are not easily findable, data is not easily shown or understood in relation to other data, data cannot be easily shared. All of these things are hypothetically possible - people can search through scattered databases and documents to find and share data, then relate data by tracking relationships on their own. In practice, this kind of behavior will not just happen on its own.
To remedy the existing situation, either people must be coerced into existing new tools, or new tools must be created that help the users in their work. The preferred situation is for people to voluntarily store, structure, and relate their data, from high level design artifacts to lower level requirements. The question becomes how to achieve this preferred situation. Storing and relating structured data is easily solved by storing all data in some form of database. Product development data is often stored in databases or even disparate documents. However, getting users to input their data with a new tool is a more complex challenge than simply setting up a database and telling users to use it.

This research will attempt to address one piece of a larger research question: can a tool provide sufficient value to the user such that the user willingly adopts and uses the new tool, which in turn allows for the structured data needed at the early stages of product development? In order to achieve this, the new tool has some significant design problems to address. How would a large data set, currently consisting of disparate databases and documents, be combined into one meaningful data model? Can a tool be created to investigate this data, and how might this data be presented to the user in a meaningful way such that they understand the artifacts inside the data and the relationships between them? This research attempts to answer the latter question by surveying different types of information visualizations and comparing these visualizations' effectiveness in showing this kind of data.

1.2 Initial Data Gathering

Since the larger research question is to evaluate if a better tool can provide enough value for users to enable more structured data, this initial data gathering was performed to gather information to inform the design of such a tool. The questions that the initial data gathering sought to answer are as follows:

- What is your role and what parts of your work involve UX, design, and development?
- How is the data created and stored?
- How are links formed between related data?
- What tools are currently available and how are they being used?
- Discuss the positive and negative aspects of the tools in the context of product development and decision making.

Unstructured interviews have a set of questions and topics that guide the discussion between the interviewer and interviewee [10]. The interviewer asks questions to get at the high-level questions and themes, but there is no specific structure to the interview or to the answers. New questions can be asked based on the interviewee's responses. This particular method was chosen because the scope of knowledge was unknown, and other, more structured interviews do not allow for more flexible data gathering needed at the early stages of a project.

Users were selected from the defined users described in 1.3 Relevant Users. Interviews were performed in one of three environments: in-person, in a café, or over the phone with screen-sharing. Each interview
was allotted up to one hour. The only data gathered was notes of the conversation. The data is of course limited to the company where the interviews were performed, so there is likely some inherent bias toward the processes of the company.

Interviews were performed with six people in total: two validation engineers, a user experience strategist, a requirements engineer, a project analyst, and a user experience manager. Many technical details were shared about how the data is created and stored and why the data is so disparate. Some existing tools were demonstrated, but five of the six participants all expressed interest in some form of visualization or reporting for the data. In the case of the two validation engineers, both had done some form of custom visualization on their own using an export of requirements and use case data. One was taking Excel exports and importing them into Tableau for analysis. The other took a huge export of use cases and attempted to relate it back to a Marketing Requirements Document using a massive manually created mind map. The user experience manager specifically had interest in a tool that would allow for ideation and brainstorming that could then be structured and developed into early product development data. The interviews made it clear that there is a huge amount of data that needs to be better understood by making it accessible and visible.

1.3 Relevant users

There are many types of users such a tool can affect. The initial data gathering showed that design and development data is a large body of work that is dependent on several roles in a large organization:

- UX professionals, from designers to architects to researchers, gather data and design the user experience of a system.
- Strategic planners create high-level data, such as roadmaps and vision, that inform the design process.
- Requirements engineers use higher level usage data to create and document systems requirements.
- Software and hardware architects utilize requirements and use cases to design systems.
- Validation engineers test use cases and verify that a system is functioning correctly.
- Project analysts support project management, scheduling, and risk during product development.
- Managers and executives are concerned with the high-level status of a project, including the user experience.

However, the use of such a tool would not be limited to these users. With product development data stored in one place, it is possible to abstract the data to a level that an even broader base of users could understand, including sales, marketing, and finance.

1.4 Context of use

UX early product development artifacts are created in a wide variety of contexts. Meetings can occur in conference rooms, design spaces, cafeterias, and other locations. UX artifacts can be created in these
meetings, with people moving papers around, writing post-its, and discussing solutions to problems. The context here varies from team to team, with some teams designing artifacts using primarily paper and post-its, some teams working only from their laptops, and of course many teams working somewhere in between those two examples. Fully co-located teams have everyone on-site, but some teams do not have this luxury and instead coordinate by sharing screens and documents with each other over distance. Some artifacts are created by people alone in cubicles, conference rooms, coffee shops, or even at home. The design methods change as well, even if the same artifacts are being created. For example, a user may use lots of post-its and whiteboarding to create artifacts in a meeting, but the same user may tend to stay behind the computer when alone, creating the artifact digitally with minimal or no use at all of physical artifacts. This existing behavior was important to understand for developing the visualizations for the tool.

1.5 Vocabulary and definitions

Some vocabulary and terms used in the context of product development, user experience, and data visualization will be used throughout this paper.

The product development and user experience vocabulary are as follows:

- **Artifact.** The term “artifact” is used in this paper to refer to any document or deliverable in a design and development process.
- **Indicator.** Indicators are used in product development to indicate some attribute of an artifact, such as due date or testing status.
- **Status.** Status refers to the completion level of an artifact.
- **Landing zone.** A landing zone sets minimum, target, and outstanding expectations for an artifact in product development.
- **User experience landing zone.** A UX landing zone takes the landing zone concept and applies it to usages. It is a way to set expectations for usages, establish a metric to test against, and prioritize usages.
- **Value proposition.** A value proposition is an overarching statement that represents what users value. It is written from the end-consumer perspective, using a first-person voice.
- **Usage category.** A usage category is simply a category of usages. It provides a way to generalize about a group of usages and to relate this group to their common value proposition(s).
- **Usage.** Generally used as the title of a complete usage model [19], a usage is a small statement that further defines the value propositions in a usage category. It provides a common ancestor for the data that follows, including usage requirements, usage scenarios, and use cases.
- **Use case.** A use case defines interactions between an actor and a system to achieve a goal [2].

The data visualization vocabulary are as follows:

- **Partition layout.** Also known as an adjacency diagram, the partition layout is a space-filling hierarchical representation of data [11]. The result is a left-to-right or top-to-bottom hierarchy of nodes.
● **Sunburst layout.** The sunburst layout is equivalent to the partition layout, only in polar coordinates instead of Cartesian coordinates [11, 21].

● **Treemap layout.** The treemap layout is a type of enclosure diagram, where containment is used to show the hierarchy of data [11, 14].

● **Hierarchical edge bundling.** Hierarchical edge bundling is a method of showing adjacency relations in a radial layout [11, 12].

● **Zoomable treemap layout.** The zoomable treemap layout is a variation on the treemap. In this paper, a two-level layout was used, meaning only two layers are visible at a time.

● **Node.** A node represents a piece of data in a visualization. In the visualizations shown in this paper, nodes represent artifacts of product development.

● **Root node.** The highest-level node in a hierarchy of data is the root node.

● **Leaf node.** A node that has no children is a leaf node.

● **Parents, children.** All of the data referred to in this paper is hierarchical in nature. A parent is simply referring to a node in relation to another node. A parent is higher in the hierarchy than its children. A child always has a parent, and a parent always has children. Not all parents have parents (i.e. the root node), and not all children have children (i.e. leaf nodes).

● **Inclusion relations.** Parent-child relations are called inclusion relations [12].

● **Adjacency relations.** Non-parent-child relations are called adjacency relations [12].

● **Relationships.** A relationship refers to the connection between two nodes in a visualization. Visualizations can have one-to-one, one-to-many, and many-to-many relationships. In the visualizations shown in this paper, all relationships are in the form of parent-child relationships.

● **Ancestors.** In a hierarchy of data, an ancestor of a node is a related node that is at least one level higher than that node. A parent is an ancestor that is only one level higher than a node. For example, an ancestor could be a parent’s parent, or a parent’s parent’s parent.

### 2 Theory and related work

The early product development data used in this research exhibits two types of relationships: inclusion and adjacency [12]. Inclusion relations are found in hierarchies, or trees, of data. All relationships in our data are formed by parent-child relations. Adjacency relations, on the other hand, are not limited to the parent-child tree structure. Data can be related to any other data, regardless of hierarchy. The adjacency relations in the data used in this research are not this flexible; in our data, the adjacency relationships are only relations to other parents or other children than are described in the inclusion relationships. The right way to think of our data set is primarily built with inclusion relations, with some additional adjacency relations that extend these inclusion relations but adhering to the strict parent-child relationships.

Because our data is hierarchical in nature, we will be evaluating different methods to show this hierarchical data. A number of hierarchical space-filling visualizations have been developed over the years, including node-link diagrams, indented trees, adjacency diagrams, and enclosure diagrams [11].
Visualizations more suited to showing networks have also been developed, including force-directed layouts, arc diagrams, co-occurrence matrices, and hierarchical edge bundling [11, 12]. Each of these techniques has strengths and weaknesses, which have been discussed at length in many published works [4, 11, 12, 14, 15, 16].

Some good work has been done in recent years around the topic of evaluation of information visualizations. Shneiderman and Plaisant proposed some guidelines for multi-dimensional in-depth long-term case studies of information visualizations [18]. Plaisant has addressed the challenges of information visualization evaluation in another publication, where she recommended gathering struggle and success stories, promoting field studies, and investigating new evaluation procedures [17]. The role of aesthetics in visualization has also been considered, and modifications have been proposed to existing task timing and goal-fulfillment methods to account for aesthetics in information visualization [6].

Research around pragmatic casual information visualization (PCIV) has produced some new metrics - motivation; immediate usability; aesthetics, fun and novelty; and perceived utility - by which to evaluate more casual information visualizations [20]. Recent focus by the information visualization community on evaluation techniques for information visualizations have produced a large body of research on the evaluation of information visualizations. A good example of this is the BELIV conferences in 2006, 2008, 2010, and 2012.

Our research is more concerned with comparing these different techniques when faced with large data sets with complex relationships. Many of the visualizations we surveyed have been applied to some sort of data analysis problem in their respective publications, and some usability tests were performed. For example, treemaps have been applied to equipment readiness in the US Marine Corps [17]. While we did find a call for papers addressing the question “Given two visualization techniques, which is best for the particular environment and task at hand?” [1], we found no work addressing this question specifically. Our research comparing different visualization techniques for a large set of business data can be considered a small first step in that direction.

3 Development of prototype

3.1 Choosing visualization software

There were some technical decisions to make as well. The most significant technical decision to make was how the visualizations would be developed. The following options were evaluated:

- Tableau is a licensed software that builds visualizations based on data.
- Processing is an open-source programming language and development environment that simplifies and enabled the creation of advanced images and interactions. It can and has been used for data visualization.
- JavaScript Infovis Toolkit is an open-source data visualization library for the web.
- D3.js is a JavaScript data visualization library that allows for the manipulation of documents
based on data.

- Raphaël.js is a JavaScript library for drawing vector graphics on the web.
- Google Charts is a set of customizable charts that can be created and viewed in a web browser.
- Many Eyes is a collection of data visualizations built in Java for the web. Data sets can be uploaded to these visualizations.

There are other charting and data visualization tools available, but the above had the best feature sets at the time of writing.

The requirements for the data visualizations built for this research were as follows:

**Compatible with all devices.** If any further development were to occur after this research is complete, a compatible visualization will provide flexibility, should follow-up work be done on desktops, laptops, tablets, or phones.

**No licensing.** Any future development should not be limited by software that must be purchased.

**Built for data.** It should be simple to add, scale, and customize data. There should be no limitations on the types, format, or complexity of the data.

**Flexible.** It should be possible to extend and transform visualizations as needed.

**Visualizations supported.** It is important that a diverse set of arrays are supported or at least possible.

Based on these requirements, d3.js [3] was ultimately chosen, simply because it met all of the above requirements and the other options did not. D3.js is compatible with all recent browsers, which means it works on the majority of computer and tablets regardless of operating system. D3.js is open source, so there are no issues with licensing. Every visualization in d3 is built on top of data, and changes can be made easily based on changes in the data. Since d3.js manipulates the document object model (DOM) of web pages and binds data to it, anything built with d3.js can easily be extended with additional attributes, interface elements, events, behaviors, and animations that are already possible with HTML, CSS, SVG, and JavaScript. A huge number of visualization are available and supported by d3.js, and custom visualizations can easily be created. Other advantages include an active community, good documentation, continual improvement, and a large and growing repository of examples.

### 3.2 Data model

Before building the visualizations, some test data was gathered from a set of documents from an old product (from 2007) no longer in development. The structure of the data was as follows:

- Product name
  - Value propositions
    - Usage categories
    - Usages
      - Use cases

There were several other types of data available, but these core artifacts were chosen for the purpose of
this research because they had clear structure and they were necessary artifacts of the user experience side of product development. All of the above artifacts are defined in 1.5 Vocabulary and Definitions.

The data is hierarchical in nature - however, some complexities emerge with usage models and use cases. Usages can have multiple inclusion relations, which means that they can be related to multiple usage categories and value propositions. Similarly, use cases can be related to multiple usages. This mildly complicates traditional hierarchical layouts of data, as elements must be repeated visually to show these inclusion relations.

### 3.3 Choosing visualizations to create

A survey of existing visualization techniques was performed [3, 4, 5, 7, 11, 12, 14, 15, 16, 20, 21, 23]. It was not clear from this survey which visualizations were best suited to presenting the data to the user. Hence, it was decided that multiple visualizations will be tested to gather insights and compare different techniques.

Four visualizations were chosen: partition [11], sunburst [21], cascaded treemap [15], and hierarchical edge bundling [12]. Examples of each are shown in Figure 1 below.

![Visualizations](image)

Figure 1: The partition (top left) shows a hierarchy of data with the root node on the far left, exposing the hierarchy moving rightward. The sunburst shows a hierarchy of data with the root node at the center, exposing the hierarchy moving outward in radial coordinates. The cascaded treemap (image taken from paper introducing cascaded treemaps [15]) is a type of treemap that that displays nodes as rectangles and represents hierarchy using a small position offset, which adds depth to the visualization. Hierarchical edge bundling shows adjacency relations between nodes in a circular format.

The partition was chosen because it shows a clear linear hierarchy in Cartesian coordinates. Separation between nodes is clear and text is rendered horizontally and often with ample room for at least one line of text.

The sunburst was chosen because it is technically equivalent to the partition, just in radial coordinates. However, this mathematical difference results in a visualization that looks very different than the partition visualization. One important result of the radial layout is that, while it cannot fill a rectangular window completely with data like the partition, it gives an increasing amount of area for each level of the
hierarchy of data moving outward from the root node in the center of the visualization. This makes it more fair in space distribution than the partition, which gives the same amount of area for each new level of the hierarchy, which makes lower levels unreadable due to the high density of nodes in a column. Some existing work [5] showed the Cartesian layout of data was empirically more effective for some tasks than the radial layout, the results were not conclusive, so we thought the comparison between the partition and sunburst would provide insight into the effect of visual representation on identical data sets with identical structure.

The cascaded treemap was chosen because treemaps can show a high density of data and the relationships between those data. The cascaded version in particular is extremely space efficient, and it shows hierarchy the most clearly of the treemap visualizations surveyed [15].

Hierarchical edge bundling was chosen because it shows the many-to-many relationships of the data set in a completely different way than the other three visualizations, which all display data hierarchically. Elements are not repeated as they are in the other visualizations to convey multiple relationships - instead, lines are drawn between single nodes and their adjacent relations.

Each visualization was chosen for how well it suited the data set. We chose these particular visualizations because they seemed the most practical for a business setting. There are other viable visualizations that we chose not to build, but certainly could be used for visualizing inclusion relations and adjacent relations, such as node-link diagrams, circle packing, force-directed layouts, arc diagrams, and co-occurrence matrices [11]. Node-link diagrams waste space in higher levels of the hierarchy and make heavy use of lines to indicate relationships; the partition layout communicates the same exact structure and data in a less visually cluttered manner that also allows for customization of the size of nodes. Circle packing wastes a fair amount of space and reduces the amount of text that can be displayed on each node; the treemap communicates the same data and structure more efficiently and thoroughly. Force-directed layouts are visually interesting, but they make finding relationships difficult and do not have text labels to indicate the data contained in a node. Arc diagrams show relationships in a similar manner to hierarchical edge bundling - the only difference is that the nodes are laid out vertically or horizontally in arc diagrams and radially in hierarchical edge bundling. Ultimately we chose the radial layout because we could fit more nodes in a circle than we could in a column or row of nodes. Co-occurrence matrices did not align with the types of relationships we were looking to visualize. A better scenario might be using a co-occurrence matrix on a larger data set to discover emerging relationships between artifacts. One particular scenario that comes up in product development, use-case to test-case mapping, is something to which co-occurrence matrices would be well-suited; however, this is out of scope of this research.

Due to time constraints, not all of the four visualizations were ready for testing. Hierarchical edge bundling was ultimately not tested due to difficulties with the implementation. Cascaded treemaps also ran into difficulties and required a customized treemap layout that was not natively supported in d3.js. Nested treemaps were possible to create, but prior work has shown that cascaded treemaps are superior.
to nested treemaps. A traditional treemap was not a good choice because it only shows leaf nodes, which was not adequate given the importance of the non-leaf nodes of the data used in this research. Hence, the decision was made to test a different kind of treemap, the zoomable treemap, to explore users’ reactions. Rather than showing all the information at once, a zoomable treemap only shows one level of data in a hierarchy, and users have to navigate through. The particular version of the zoomable treemap implemented for this work shows the structure of the next level of data in addition to the one level of data already shown. At the time of user testing, the prototype had functioning partition, sunburst, and zoomable treemap visualizations, all shown below in Figure 2.

![Figure 2: The versions of the partition, sunburst, and zoomable treemap visualizations that were ready in time for the usability testing.](image)

### 3.4 Building additional features onto visualizations

Three features were included beyond the data visualization itself: a collateral viewer for showing other data associated with an artifact, a hovering feature for showing multiple instances of the same artifact, and an indicator layer showing the UX health of an artifact. The collateral viewer displays more detailed information for any artifact, including a full description of the artifact, its UX health value, and any files (i.e. documents, images, sketches) associated with that artifact. The hovering feature was implemented on the sunburst visualization and it gives the user the ability to explore how artifacts are repeated through the hierarchy, showing the many-to-many relationships between artifacts, as well as highlighting the parent of each of these artifacts. The UX health indicators are a proposed concept for this visualization to communicate the aggregate UX health of an artifact based on testing and evaluation of itself and lower-level artifacts. Though these features were kept simple in this prototype, these are good examples of features that should be built onto the visualizations to provide more functionality and information for the target users.

### 3.5 Technology requirements

Since the tool is developed for web browsers with HTML5, CSS3, JavaScript, and d3.js, the technology requirements are very basic. The hardware and operating system are not important - what is important is the browser. Recent versions of Safari, Chrome, Firefox, and Opera work, as well as Internet Explorer 10. Android tablets and iPads are also compatible. Mouse and touch input are both accepted.
3.6 Limitations of prototype

It is important to note some of the limitations of the prototype that was tested. There are some features which ideally would have been developed further before testing, but were not due to time constraints. These include improved icons and buttons, legends, improved text rendering in SVG, categorization and labeling of nodes, and other types of visualizations. As mentioned before, having the cascaded treemap and hierarchical edge bundling visualizations ready would also have made the resulting prototype and the usability study more robust.

4 User testing methodology

After the prototype was developed, user testing was performed. The testing measured users’ understanding of the prototype and the perceived value of such a tool. The goals of the testing are to evaluate users’ understanding of the prototype, find which visualizations are suited for which activities, and to gather insights for future development of the tool.

4.1 Research questions

- What attributes of the visualizations were successful and unsuccessful with users?
- How do the visualizations perform for certain goals?
  - Ease of understanding the data, its structure, and relationships between data
  - Exploring, moving, and staying oriented within a data set
  - Finding more information about an artifact
  - Novel and fun visualization
  - Perception of value
- How do the different layouts of data affect users' understanding of the data?
- What were the biggest roadblocks for users in understanding and using the different visualizations?
- The testing will also be used to gather insights from participants.

4.2 Participant characteristics

As discussed in 1.3 Relevant users, any user interested in a product in development could potentially make use of the tool. Hence, recruiting was done via employee forums to get as broad a base as possible. One inherent benefit to testing internally at a company is that the intended users of the tool are employees, and the testing was performed on these employees. In addition, care was taken that each user had a different role in the company. The six final users tested each represented a different role at the company: chip design, quality engineering, user experience design, employee insights, technical talent management, and finance. Users had worked at the company anywhere from 1.5 to 30 years.
4.3 Test design

The test was performed with 6 participants total, with 2 pilots to check for flaws in task design, correct bugs in the tool, and provide some practice for the moderator before the real testing begins. The test was performed one user at a time with one moderator. The moderator’s responsibilities were as follows:

- Set up the room, computer, and prototype
- Hand out waivers
- Ask the pre-test questions and record answers
- Introduce the user to the system
- Setup screen and audio recording
- Give the user the tasks
- Encourage think-aloud feedback
- Prompt users as necessary
- Take notes during the tasks

Exploratory, qualitative comparison test

- Within-subjects, so each visualization will be tested by each participant
- Counterbalanced to mitigate learning effects
- Some tasks are specific, some are probe tasks
- Encourage user to talk out-loud through tasks
- No explicit success, since test is comparison/exploratory; however, a positive result is the participant having a clear understanding of each visualization and its features

Some tasks test specific features or attributes of the tool, while others are probe tasks to gather insights. Each visualization was tested by each participant, albeit in different orders. Each participant was given a brief, scripted introduction to the tool before seeing the tool. The introduction script can be found in Appendix A.

Before presenting the user with the prototype, some background information is given to help the user understand the problem space. Though there is potential for learning effects to be introduced, this was deemed necessary, as the pilot users were not familiar with the data model or even the data itself and thus had difficulty speaking in detail about what they were seeing. This background information is printed, explained by the moderator, and given to the user as a reference for the duration of the test.

The first piece of background information is a simple diagram that provides the terminology and structure of the data the user is about to see. This diagram, shown in Appendix B, shows the types of artifacts the participant will encounter in the visualization and explains the relationships between these artifacts. The second piece of background information is an experience definition document, which is a document used to communicate the desired user experience of a product across the teams working on the product. This document is a different way of showing the data shown in the visualizations. The user
was allowed to flip through the document, and the moderator walked the user through the document and its contents to better understand an alternative to the data visualization that the user would see next. The definitions of the types of artifacts the participant will encounter in the tool were also provided. These definitions can be found in Appendix C.

4.4 Task list

Visualization 1: Partition

Task 1: Assess initial reaction to and understanding of visualization
- Time: 3 minutes
- Question/prompt for participant:
  *Tell me your initial reaction to this visualization. Thoughts on the data, hierarchy, relationships?*
- Moderator role:
  Allow participant to look at and freely navigate the visualization. Encourage navigation and interaction if user is stuck. Reiterate talking aloud.

Task 2: Assess interaction; assess extent of health indicators understanding; gather expectations for insights that could be learned from indicators, like these or otherwise
- Time: 3 minutes
- Question/prompt for participant:
  *Find the item labeled “Sharing”. The yellow indicator means there is a warning associated with this item. Do you have any idea why? (Later) What kind of indicators would you be concerned with in your work?*
- Moderator role:
  Enable indicators on the visualization. Do not explain their purpose (next task you will).

Task 3: Evaluate usages against UX Landing Zone to assess participants’ understanding of health indicators
- Time: 4 minutes
- Question/prompt for participant:
  Now that you know what the indicators show, you will perform an activity. Are you familiar with a UX landing zone? A UX landing zone is a way of measuring the overall user experience of a product by testing to see if the product is meeting the expectations assigned to its usages. For each usage, minimum, target, and outstanding outcomes are specified. Given these indicators and the data in the visualization, please assign a status to the UX landing zone of the “Browse news” usage.
- Moderator role:
  Explain that indicators are communicating the health of the different artifacts as a result of testing and evaluation.
  Hand the UX Landing Zone activity handout to the participant.
Introduce concept of UX landing zone, have participant grade a particular usage against its landing zone. Does the usage reach minimum, target, outstanding? Does it even meet minimum?

- Note: the usage name and UX landing zone handout cannot be shown in this paper because the data cannot be shared outside of the company.

Visualization 2: Sunburst

Task 1: Assess initial reaction to and understanding of visualization
- Time: 3 minutes
- Question/prompt for participant:
  
  Tell me your initial reaction to this visualization. Thoughts on the data, hierarchy, relationships?

  Moderator role:
  
  Allow participant to look at and freely navigate the visualization. Encourage navigation and interaction if user is stuck. Reiterate talking aloud.

Task 2: Assess understanding of relationships shown while hovering
- Time: 3 minutes
- Question/prompt for participant:
  
  When you move the mouse around, you will notice the item beneath the mouse changes color, and sometimes more than one item will change color.
  
  First, tell me what are the other highlighted items and why they are highlighted.
  
  Second, tell me what this means. Why do some items have highlighted counterparts while others do not? What does this mean for an item's children and parents?

  Moderator role:
  
  Encourage participant to think and answer question, but don't push too hard. Also, be careful not to give away any answers in the questions.

Task 3: Evaluate interaction and assess collateral viewing mechanisms
- Time: 4 minutes
- Question/prompt for participant:
  
  I now need you to find some more information about a particular artifact. There is a use case named Setup LR under the “Simple and Cool” value proposition. Please find this usage category and access more information about it.
  
  (Once collateral view is up) Tell me what kind of data you are seeing. Check the studies and tell me how the product has changed between iterations.

  Moderator role:
  
  Do not give away the location of the use case. Let the participant hunt for it. If time gets short help the participant.

Visualization 3: Treemap

Task 1: Assess initial reaction to and understanding of visualization
- Time: 3 minutes
● Question/prompt for participant:
   *Tell me your initial reaction to this visualization.*
   *Thoughts on the data, hierarchy, relationships?*

● Moderator role:
   Allow participant to look at and freely navigate the visualization. Encourage navigation and interaction if user is stuck. Reiterate talk aloud protocol.

Task 2: Understanding type and structure of data and interaction of zoomable treemap

● Time: 3 minutes

● Question/prompt for participant:
   *Start moving around the visualization.*
   *(After drilling down a few levels) Now stop and tell me how many items you currently see on the screen.*
   *What do the white lines indicate? What does the top bar indicate?*
   *Please continue to click deeper into the visualization.*
   *(Once at final node) Now, return to the highest level of the visualization, where you started.*

● Moderator role:
   Only push the participant if absolutely necessary to complete the task.

### 4.5 Environment, equipment, and logistics

Testing was performed in a conference room. Users sat in front of a 23-inch height-adjustable widescreen monitor, and the seat and monitor were adjusted for the user prior to the start of the testing. A laptop is connected to the monitor, but this display is turned off for the duration of the test. Audio and screen recording was done using Quicktime, which has built-in screen and audio recording. The moderator sat beside the user, taking notes and assisting with the tasks.

Each test was allotted one hour and began with a five-minute period for introductions and time for the user to read and sign the waiver. Then five more minutes were used to give a brief high-level explanation of the tool’s purpose (read from script by participant) and to ask three basic screening questions. Up to 30 minutes was allowed for the tasks (10 minutes per visualization). Following the tasks, ten minutes were provided for any additional feedback from the user. The final ten minutes were set aside for the moderator to prepare for the next user.

### 4.6 Data collected and evaluation measures

Data was collected primarily as notes during the testing. Further notes were gathered from the screen and audio recordings, which were watched two more times after the study was finished. Data was collected in the context of answering the research questions. These data include:

- Quantitative: completion of tasks (yes/no)
- Qualitative: successful/unsuccessful attributes of the visualization
- Qualitative: user orientation
4.7 Limitations of user testing

The main limitations of the usability study stem from the limitations discussed in 3.6 Limitations of prototype. Features and interface elements that were not completed would likely test poorly and could potentially confuse some of the tasks for the users. Luckily, the bulk of the work was ready for testing, so we were confident that these shortcomings only detract slightly from the overall body of work.

5 Results and discussion

5.1 Partition

Task 1 for this visualization was made to assess the users' first reactions and initial understanding of the partition visualization. All six users had a clear understanding of the hierarchy and relationships of this visualization. The layout and hierarchy was clear - the term “bucket” or “section” was used to describe the five higher-level nodes (value propositions), and users understood that lower-level nodes were related to their parents leftward. This hierarchical relationship is shown in Figure 3.

Figure 3: The initial state of the partition visualization when users began their first task with the visualization. The hierarchy begins on the left side and moves rightward down the hierarchy. The right images shows the highlighting effect that occurs when hovering over a node with the mouse.

Four of the six users mentioned at some point while using the visualization that it was not immediately clear which nodes belonged to which ancestor in the hierarchy. Three users attributed this to the weight of the lines separating nodes, and two users attributed this to not having a way to see a node's ancestors.
or children visually. The only way to understand this parent-child structure was to trace the outline of the section with the eyes. The amount of effort this required varied between users. This difficulty was also observed in the sunburst visualization. One way to improve understanding of these relationships is by implementing larger and clearer separation lines between sections of the visualization. Another way to improve understanding is to use highlighting and some form of interaction, like clicking or hovering, to show a node in relation to its parents, ancestors, and children. Both improvements could be used together.

One concern while building the prototype was that users would have difficulty orienting themselves through the transitions. In the partition visualization, all of the users understood the interaction traversing down and up a hierarchy of data, and everyone had a clear sense of orientation. Figure 4 shows the visualization before, during, and after a transition.

![Image of visualization changes](image)

Figure 4: Changing focus from a lower-level node (left image) through a transition effect (center image) back to the root node (right image) in the partition visualization. Changing focus to another node is accomplished by either clicking a node or by clicking the top-level or back-one-level buttons.

Three of the users expressed concern with the boundaries of the partition visualization. Because the entire browser window was filled with data, and data is displayed in columns, these users expected and searched for some form of scrolling. Within the first minute of seeing the visualization, all users did come to understand, without help from the moderator, that there is no scrolling available or needed and that there is no other data on the screen. This is a relatively minor usability problem that is easily addressed by adding a frame or border to the visualization to denote the boundary.

All of the users discovered that data was repeated (see Figure 5), even though the visualization did nothing to inform the users of this fact. This is not a surprising result for the three users that had already seen the sunburst visualization, where repeated nodes are highlighted, before seeing the partition visualization. For those three users that had not already seen the sunburst, there was confusion as to why the nodes might be repeated. Two users correctly deduced that these nodes have more than one parent node. Another user, on the other hand, found the repetition disconcerting and viewed it as an indication that something was wrong with the product shown in the visualization.
Figure 5: Nodes are repeated in all visualizations shown to the users. The repetition occurs because lower level artifacts are related to multiple higher-level artifacts in various ways. The sunburst visualization, shown on the right, assists users in finding other instances of the artifact and shows their respective parent artifacts, while the partition visualization does not show the other instances of the artifact or other related artifacts.

Three of the users expressed being initially overwhelmed by the partition visualization, even having seen one of the other visualizations beforehand. However, this feeling quickly subsided for all but one user, who remained overwhelmed throughout the tasks with the partition visualization (this same user was also overwhelmed with the sunburst visualization). Different reasons were given for the overwhelming feeling, including large amounts of text, uniformity of the artifacts, and density of the nodes. Some of this could also be attributed to some users’ lack of in-depth knowledge of the types of user experience deliverables shown in the visualization.

This overwhelmed reaction lies somewhat in contrast to a sense of relief for two users who had just seen the zoomable treemap visualization, which only shows one level of the hierarchy of data at a time, instead of all levels at once. Three users specifically stated that they appreciated the ability to see a lot of data in a limited space, with two of those three mentioning that they don’t even need to click to traverse down the tree, since all the nodes and text were visible on the large monitor used for the user testing. This luxury would unfortunately disappear with a larger data set, but clearly there is some value in displaying a high density of information. It may be worth doing some simple testing of different levels of information density with different amounts of data to test boundaries and understand the tradeoffs.

During Task 2, users were shown an indicator layer on top of the visualization, as shown in Figure 6. Every user had the same initial reaction: that the colors were a form of categorization that showed relationships between all the nodes with a particular color. Once the concept of the indicators as a measure of “user experience health” was explained, users saw the data differently. The three users who had engineering backgrounds all mentioned that they were accustomed to the red-yellow-green indicators in other software they use. This familiarity could be utilized in designing meaningful status
indicators. If the tool will also be shared with others, it will be important that the colors used are meaningful to all target user groups. A good example of a confusing color was the light green color used as a “good” indicator in this test. Even when explained, all users were still uncomfortable with it and unsure of its relative position compared to the other indicators.

Figure 6: Indicator layer shows the user experience health of the artifacts shown in the partition visualization. User experience health is the result of testing performed on an artifact and/or its child artifacts.

Three important results came out of Task 3. In this task, users are given an activity based on the concept of a user experience landing zone and asked to grade a usage given the information (indicators, relationships, hierarchy, text, collateral) that they can find in the visualization. The first result is that one aspect of this task was not understood by any of the users. Users were supposed to examine the content of the visualization and see if the landing zone criteria was met. Instead, users simply made an educated guess based on the indicators. It was clear there was an almost 1:1 relationship expected by users between the indicators shown and the grade of the landing zone.

The second result of Task 3 is that there was variation in the interpretation of the indicators (see Figure 7). Three of the users thought indicators were determined as an aggregate of other indicators, and the other three thought that the indicators were determined for each artifact individually. The original intent was for the indicators to be aggregates of the indicators beneath them in the hierarchy. But this result brings up an interesting question that should be considered in future iterations of this design: should
indicators represent something unique in each artifact instead of simply aggregate lower-level indicators? The stance taken while designing this prototype was that an aggregation of lower-level indicators would provide clarity concerning the health of the artifacts at any level in the hierarchy, lowering the amount of thought required by a user to deduce the health of an artifact. Furthermore, a larger data set would make it very difficult for users to understand the health of higher level artifacts without some aggregation mechanism. Whether an aggregate or individual indicator approach is taken, it is clear that this will need to be explained to users in some way, as it is subjective and not intuitive one way or the other for users.

Figure 7: The highlighted usage was graded against a UX landing zone in Task 3 of the partition visualization tasks. Note that the data shown in the screenshot is fake - company data was used in the usability study.

The third result of Task 3 is that, even though the users did not understand one aspect of the task, they all made decisions based on the available indicators. Despite the usage having a “warning” indicator associated with it, all users graded the usage as at least meeting the “minimum” criteria for a user experience landing zone, with the exception of one user who thought the usage performed even better, at a “target” level. Four of the six users only evaluated the usage based on the indicators, not on the content of the usage or its child use cases. The other two users did look at the data, but had very different interpretations of that data. This emphasizes the importance of using a set of meaningful indicators that work for the target users. If it is possible to deduce from a “warning” indicator that a usage is at an acceptable state, then the indicators need to be adjusted further and re-tested so this does not happen. The only indicator taken as completely unacceptable was the “fail” indicator. Perhaps a better approach to solve this kind of problem is an automated rule-based system, if possible, so the indicators are not left to human subjective interpretation at all. For example, some failure or warning rate of low priority usages or use cases may be acceptable and still allow an artifact to meet a minimum requirement. Even in this case though, some color or indicator will need to be chosen to communicate the tool's grading, and that indicator will need to be easily and clearly understood by users.
5.2 Sunburst

Task 1 for this visualization was made to assess the users’ first reactions and initial understanding of the sunburst visualization, shown in Figure 8. All users tested expressed a lack of familiarity with this particular representation of data. The emotional reactions associated with this lack of familiarity were varied. Three of the users had positive reactions and found the visualization different, interesting, intriguing, attractive, and fun. The other three had negative reactions and expressed skepticism as to the utility of the layout of the data. Within the three minutes allotted to Task 1, five of the six users said they now were more comfortable with the visualization and could continue to use it - only one user went the other direction and said that she is unable to understand or use the visualization.

Figure 8: The initial state of the sunburst visualization when users began their first task with the visualization. The hierarchy begins in the center and moves outward down the hierarchy.

Five of the six users understood the hierarchy and structure of the data; one did not understand the relationships or structure at all. Two users even understood that the structure of the data was identical to the partition visualization, just displayed radially. Four of the six users mentioned at some point while using the visualization that it was not clear which nodes belonged to which ancestor in the hierarchy without tracing the outline of the section with the eyes to confidently determine structure. Even after discovering or being shown the hovering interaction which highlights identical nodes and their parents, these four users were still not completely satisfied and thought more could be done to visually show the
ancestors and children of a node. Two users attributed this to the weight of the lines (shown more closely in Figure 9 below) separating nodes, and the other two users attributed this to not having a way to see an entire section (a node, its ancestors, and its children) visually grouped. This identical result was found when testing the partition visualization. Both options, increasing the weight of the lines separating nodes and visually grouping nodes, are simple improvements to make, though care should be taken, especially with larger data sets with more levels in the hierarchy, to show only meaningful separation and grouping. If the visualization were to have 10 or more layers of data, highlighting all the ancestors and children of a node may be excessive. Further testing with differently sized data sets is required to determine this.

Figure 9: Thin lines separate nodes and groups of nodes from each other in the partition and sunburst visualizations.

All users stated the utility of the relationship hover (see Figure 10 below) and expressed interest in having the same type of interaction in the partition layout, regardless of whether the user saw the partition visualization before or after. Two users specifically commented that they were investigating the relationships between nodes on their own. All users recognized that the hovering was showing the same node located in multiple locations in the visualization. After being introduced to the hovering in Task 2 (or if the user discovered the feature in Task 1), all users commented that they wanted to see more relationships (ancestors, children, other relationships) shown in the visualization. The prototype only showed nodes of the same name and their respective parents, but not other related nodes.
Figure 10: The hovering interaction showing nodes with the same name highlighted. Nodes are repeated in all visualizations shown to the users. The repetition occurs because lower level artifacts are related to multiple higher-level artifacts in various ways. The sunburst visualization assists users in finding other instances of the artifact and shows their respective parent artifacts.

Two users noticed sizing differences between nodes and were unsure if the sizes were important. The prototype can change sizes of nodes based on some attribute of the data, but for this version of the prototype, nodes simply split the available space evenly. The relative space used was identical to the partition visualization, but no users commented on sizing in that visualization. It may be that the radial layout causes users to notice or look for spatial differences, and it is not as simple to compare relative sizes in a radial layout. If size is always interpreted as important, it may be beneficial to intentionally use size in the design to communicate some attribute of the data, rather than have it be used unintentionally.

The three users with engineering backgrounds all preferred the linear flow of data in the partition visualization over showing the data in this way, even though they all understood that the structure and content of the data was identical. One of these users was quite blunt regarding the layout, saying “show me the data and don't mess around.” All three of these users expressed concern with what value the sunburst layout of data provides over the partition layout of data. These reactions are important to keep in mind for future iterations of this work, especially for any aspect of the tool targeting an engineer-dense
An anticipated usability concern was the rendering of text at angles. Three users commented specifically on the difficulty of reading text at steep and near-vertical angles. However, this aspect did not prevent users from completing the tasks, and it did not elicit more than one negative comment. It is possible that the limited amount of text minimized this concern - users were able to read nodes when they needed, but they never had to read more than two or three words at any angle. It is not advisable to render any significant amount of text at angles [9], so this visualization is perhaps best suited to a birds-eye view of the data with a limited amount of text to indicate the content of each node.

5.3 Zoomable treemap

Many features of the sunburst and partition visualizations were not present in the zoomable treemap visualization (see Figure 11), which led to some negative results. However, many of these negative results were expected, as this visualization was designed to test a different set of features. Users who had seen the partition or sunburst visualization before seeing the zoomable treemap were already comfortable with the data. Hence, when moving through each level of the data, users knew to expect multiple levels of data and recognizing artifacts they had seen before. On the other hand, two users had not seen any data or visualization prior to the zoomable treemap, and these users had a difficult time understanding the overall structure of the data. These users first simply stared at the visualization to understand the structure. Both required prompting to move the mouse around and begin clicking nodes. Once these users realized they could interact with the visualization, they began moving down and up the hierarchy. Both users understood that they were moving down and up a tree of data, but by the end of Task 1 they were still quite unsure what they were looking at. This stands in stark contrast to the other four users, none of whom needed the three minutes allotted to Task 1 to understand the visualization, since they quickly understood the interaction and structure of the familiar data.
Figure 11: The initial state of the zoomable treemap visualization when users began their first task with the visualization. The breadcrumb section shows the root node in this initial state, the primary visible, labeled sections show the second level of the hierarchy, and the thinner lines within these sections show the structure of the next level of the hierarchy (but not any content in that next level).

Three of the users did not recognize the breadcrumb-style path of links, shown in Figure 12, that take the user to any level back to the highest level of the visualization. Once told how the breadcrumbs work, users began to use them. There are two potential problems here. Some users may be unfamiliar with or out of practice with the breadcrumb-style navigation, or the breadcrumb itself did not make it clear that it was a path of links back to the first level the users saw.

Figure 12: The darkly colored breadcrumb section at the top of the visualization shows a path of links (levels in the hierarchy) back to the root node. This tested poorly with users.
The movement from parent to child nodes was understood without question by four users, but not by two users. The two users who did not understand the interaction were unsure where they originated from in the visualization and where exactly they had landed. Both of these users went back and forth to be sure they knew their location within the data.

All users found this visualization to be overly simplified and to contain too much empty space. This was an expected reaction; however, users had some interesting ideas for what to do with the space. Five users compared the layout to Windows 8 or the Xbox, which both have flat tiles with content inside. All users expected content inside these “tiles”, such as metadata associated with an artifact or just a preview of important parts of the data. This idea is further discussed in 6 Conclusion and recommendations.

One feature of the zoomable treemap that tested positively was the use of thin lines to indicate the structure of the next lower level of data. All users understood this exactly as intended - even the two users who had seen no other visualization before the zoomable treemap. More testing should be done on other multi-level treemap visualizations (for example, the cascaded treemap discussed in 3.3 Choosing visualizations to create).

An artificial size difference based on a priority attribute (i.e. important of a usage) was built into the zoomable treemap visualization. The other visualizations evenly split the available space between nodes. Four users recognized the different sizing and wondered what these differences meant. This is a positive result for using sizing to express some attribute of the data, though users had no immediate intuition as to what the sizing means. (Unfortunately, two users of each of the partition visualization and two users of the sunburst visualization also wondered if sizing was important when it was not intended to be. An important limitation of sizing is discussed in 5.4 Results relevant to all visualizations.) Sizing could be used in one of three ways with these visualizations: using sizing to express some attribute of the data and explicitly showing users what sizing indicates, using sizing but not explicitly showing users what is indicated, or not using sizing at all to avoid confusion. Explicit sizing may be helped by limiting sizing to a few easily comparable sizes - that way users are not confused by nodes that are sized similarly. Sizing without an explicitly stated definition could be used to give some nodes more visual weight, based on some attribute of the data, to make those nodes easier to see. It may be that sizing is too ambiguous to use for an important or precise attribute of data - if sizing is used, some testing is highly recommended to be sure that sizing is communicating what is intended.

5.4 Results relevant to all visualizations

Across all visualizations, 5 of the 6 users did not click on the first visualization they used until prompted to do so. In the sunburst visualization, three users did not even grab the mouse to move around the visualization, and hence did not see the hover effect until prompted to do so. Two users even performed all the tasks given to them on the partition without clicking on the visualization once, and another user simply said that click was a “last resort” if he cannot find the information he is looking for otherwise. It seems that some users do not intuitively expect visualizations to be interactive. A similar sentiment was
noted by Mike Bostock, creator of the d3 visualization library used in this project, on a podcast when asked about an interactive visualization he created for the New York Times [22]. It is important to note that, for the partition and sunburst visualizations, all of the data was visible and readable without any interaction, so it may be that a visualization that had even more data but was not readable (i.e. small shapes without text) would encourage a user to start clicking around to find a way to access and read this data. However, that is not a complete solution - a good solution needs to work in all cases with small and large data sets. In order to provide an affordance to click and interact, one option is to implement a small set of custom buttons that the user is shown upon hovering over a node. If there is no good way of encouraging interaction via an affordance in the interface, it may be necessary to prompt the user to click and hover around a visualization upon first use, perhaps using a lightweight tutorial or walkthrough. In any case, this seems to be an outstanding problem in the field of interactive visualizations.

The buttons (see Figure 13) used in the partition and sunburst visualizations delivered mixed results. While this study was more focused on users’ understanding of the visualization itself, and not the interface buttons, the results are still worth discussion. The back button were understood and utilized by all users without prompt. When asked what behavior was expected, all users assumed it performed exactly as a back button in a web browser. Since the prototype was built in a web browser, several users attempted to use the browser’s back button before the interface’s back button, which was not tied into the functionality of the visualization. Future versions of the tool should definitely integrate the browser’s back and forward buttons so users are not confused by unexpected behavior. There was hesitation to use the top-level button - one user stated that she was “unsure” what the button did and hence did not use it. Two users expected the top-level button to take them up one level in the visualization. Based on the users’ reactions, it was clear users did not have a clear expectation for what the top-level button does. Those users that did click the button were all somewhat surprised at first, then quickly understood that they were back at the top level. Despite the quick understanding of those who did click the button, the overall hesitation shows the button is not familiar or intuitive and should be improved. One option would be a Home button, which users may understand as a button that takes them back to the initial view of the visualization. If no immediately familiar or intuitive button can be found, the button should be clearly explained in an initial walkthrough or tutorial.
Figure 13: Blue arrows indicate the top-level, back-one-level, collateral, and indicator buttons. Users were not comfortable with the top-level, collateral, and indicator buttons, and most did not interact with them until prompted to do so.

Only three users used the collateral button without being prompted, and these users only used it in the sunburst visualization. One of these users thought she made a mistake when the collateral viewer slid into view (see Figure 14 below), while the other two users understood the concept and continued to use the button on other nodes. No users used the collateral button or the indicators button in the partition visualization, even though they used the same style as the back and top buttons. This is a very discouraging result, as users did not intuitively understand how to find more information about an artifact. Two users mentioned that they saw the button but did not think to interact with it. It is difficult to deduce why exactly users did not interact. It is possible that this is part of the overall hesitancy to interact with the visualization discussed at the beginning of this section. Other contributing factors may be that the button did not have a clear affordance to be clicked and that it was too visually similar to the rest of the visualization. In the case of the indicators button, the bottom-left placement inside the visualization may have also contributed to its poor performance.
There were some unique effects of the structures of the visualizations on how the users consumed the data. Users of the partition visualization tended to read the data left to right and top to bottom. Users of the sunburst visualization read data in many different orders - some started in the center and moved outward, some started with the biggest nodes and then looked at nearby nodes, and some started from outside nodes and moved inward. Users of the zoomable treemap visualization were forced to read from high levels down to lower levels, since only one level is visible at a time. Note that these results are gathered from the qualitative and talk-aloud nature of the study - more rigorous testing would be needed to prove these results more conclusively. Regardless, it is clear that even among hierarchical visualizations, there is a good deal of variation between how users consume the data. This effect on users’ understanding of the data should be considered in the design of a visualization.

The sizing of nodes in the visualization produced a variety of results between participants and visualizations. In all visualizations, users wondered if sizing was important, even in the visualizations where sizing was not used to indicate any particular attribute. When nodes are given an equal amount of space in their level, the size of a node is dependent on the number of its siblings, the number of its children, and the number of other nodes in the visualization. So sizing is communicating that a node has a certain number of children and a certain number of siblings, each with their own number of children, relative to other nodes in the visualization. This means that sizing cannot really be “disabled” in these visualizations. If sizing is changed to be based on some attribute, then that attribute will determine the relative size of the affected nodes, and the number of children and siblings. In other words, sizing always communicates some aspect of the data, even when not intended. It may in fact be better to have sizing indicate some attribute simply to avoid the confusion of having sizing indicate something else unintentionally.

Three users fully understood that artifacts with the same name are associated with multiple artifacts higher in the hierarchy after being shown the hover feature of the hover visualization. Unfortunately,
that still left three users who never quite understood why there were artifacts that had the same name. Based on the verbal reactions of these users, it appears possible that these users did not understand the concept of many-to-many relationships and that these kinds of relationships are both permitted and necessary for this kind of data. This is important to note for any training or walkthrough that is created for these visualizations for users not familiar with this type of data. Alternative visualizations may also help users to understand the nature of these relationships. In particular, the hierarchical edge bundling visualization, discussed in 3.3 Choosing visualizations to create, shows unique nodes and how they relate to each other.

Though it was not asked for specifically, all users gave some form of feedback on the aesthetics of the visualization. Four users found at least one visualization boring, saying that the visualization is “all grey” and “uninspiring”. These comments disappeared when indicators were enabled, or when users started hovering nodes with the mouse. Perhaps the initial state of the visualization should show some attribute of the data that would be interesting to a broad audience, as well as indicate how to add more attributes to the data. Three users also expressed higher expectations for the interface, saying that they were accustomed to Xbox, Apple TV, Roku, iPads, iPhones, and Windows 8. Since this is a prototype there is not much to be said in reaction to this, other than to note that quality, compelling interfaces on the market have raised the expectations of users.

Other unexpected feedback came from two users who critiqued the data itself. The data was set up to reflect a product early in its design process, with some content finalized but much still being changed and expanded, and this intent was described to the user in the introduction to the tool. An example of this feedback occurred when both users were unhappy with the constituent usages of certain value propositions, especially the “Simple and Cool” value proposition. Both were unconvinced that the usages shown for “Simple and Cool” would actually make this product simple and cool. This was great to observe, as these users were actually starting to use the tool as it could be used by a team, critiquing and data and providing suggestions for changes.

Three users expressed delight and relief upon seeing photos in the collateral during Task 2 in the sunburst visualization. While this is by no means conclusive, this result suggests that users very much want to see visual aids when moving around these large data sets. Since much of this design data has visual aids associated with it (sketches, renders, pictures), a more compelling version of this prototype could be implemented that puts visual content more front-and-center. The images could fill part or all of the area of nodes, and this could improve the browsing experience, especially for those users who are new to the data and/or exploring it out of interest (not for a specific work task).

Many of the users' thoughts and opinions were stated in comparison to one of the other two visualizations they had been shown. These comparisons were synthesized into the following:

- Users coming from the zoomable treemap to either the partition or sunburst were pleased to see greater information density, a more birds-eye view of the data, and the structure of the data.
- Users coming from the partition to the sunburst were pleased with the hovering, which provided
them with an additional way to make sense of the data beyond its structure.

- Two of the users coming from the zoomable treemap or partition to the sunburst enjoyed the aesthetics of the sunburst. The other visualizations received no such comments.
- Users coming from the sunburst or zoomable treemap to the partition liked the logical, linear, easy-to-read structure of data compared to the other visualizations.

Since each visualization had clear strengths and weaknesses, using the visualizations together and playing to their respective strengths may be the right strategy in building a larger, more comprehensive system, rather than simply building on one concept.

5.5 Summary of results

Successful attributes. There were many successful attributes that should be considered for inclusion in future prototypes of the tool. These attributes are as follows:

- The tree-like structure of the data was well-understood in all three visualizations.
- The usability testing showed the importance of carefully choosing particular layouts, as users do not understand the data in the same way across visualizations, even though the data and structure was identical in all three visualizations.
- The hovering interaction received very positive feedback. Users wished this interaction was available on the partition visualization, instead of just the sunburst visualization.
- The partition visualization was intuitive, clear, linear and familiar, which was appealing for several users.
- The sunburst visualization received positive feedback for showing data in an interesting and compelling way. Users were more engaged with this visualization than any of the others.
- Even though the sunburst was unfamiliar, users understood the structure and interaction.
- The hovering interaction was well-understood and considered as helpful by all users. All users expressed interest in having this extra dimension of information in the partition visualization.
- The separation between nodes and sections of data was not clear enough. Thicker lines should be used to denote boundaries and separate nodes visually.
- Several users did not attempt to click the visualizations on their own - they had to be prompted. This is a major concern that needs to be addressed.
- Users began to critique the data without prompt. This was afforded by the bigger picture view of the data.
- Intentional sizing differences between nodes in the zoomable treemap were recognized by all users.
- Movement between nodes in the partition visualization performed extremely well - no participants expressed any negative reaction to the movement. The sunburst also performed well, though the movement took a few seconds for users to understand and adjust.
Unsuccessful attributes. There were also many unsuccessful attributes that should be improved, redesigned, or removed in future prototypes of the tool.

- Several users did not interact with the visualization in any way until being prompted to do so.
- The sizing of nodes and groups of nodes communicates information in visualizations even when no attribute is used to explicitly communicate sizing.
- The logic behind repeated data was not clear to all users, even after seeing the sunburst visualization. Some users even saw the repetition as a problem with the tool that needs to be resolved.
- The separation between nodes was too subtle for most users, making it difficult to understand the separation between nodes and groups of nodes.
- The hovering effect received positive feedback from all users, but all users also would have preferred all the relationships, from grandchildren to ancestors, of a particular node to be highlighted.
- The buttons were underused. We postulate that this is a combination of the overall hesitation to interact and and the buttons having poor affordance to click.
- Movement between parent and child nodes was not as intuitive in the zoomable treemap as it was in the sunburst and partition visualizations.
- The breadcrumb navigation performed poorly in zoomable treemap.
- The UX health indicators were not intuitive or clear to users, and some indicators confused more than they helped.
- The lack of visual content up front was problematic for some users, and finally being able to view visual content in the collateral was a relief to these users.

Ease of understanding the data and its structure. The partition visualization performed very well in this aspect. Users perfectly understood the structure of the data, and they understood it very quickly. The sunburst visualization was also understood by most users, though not as quickly or easily. There was one notable exception for the sunburst visualization: one user was convinced that data was changing as she traversed up and down the hierarchy. The zoomable treemap also performed well here, even though users can only see one level at a time. All users, including those who saw the zoomable treemap before any other visualization, understood that they were traversing down a tree of data, and nodes were composed of the nodes beneath them in a hierarchy.

Exploring, moving, and staying oriented within a data set. All three visualizations succeeded in having users look around a set of data to understand the data. Interacting with the data, like while moving to a new level of data, had the best results with the partition visualization. All users understood intuitively where they were moving from and to while moving around the data. In fact, when the interaction and movement is clear, users do not mentioned being oriented - orientation is only something that is mentioned or noticed when it is not functioning well. The sunburst visualization fared slightly worse, with some users being disoriented by the animations, though all users seemed to understand the location and origin of any of their movements around the data. The zoomable treemap performed worst of the three. Even though users could only move up and down the hierarchy of data
one level at a time, users still found this visualization the most disorienting and confusing.

**Finding more information about an artifact.** All three visualizations performed poorly here. Most users were unable to find any more information about an artifact, other than the artifacts beneath and above it. The sunburst performed slightly better, possibly due to the placement of the collateral button, but overall it is clear that some work needs to be done here. Two users proposed filling the blank space in the zoomable treemap with content, which is an idea discussed further in 6 Conclusion and recommendations.

**Understanding relationships.** The relationships between parents and children were clear in all three visualizations. The zoomable treemap is of course the least suited to this task since only one level is visible at a time, so the real comparison is between the partition and sunburst visualizations. The relationships between artifacts two or more levels apart were immediately clear to users of the partition visualization. These relationships are not as clear in the sunburst visualization, where some users were not completely sure if outer nodes were related to their parents further inward. The hovering interaction in the sunburst visualization performed well with users, with most understanding that artifacts can be repeated in the visualization because they are related to multiple parent artifacts. This was not clear to users of the partition visualization who had not yet seen the sunburst visualization. All visualizations lacked sufficiently clear separation and visual grouping for users to be completely confident that a low-level artifact is related to a high-level artifact.

**Novel and fun visualization.** The sunburst was appreciated aesthetically by all users, both for its overall shape and structure and for the interaction. Users remained engaged with the sunburst visualization during its first task more than either of the other visualizations, mostly due to the hovering interaction. It is important to note that the perceived novelty of interaction in visualizations can quickly turn to annoyance and negative reactions after prolonged exposure [20], so it is important to test these novel and fun aspects to ensure they do not degrade or even backfire after prolonged use.

**Perception of value.** Both the partition and sunburst visualizations were perceived as providing some value in understanding the data shown in the document provided prior to the tasks. Since most of the users did not actively work with digitally stored user experience data, it is difficult to use their reactions to compare the visualization to existing methods of viewing this data. One user, who worked in the use cases and requirements space at the time of testing, saw a lot of potential in simplifying certain parts of his work and discovering relationships in the data.

**Effects of different layouts of users’ understanding of the data.** Since users tested multiple visualizations, only results from testing the first visualization saw by users can be used to determine how the different layouts affected users’ understanding of the data. Users who began with the partition and zoomable treemap layouts had a predictably linear, tree-like understanding of the data. The sunburst visualization, though also being a tree-like data structure, had more varied interpretations. While users generally understood that the visualization displayed data from the center outward, users were much
more likely to look for relationships in the data and follow non-linear paths through the data.

**Biggest roadblocks for users in understanding and using the visualizations.** The interactive parts of the visualization, including the buttons and changing focus to a new node, proved to be a major roadblock simply because users were not interacting with the visualization as intended. This hesitancy to interact must be understood and overcome in future versions of the prototype in order to have a successful visualization that encourages the exploration of data. Another roadblock was the lack of any collateral more up-front in the visualization, which seemed to be discouraging for several users, who expressed relief when finally discovering some visuals and metadata in the collateral of an artifact. The last major roadblock was the indicator layer over the partition visualization, which, while communicating some clear information for failing and passing artifacts, also confused users, who did not understand some of the indicators, nor how indicators related to the indicators of other artifacts.

6 Conclusion and recommendations

Some of the strengths and weaknesses of the visualizations tested are now known and can inform future iterations of the tool. The visualizations which are iterated further should be tested in two ways: separately and together as a package. The separate tests can help to understand a reasonable number of users’ understanding of the visualizations and the interaction. These tests can also target specific user groups. The package tests can take a different approach - a similar counterbalanced methodology as used in this research can be utilized to understand the visualizations as part of a larger system. Testing in these two ways is important because the visualizations have to work on their own to their purpose, but also together as part of a system, and it would be difficult to test both of these aspects in the same test.

We recommend five avenues of future work. The first avenue is to finish the work on the hierarchical edge bundling visualization (or an arc bundling equivalent) and the cascaded treemap (or at the very least a nested treemap), as well as to correct the clear issues raised by the usability testing performed in this research. The clearest issues were the hesitancy or unawareness of users to interact with the visualization, orientation difficulties in the treemap and sunburst visualization, lack of visuals more up-front, and the smaller issues users had with the interface elements of the visualization. More serious effort is also needed to improve the UX health indicators, as well as to expand their functionality to more layers to show other attributes of the data (assuming the visualizations are connected to more complete, robust data sets). The concept of UX health should also be revisited in relation to the UX landing zones. We propose that the UX health concept is tied directly to the UX landing zones. The resulting health indicator for artifacts is either defined by tests performed against the UX landing zones directly, or as an aggregate of the UX health of the artifact’s child nodes.

The second avenue is to get the visualizations communicating with some databases of product development data, preferably live ones, or at the very least an active sandbox. This will help with developing the visualization to ensure they work well for different data sets, not just a single data set. The
visualizations can easily be extended, even from their current state, to handle many more levels of data, so even a complex database with many levels in the data hierarchy is possible to visualize.

The third avenue is to build a creative component to the tool, where users can create their own data set collaboratively with other users. This is the most interesting direction to me because the problem of getting users to create product development data in a new tool is the most difficult hurdle to get to the desired end state of structured data throughout product development. We built some early versions of how this could function - these are shown in Figure 15 below.

![Figure 15: Several visualizations of the same randomly generated data set. Nodes can be added as children to any element in any of the visualizations. Some work still needs to be done before testing with users, such as building the interface for the input of more data than just node labels.](image)

The fourth avenue is to run a new usability test, but this time with a much more specific target audience of strictly strategic planners, UX designers, and requirements engineers. This avenue requires the third avenue to be completed as well, and preferably the first and second avenues as well. This testing would again test these users’ understanding of the visualization, as well users’ perception of value, engagement, and a direct comparison to their existing work processes. It may even be necessary to do a formal contextual inquiry to discover more about existing work habits and where a tool like this may limit the users. Eventually, running a trial of the tool on a full product strategy and UX team should be done to stress-test the concept.

Follow-up research could be performed on certain usability concerns discovered in the user testing. Fully
understanding why users are hesitant or unaware of the interaction possibilities with interactive visualizations would be a worthwhile piece of research. Strategies could be recommended for developing successful interactive visualizations. Heuristics could be developed for expert evaluations of interactive visualizations.

In summary, some promising positive results were found showing that interactive data visualization could be used for analyzing and understanding product development data. But a significant amount of work is still left to iterate the tool assess the utility of different visualizations, as well as the changes to users' work processes that would result from changing tools.

Acknowledgements

I first want to thank my Intel supervisor, Delia Grenville, and my Uppsala University supervisor, Else Nygren. Delia provided me with a fascinating business problem to solve, and Else allowed me to take this problem and use it as part of my thesis project. This work would not have been possible without their assistance and sponsorship. I also owe a huge thank you to Mike Bostock for his open source D3.js library and extensive set of examples that made this work possible in the given time frame. I would also like to thank my manager, Ralph Brooks, a manager in the Project Controls and Solutions Office of Intel, for enabling this work.

References


Appendices

Appendix A: Introduction
“The interface you are about to see is a concept of a tool that could be used to visualize user experience data that has been collected for a product in development. The data you will see here is from an old [company] project that was developed in 2007.”

Appendix B: UX Landing Zone task

<table>
<thead>
<tr>
<th>Value Proposition</th>
<th>Usage category</th>
<th>Usage</th>
<th>Usage</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple and cool</td>
<td>Setup device</td>
<td>Setup content services</td>
<td>Migrate existing accounts to device</td>
<td>Use case</td>
</tr>
</tbody>
</table>

Appendix C: Artifact definitions for test users

A value proposition is an overarching statement that represents what users will value. It is written from the end-consumer perspective, using a first-person voice.

A usage category is just like it sounds: a category of usages. It is simply a way to generalize about a group of usages and to provide a bridge to their associated value proposition(s).
A *usage* is a small statement that further defines the value propositions in a usage category. It provides a common ancestor for the data that follows, including usage requirements, usage scenarios, and use cases.

A *use case* defines interactions between an actor and a system to achieve a goal.