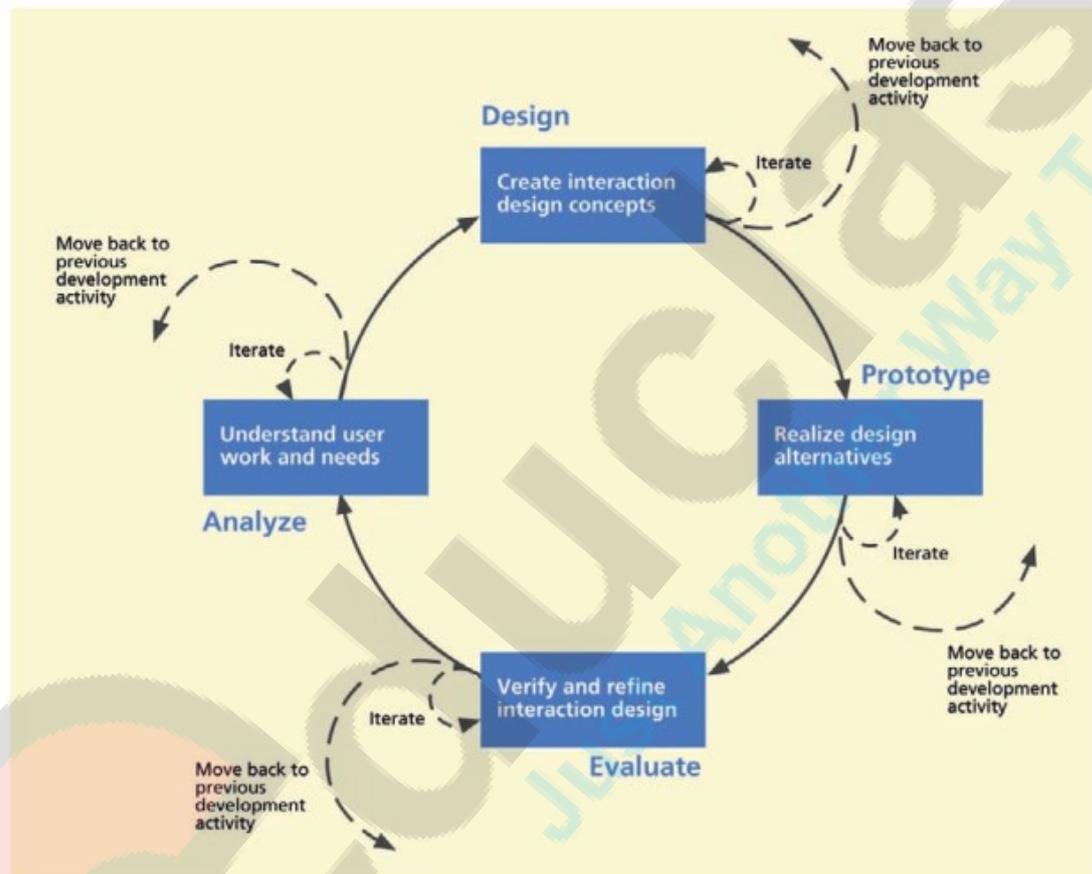


## 1. Draw and explain UX lifecycle template?

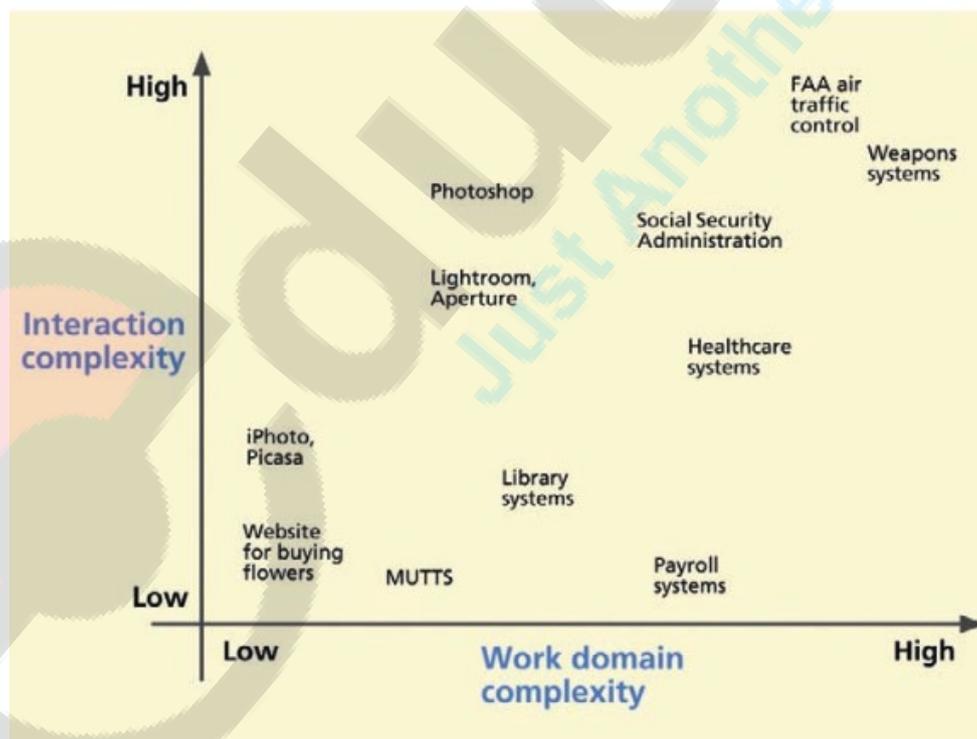
- In Figure 1 we depict a basic abstract picture of activities for almost any kind of design, a cycle of the four elemental UX activities—Analyze, Design, Implement, and Evaluate—that we refer to generically as analysis, design, implementation, and evaluation. These four activities apply whether you are working with an architectural design, a hardware design, or a new car concept.
- In our lifecycle concept, specific to a UX process, **analysis** translates to understanding user work and needs. **Design** translates to creating conceptual design and determining interaction behavior and look and feel. **Implementation** translates to prototyping, and **evaluation** translates to ways to see if our design is on track to meet user needs and requirements.



- In the context of interaction design and UX, this abstract cycle translates to our UX lifecycle template of Figure 2, which we call the Wheel.
- In a larger system view, implementation includes a final production of hardware and software, including the user interface. However, in our UX lifecycle template, implementation is limited to the interaction design component and prototyping is the design manifestation we use for evaluation before it is finalized for production.
- The evaluation activity shown in Figure 2-2 includes both rigorous and rapid evaluation methods for refining interaction designs.
- Beyond that evaluation activity, the entire lifecycle is evaluation centered in the sense that the results of potentially every activity in the lifecycle are evaluated in some way, by testing, inspecting, analyzing, and taking it back to the customers and users.
- The entire lifecycle, especially the prototyping and evaluation activities, is supplemented and guided by UX goals, metrics, and targets.

## 2. System Complexity Space?

- One of the things that makes it difficult to define a process for system design is that there is a spectrum of types of systems or products to be developed, distinguished mainly by complexity, each needing a somewhat different process and approach.
- Some systems are a combination of types and some are borderline cases. System or product types overlap and have fuzzy boundaries within the system complexity space. While there undoubtedly are other different ways to partition the space, this approach serves our purpose.
- In Figure we show such a “system complexity space” defined by the dimensions of interaction complexity and domain complexity. Interaction complexity, represented on the vertical axis, is about the intricacy or elaborateness of user actions, including cognitive density, necessary to accomplish tasks with the system.
- Low interaction complexity usually corresponds to smaller tasks that are generally easy to do on the system, such as ordering flowers from a Website. High interaction complexity is usually associated with larger and more difficult tasks, often requiring special skills or training, such as manipulating a color image with Adobe Photoshop.
- On the horizontal axis in Figure we show work domain complexity, which is about the degree of intricacy and the technical nature of the corresponding field of work. Convoluted and elaborate mechanisms for how parts of the system work and communicate within the ecology of the system contribute to domain complexity.
- The work in domain-complex systems is often mediated and collaborative, with numerous “hand-offs” in a complicated workflow containing multiple dependencies and communication channels, along with compliance rules, regulations, and exceptions in the way work cases are handled.



### 3. The System concept Statement

- A system concept statement is a concise descriptive summary of the envisioned system or product stating an initial system vision or mandate; in short, it is a mission statement for the project.
- A system (or product) concept statement is where it all starts, even before contextual inquiry. We include it in this chapter because it describes an initial system vision or mandate that will drive and guide contextual inquiry.
- Before a UX team can conduct contextual inquiry, which will lead to requirements and design for the envisioned system, there has to be a system concept. Rarely does a project team conceptualize a new system, except possibly in a “skunk-works” kind of project or within a small invention-oriented organization.
- The system concept is usually well established before it gets to the user experience people or the software engineering people, usually by upper management and/or marketing people.
- A clear statement of this concept is important because it acts as a baseline for reality checks and product scope and as something to point to in the middle of later heated design discussions.
- Writing a good system concept statement is not easy.
- The amount of attention given per word is high. A system concept statement is not just written; it is iterated and refined to make it as clear and specific as possible.
- An effective system concept statement answers at least the following questions:
  - What is the system name?
  - Who are the system users?
  - What will the system do?
  - What problem(s) will the system solve? (You need to be broad here to include business objectives.)
  - What is the design vision and what are the emotional impact goals? In other words, what experience will the system provide to the user? This factor is especially important if the system is a commercial product.
- The audience for the system concept statement is broader than that of most other deliverables in our process and includes high-level management, marketing, the board of directors, stockholders, and even the general public.

### 4. Data Driven v/s Model Driven Enquiry

- Simply stated, letting data do the driving means that if you encounter any information that seems relevant to the work practice and its milieu, collect it. This approach means forestalling any influence from your own knowledge, experience, or expectations and just gathering data as they present themselves.
- Data-driven contextual inquiry results in voluminous raw data describing a wide variety of topics. To digest this mass of disparate data points, make sense of them, and put these data to work in informing design, practitioners must apply contextual analysis to extract the concise and meaningful points and issues and then sort and organize them into piles or affinity diagrams.
- Then the sorted categories must be converted into design-informing models such as flow models, user models, and task models. In the purely data-driven approach, these categories and models are dictated by the data content.
- However, there is more than one effective approach to gathering contextual data to inform design. They promote a method they call model driven, which is in important ways the reverse of the data-driven approach.

- In their “use what you know”(i.e. Model Driven) approach, using knowledge and expectations from experience, intelligent conjecture, knowledge of similar systems and situations, marketing analysis, mission statements, and preliminary requirements to focus your contextual inquiry data gathering to anticipate preconceived data categories and target the most useful data and to get a head start on its organization and analysis.
- From this experience, most practitioners know what kinds of models they will be making and what kinds of data feed each of these models. This knowledge helps in two ways: it guides data collection to help ensure that you get the kinds of contextual data you need, but at the risk of analyst bias in those data. It also helps with analysis by giving you a head start on data categories and models.
- Certainly not all of this anticipatory information will be correct for a given work practice or system, but it can provide an advantageous starting point. Although their process might seem that it is about modeling and then finding just the data to support the predefined models, it really is about starting with some initial “exploratory” models to guide data collection and then focused data collection to find answers to questions and outstanding issues, to refine, redirect, and complete the models.
- This “model-driven inquiry” approach also has a solid real-world track record of effectiveness.
- The data driven approach works because in the end, data will determine the truth about work practice in any specific real-world customer or user organization.
- However the model driven approach works because it encourages you to use your experience and what you know to anticipate data needs in contextual inquiry and contextual analysis. While data-driven inquiry assumes a “blank slate” and a completely open mind, model-driven inquiry acknowledges the reality that there is no such thing as a blank slate.
- Data driven approach is rooted in real data untainted by guesswork or analyst biases. But the Model Driven approach claims advantages in lower cost and higher efficiency; they claim the search for data is easier if you know something about what you are looking for.

## 5.Contextual Analysis

Contextual Analysis Is Data Interpretation Now that you have used contextual inquiry to observe and interview users about the nature of their work in context and collected corresponding contextual data, it is now time to analyze that data to understand the work domain.

Contextual analysis consists of user work activity data interpretation, consolidation, and communication. Interpretation of raw work activity data is accomplished through:

- building a flow model and
- synthesizing work activity notes

Data consolidation and communication are accomplished by, respectively:

- building a work activity affinity diagram (WAAD) from the work activity notes
- walkthroughs of all these work products

The main point of contextual analysis has two basic parts:

- Converting raw contextual data into work activity notes
- Converting work activity notes into a work activity affinity diagram(WAAD)

**Explain how to synthesize and manage work activity notes from raw contextual data?**

- **Converting raw contextual data into work activity notes**
- Transcribing Interview and Observation Recordings
- If you recorded your user and customer interviews in the contextual inquiry process, video and/or audio, you must begin contextual analysis with transcription, so you can see the raw observation and interview notes.
- The written notes or transcripts will of course still be just as raw as the recordings, meaning you still have to do the analysis to filter out noise and boil it down to the essentials.
- **Reviewing Raw User Work Activity Data**
- Recounting one interview at a time, researchers:

- review interview and observation notes and any recorded audio-
- retell the events
- in discussion with the group, capture key points and issues, design ideas, missing data, and questions arising in the course of the discussion
- A moderator in each group keeps things on track, while user researchers give accounts of each interview.
- In general, people may interrupt to ask questions to clarify issues or fill gaps.
- After the group data interpretation sessions, the groups get back together for brainstorming to tie up loose ends on the data interpretation.
- Work activity notes are shared, discussed, and adjusted as needed and new ones that come from this discussion are added.
- Finally the group engages in introspection about lessons learned. The group brainstorms to evaluate their process reflecting on what went well and what could be improved for next visit and how. The outputs of this process of review and interpretation are:
  - sets of work activity notes synthesized from raw data
  - a work activity affinity diagram to organize the work activity notes
- **Synthesizing Work Activity Notes**
- The work activity note synthesis should be done by people who have already been immersed in the contextual data, probably the same people who did the interviews and observations.
- The notes should be captured in some kind of computer-readable form, whether it is in a word processor, a spreadsheet, or directly into a database system.
- Ideally, the final set of synthesized work activity notes should represent raw data so well that the team never has to go back to the raw data to answer questions, fill in blanks, determine what the real point was, or to sort out context.
- GuideLines
  - As you create each new work activity note, tag it with a source ID, a unique identifier of the person being observed and/or interviewed when the note was written.
    - These tags are essential links to follow back to the source person in case further questions must be asked about missing data, unanswered questions, etc.
  - Paraphrase and synthesize instead of quoting raw data text verbatim.
    - It is perfectly acceptable and often advised to paraphrase and rephrase or to condense or summarize to make your own synthesized user “statements.”
    - We want the user’s perspective but not necessarily verbatim quotes of the user’s words, which can be verbose and indirect. For paraphrased statements, you should maintain the user’s perspective and remain true to the user’s intentions.
    - You should not introduce any new content and keep the expression terse and to the point. It is the analyst’s responsibility to abstract out a clear and concise statement conveying the substance of the issue in question.
  - Make each work activity note a simple declarative point instead of quoting an interviewer’s question plus the user’s answer.
    - Questions coming from the interviewer and confirmed by the user should be worded as if they came from the user.
  - Filter out all noise and fluff; make each note compact and concise, easily read and understood at a glance.
    - Raw user data are usually too verbose. You must filter out the noise and irrelevant verbiage, boiling it down to the essence.
  - Each note should contain just one concept, idea, or fact, with possibly one rationale statement for it. Break a long work activity note into shorter work activity notes.
  - Make each note complete and self-standing.
  - Never use an indefinite pronoun, such as “this,” “it,” “they,” or “them” unless its referent has already been identified in the same note.
  - State the work role that a person represents rather than using “he” or “she.”

- Add words to disambiguate and explain references to pronouns or other context dependencies.
- Avoid repetition of the same information in multiple places.
- **Extending the Anticipated Data Bins to Accommodate Your Work Activity Note Categories**
  - Extend the set of existing bins to cover all the anticipated data categories for your work activity notes, as you synthesize them from the raw data.
  - Keep the bins as labeled stacks of notes on your work table so that the whole team can see them. The labels will denote all the useful categories plus a few generic ones for “open questions to pursue” and “issues for further discussion or debate.”
  - Examples of typical data categories you might encounter in your raw data are:
    - User and user class information
    - Social aspects of work practice (how people interact with and influence each other)
    - Emotional impact and long-term phenomenological aspects
    - Task-specific information

### Printing Work Activity Notes

- Notes printed or handwritten on colored bond printer paper formatted, say, six to a page, also work fine. If your work activity notes come from a database, you can use the “mail-merge” feature of your word processor to format each note into the table cells for either plain paper or Post-it printing.
- Whichever stock you choose, print your work activity notes on plain white or yellow paper or Post-it stock to distinguish from other colors that you might use later for labels in the WAAD.

### Converting work activity notes into a work activity affinity diagram(WAAD)

It is same as WAAD refer that answer

### 6.Extracting the interaction design requirement

Contextual inquiry and analysis are about understanding existing work practice and context. Then we move on to producing designs for a new system to support possibly new ways that work gets done. But what happens in between? The output of contextual inquiry and analysis does not speak directly to what is needed as inputs to design. There is a gap. Information coming from contextual studies describes the work domain but does not directly meet the information needs in design. There is a cognitive shift between analysis-oriented thinking on one side of the gap and design-oriented thinking on the other.

#### NEEDS

- In the UX domain, interaction design requirements describe what is required to support user or customer work activity needs. To that end we are also concerned with functional requirements to ensure the usefulness component of the user experience. Finally, we will have requirements to fulfill the need for emotional impact and long-term phenomenological aspects of the user experience.
- Before we get into extracting requirements from contextual data, let us look briefly at the forms interaction design requirements can take. One term we often think of when coupled with “requirements” is “specifications.” In past software engineering traditions, a formal written requirements and even designate details about how the corresponding software is to be implemented, including such software stuff as object models, pseudo-code, use cases, and software structure.
- However, on the UX side we are talking about only interaction design requirements, nothing about software or implementation. Also, as we will see, it is not easy to lay down that same

kind of requirements specification for the interaction design, nor is it particularly useful to try. However we specify our requirements, there is a broad range of acceptability for completeness and detail. For domain-complex systems, with many requirements for compliance and risk avoidance, you may need a rather complete specification of requirements.

- Therefore, at the end of the day, or more likely the end of the week, requirements extraction produces an assortment of deliverables, each of which can be thought of as a kind of “specification”—for needs and requirements and for design-informing models such as personas, tasks, user experience goals, or usage scenarios.
- User needs are really not just interaction needs. Usability and UX include usefulness that we get from functionality.
- Often an initial requirement extracted from contextual data first appears as a requirement for a broad overall system capability—that is, it expresses a need for both functionality and user interface support.

### **FORMAL REQUIREMENTS EXTRACTION**

This process of extracting needs and requirements is similar to data interpretation and consolidation sessions of contextual analysis in that it involves a group sitting down together and going over a large amount of data. But here it is actually easier because much of the hard work is already done.

- Switching from Inductive to Deductive Reasoning
  - Extracting requirements from the WAAD calls for a deductive thinking process. It is deductive because each work activity note in the WAAD is treated as the major role.
- Preparation
  - Select a requirements team, including people you think will be best at deductive reasoning and creativity. You will need both UX and software people represented, plus possibly system architects and maybe managers.
  - You may need a requirements “record” template in a word processing document, a spreadsheet, or a database schema to capture the requirement statements in a consistent and structured format in an interaction design requirements document (or requirements document, for short in this context).
  - The requirements team will work in the room where the WAAD is posted on the wall.
  - If there is a need for all to see each requirement statement, you can connect the recorder’s computer to a screen projector and show the requirements document on an open part of the wall.
  - The leader is responsible for walking the team through the WAAD, traversing its hierarchical structure systematically and keeping the team on track.
- Deduction of Needs to Get at Requirements
  - As the leader walks the team through the WAAD, one node and one note at a time, the team works together to ask what user needs, if any, are reflected in this work activity note and the hierarchical labels above it.
  - Such user needs are still expressed in the perspective of the user and in the work domain.
- Terminology Consistency
  - Your contextual data will be full of user comments about an infinitude of usage and design concepts and issues.
  - For example, users of a calendar system might use the terms “alarm,” “reminder,” “alert,” and “notification” for essentially the same idea.
- Requirement Statements
  - Each requirement statement describes a way that you decide to support the user need by providing for it in the interaction design.
  - Now the recorder writes the requirement statement in the requirements document by first finding the appropriate headings and subheadings. If the necessary headings are not already in the requirements document, now is the time to add them and grow the document structure as the process continues.

- Interaction requirements often imply functional requirements for the system, which you may also capture here for communicating to your software people.
- For example: Interaction requirement: “Ticket buyers shall be able to see a real-time preview of available seating for a venue.”
- Requirement Statement Structure
  - A requirements document is essentially a set of requirement statements organized on headings at two or more levels.
  - For systems where risk is high and traceability is important, each requirement is tagged with the WAAD source node ID, which serves as a link back to the source of this requirement statement within the WAAD. The WAAD in turn has a link back to its source in raw work activity data.

### **Security**

#### *Privacy of ticket-buyer transactions*

Shall protect security and privacy of ticket-buyer transactions [C19]

Note: In design, consider timeout feature to clear screen between customers.

## 7. Constructing design information models

### 9. WORK ACTIVITY AFFINITY DIAGRAM (WAAD)

- **Introduction**
  - Work Activity Note: A work activity note is used to document a single point about a single concept, topic, or issue as synthesized from the raw contextual data. Work activity notes are stated as simple and succinct declarative points in the user’s perspective.
  - Affinity diagramming is a technique for organizing and grouping the issues and insights across all users in your contextual data and showing it in a visual display that can cover one or more walls of a room. By pulling together work activity notes with similarities and common themes, a work activity affinity diagram, guided by the emerging flow model, helps consolidate contextual data and generalizes from instances of individual user activities and issues to highlight common work patterns and shared strategies across all users.
- **What You Need to Get Started**
  - You are going to build a hierarchical diagram of common issues and themes taken from the data. An affinity diagram is used to organize an enormous mound of individual work activity notes into a structure that yields sense, affords visualization of the user’s workand, eventually, suggests ideas for designs to support it.
  - Make sure you have in hand the huge stack of work activity notes. Line up the players, the WAAD team. You will need about two people per 100–150 work activity notes, as the goal will be to complete the WAAD in a short time (1 to 1 1/2 days).
  - Establish roles and responsibilities. Appoint one of the original interviewers or note takers as leader or moderator to manage the process and the people, and to keep the WAAD building on track. The larger the group, the more leadership and moderation needed.
- **Set Rules of the Game**

- The moderator explains how it works. Shuffle the work activity notes so that each player gets a variety and no person in your group gets the notes from just one interviewee. Deal out a limited number of notes to each team member.
- Because you are just getting started, you will not know how the structure will turn out.
- Therefore, it is best to start as low on the butcher paper as possible, building upward and leaving room for more and more levels.
- **Growing Clusters**
  - After notes begin to be posted:
  - Each team member in turn looks through his or her pile of notes, looking for other notes that are topically similar, for example, about the same user concern or work activity, to ones that have been posted.
  - Notes that seem similar are said to “have an affinity for each other” and are read aloud and posted together in a cluster or “cloud” on the wall.
  - Neatness is not essential at this point; just get birds of a feather to flock together.
  - If there are two or more of essentially the same note, derived from different users, include them all in the cluster to show the “weight” of that issue.
  - When no more notes can be found immediately to match the affinity of an existing cluster, someone will pick a new note from their hand to start a new cluster, and so on.
- **Topical Labels for Clusters**
  - As the number of clusters grows, we have found it difficult to remember by what criterion each cluster was formed (the topic of their affinity). Work activity notes are put into the same cluster because they have an “affinity” for each other; that is, they share some common characteristic.
  - But a quick glance at a cluster does not always reveal that characteristic. As a solution, make a temporary label to make the “topic” of each cluster explicit, to identify the “gestalt” of the whole group, the theme that brought them together.
  - Temporary cluster labels allow analysts to consider the cluster as a candidate for posting further notes without having to look through the notes themselves every time.
- **Speeding It Up**
  - Later on, when everyone is up to speed, all the players can come up to the WAAD and move things along by “playing” their work activity notes in parallel. Each one walks up to the growing WAAD and posts his or her work activity notes where appropriate, while trying to stay out of each other’s way.
- **Stay Loose**
  - If needed, for example, to split a work activity note into two, handwritten notes can be added, but do not make up your own data at this point. If you think a note should or could be in more than one place in the WAAD, break it into more than one note by making copies of the note and indicating that other copies of the note exist
- **Do Not Get Invested in Data Ownership**
  - No work activity note or group is “owned” by any team member; you just have to go with the flow and see how it develops. The success of the WAAD-building process is determined somewhat by the competence and experience of the analysts at organizing and classifying information, identifying common characteristics, and naming categories.
- **Labeling Groups**
  - The team invents a label for the group derived from the notes, representing the theme of the group, and often adapted from the cluster topical label. The label for each group is handwritten on a Post-it of the color chosen for group labels and posted at the top of the column of notes in the group.
- **Grouping Groups**

- After all the groups have been labeled, build up the hierarchy to reduce the structure breadth and increase the depth by grouping the groups. Looking at the group labels, move them around into larger groupings.
- When you get groups of, again, up to about a half dozen group labels, they are supergroups or second-level (going up from the group labels at bottom) groups, which are labeled in a different color, the second-level color.
- **Walkthrough of the WAAD: Consolidation and Communication**
  - One of the purposes of doing a walkthrough of the WAAD is communication, to share an appreciation of user work activities and associated issues with all stakeholders. At the same time, you can review and unify your work activity notes within the structure of the WAAD and look for data holes, work activity notes you still need.
  - Invite all stakeholders, including marketing, customers, potential users, engineering and development staff, and so on. Decide on a strategy for sharing and communicating the contextual inquiry and analysis results.
  - Tell everyone upfront (before the meeting) how it will work, who is involved, what is needed, how long to plan for, etc. Explain your process in a nutshell.
  - For management, emphasize high-level issues, cost justification, data integrity, security, and such corporate goals.
  - Highlight the most important points and issues discovered.
  - Create interest with unexpected things learned.
  - Show graphical representations; flow models can be the most effective, as they show your interpretation of the flow of information and materials within their business process.
  - Get management engaged to show them the effectiveness of your process.
  - Get developers engaged to obtain buy-in for the upcoming requirements and design activities.
  - Try to fit your process into the established methodologies of your organization; keep discussion user centered or usage centered with a user perspective, and real user quotes.
  - design ideas—capture them while you can by adding them as “design idea” notes, distinguishable from the work activity notes by using a different color and/or by adding them at a different angle on the wall
  - questions—to be answered by the team or by further data collection; add as “question” note in a different color or orientation
  - data “holes”—missing data that you have discovered as necessary to complete the picture, used to drive further data collection in the field and added as “hole” notes in a different color or orientation.