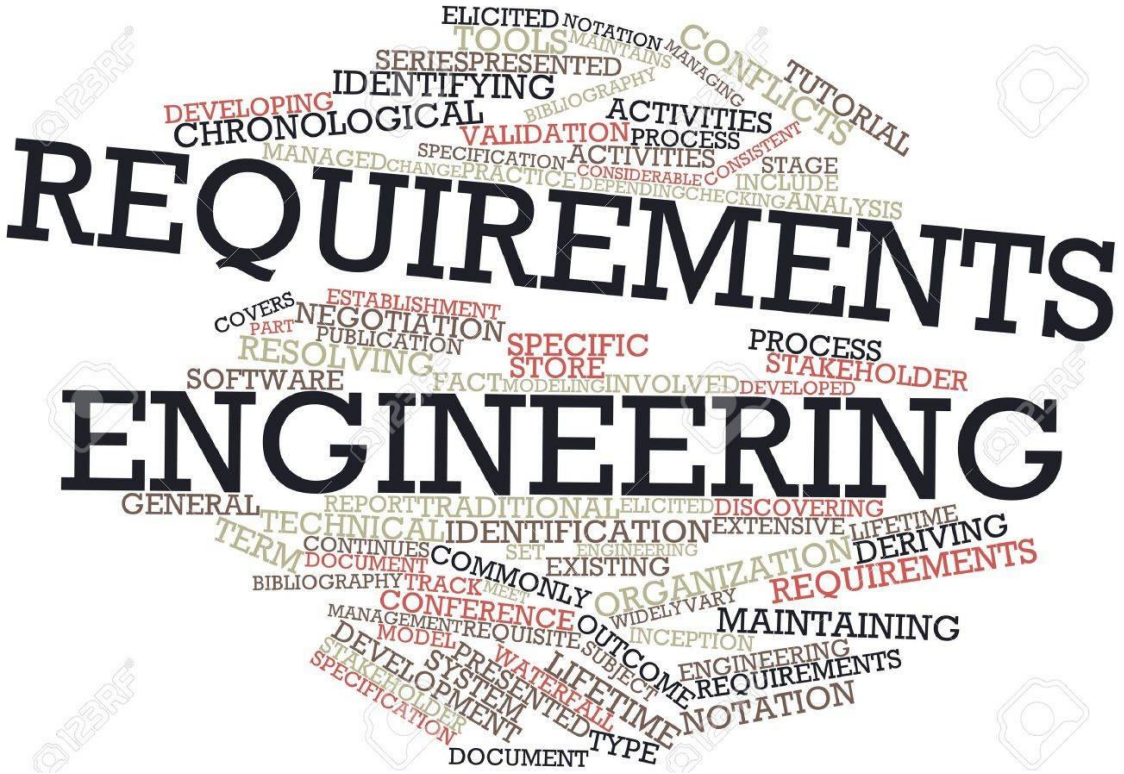


SWE 215: Software Requirements Engineering

Lecture 7

Requirements Discovery

So, ummm, how many undiscovered ruins are there?

—Anonymous tourist, winner of the "stupidest question asked by a tourist award," Mesa Verde, Colorado tour guides

Course Topics

- ~~Why Requirements Engineering?~~
- ~~Introduction to Requirements~~
- ~~RE in Software Development Life Cycles~~
- ~~System Vision, Context, and RE Framework~~
- ~~Fundamentals of Goal Orientation~~
- ~~Fundamentals of Scenarios~~
- Requirements Discovery
- User Stories and Agile Estimation
- Features Prioritization
- Requirements Negotiation
- Requirements Validation
- Fundamentals of Requirements Management

Lecture Objectives

Learn many requirements discovery methods:

- The Requirements workshop
- Brainstorming
- Questionnaires
- Interviews
- User Experience Mock-ups
- Ethnography
- Competitive analysis
- Customer change request systems
- Use Case Modeling

Requirements Discovery Challenge

Challenge:

How teams should go about understanding on a more systematic basis:

- What problems their solution is intended to address?
- What markets or types of customers it is intended to serve?
- What are the functional and non-functional requirements?

Represents one of the most critical competences required in the industry.

Variety of circumstances, markets, consumers, uses, products, systems, services, and so on,

→ need a **variety of techniques to discover requirements**

The Requirements Workshop

- **Goal:** Drive **consensus** on the requirements of the system or application and to gain rapid **agreement** on a course of action from the key stakeholders, in a **very short time**.

- Key stakeholders of the project are gathered together for a short, intensive period, typically no more than a day or two.

- The workshop may be facilitated by a **product owner**, **product manager**, **team member**, or **outside facilitator**.

The Requirements Workshop

A properly run requirements workshop has many benefits:

1. It forges an agreement between the stakeholders, the product owners, and the development team as to what the application must do.

2. All stakeholders get their say; no one is left out.

3. It can expose and resolve political issues that may otherwise interfere with project success.

Running the workshop

- These decision-making workshops are often characterized by **a highly charged atmosphere.**

- **Difficult to get consensus**, including different **opinions** and **expectations** for the solution requirements and the **impact on resources and budgets.**

- Let the facilitator take the heat and manage the meeting

- The most important part of the workshop is the **brainstorming process:**
  - Ideally suited for the workshop setting.
  - Fosters a creative and positive atmosphere and gets input from all stakeholders.

Brainstorming

Brainstorming is a **simple**, **fun**, and easy way to get stakeholders to **contribute**. **Brainstorming** does the following:

- Encourages participation by all parties present
- Allows participants to "piggyback" on one another's ideas
- Has high bandwidth—many ideas can be generated in a short period
- Identifies multiple potential solutions to whatever problem is posed
- Encourages out-of-the-box thinking—unlimited by the usual constraints

Brainstorming rules

Rule 1: Quantity over quality.
Rule 2: Free association and visionary thinking are explicitly desired.
Rule 3: Combining contents
Rule 4: Criticism is forbidden
Rule 5: Questions for clarification are allowed
Rule 6: Do not abort the brainstorming at the first deadlock
Rule 7: The brainstorming shall come to a natural end.

(Klaus Phol "Requirements engineering" book)

Brainstorming main phases

Brainstorming has two main phases:

1. Idea generation

2. Idea reduction

Optional step: Idea Prioritization

Brainstorming: Idea Generation

The first objective is the generation of as many ideas as possible in a short time frame.

When a person comes up with an idea, he or she also writes it down in order to assure the following:

- Ideas are not lost
- Ideas are captured in that person's own words
- Ideas can be posted for later discussions

Brainstorming: Idea Reduction (Pruning)

- The first step is to "prune" those ideas that are **not worthy** of further investment by the group.
- The presence of ideas that can be **easily pruned** is an indicator of a **quality process**.
- The absence of a fair number of **wild and crazy ideas** indicates that the participants were not thinking far enough "**out of the box**".
- The facilitator asks the participants whether each idea is worth of further consideration and then **removes an invalid idea, but if there is any disagreement among the participants, the idea stays on the list.**
- If participants find two sheets with the **same idea**, **group them together**.

Brainstorming: Idea Prioritization

- In some situations, **the generation of ideas is the only goal**, and the process is then complete.
- However, it is often useful to **prioritize** the remaining ideas.
- After all, no development team can do "everything that anybody can think of," and the key stakeholders are still present.
- A variety of techniques can be used for prioritization.

Examples:

- Cumulative Voting: The $100 Test
- "Critical, Important, Useful" Categorization

Brainstorming: Idea Prioritization: Cumulative Voting: The $100 Test

- Each person is given **$100** of "**virtual idea money**" to be spent on "**purchasing ideas**".
- Each participant decides how much money to spend on each idea. Then votes are tabulated, and the results are rank ordered.
- A quick histogram of the result can help participants see the visual impact of their decisions.

Brainstorming: Idea Prioritization: Cumulative Voting: The $100 Test

This process is straightforward and usually works quite well.

Caveats:

- It may be necessary to **limit the amount** anyone spends on one feature. Otherwise, a participant, might put all of their money on a single feature in order to elevate it to a higher priority.

- It may work only once, because once the results are known, participants may bias their input for the second vote.

Brainstorming: Idea Prioritization: "Critical, Important, Useful" Categorization

- Each participant is given a number of votes equal to the number of ideas, but each vote must be categorized "critical", "important", or "useful". Each stakeholder is given **only one-third of the votes** from each category.

1. Critical features are mandatory: a system deployed without this feature could not fulfill its **primary mission or meet the market need**.
2. Important means that failure to include the feature could cause a **loss of customer utility, market share, or revenue.**
3. Useful means nice to have. Useful features make the system more appealing to use or deliver higher utility to some class of users.

Brainstorming: Idea Prioritization: "Critical, Important, Useful" Categorization

- After voting, each feature will probably have a mix of categories
- Multiply "**critical**" votes times **9**, "**important**" by **3**, and "**useful**" by **1**; then **add up the score**.
- This tends to spread the results to **heavily favor** the "**critical**" votes, and thus every stakeholder's "critical" need will tend to bubble toward the top.

Brainstorming: Online Brainstorming

Suited for developing **advanced applications** for which **research is required**, the concept is initially fuzzy, and **a wide variety and significant number of users and other stakeholders inputs are involved.**

Advantage: Ideas and comments can be circulated **over a long period of time**, with full recording. Ideas can grow and mature with the passage of time.
Disadvantage: There is no time to contemplate the issues, and the team **needs resolution now**. In such cases, distributive, immediate collaboration is still possible through any number of online meeting/collaboration tools.

Summary of Requirements Workshops and Brainstorming

The goal of these techniques is to maximize the contribution of each team member in harmony with the objectives of the project and its mission.

Interviews and Questionnaires

- A **simple** and **direct** technique that can be used in virtually every situation.
- **As solution providers, we rarely find ourselves in a situation in which we have no idea what types of potential solutions would address the problem.**
- However, we must not let our current context interfere with a better understanding of a new problem to be solved.

Context Free Questions

- Asking questions about the **nature of the user's problem without context for a potential solution.**

Examples of context-free question:

- Who is the user?
- Who is the customer?
- Are different user needs different?
- What other stakeholders will be impacted by this product or project?

Successful Interview

- **Understand the background of the stakeholder** to be interviewed.
- Make sure that **the script is not overly constraining**.
- The customer may well launch into a stream-of-consciousness dialogue, **describing in detail the horrors of the current situation**. *Do not cut it off prematurely with another question; rather, write down everything as quickly as you can, letting the user exhaust that particular stream of thought.*
- **Ask follow-up questions** about the information that has just been provided.

The Analyst's Summary

- Used for recording **the most important needs** or **problems uncovered** in the interview.

- In many cases, after just a few interviews, **these highest-priority needs will start to be repeated**. This means you may be starting to get convergence on some common needs.

Questionnaires

- It consists of a **series of questions** for the purpose of **gathering information from respondents**.

- Used when it is impossible because of **time**, **distance** or **cost constraint**, to **interview all the desired people** involved in a system.

- It allows the analyst to **collect facts from a large number of people**

Good Questionnaires Design

- Begin with non-threatening and interesting questions
- Group items **into logically coherent sections**
- Do not put important items at the very end of the questionnaire
- Do not crowd a page with too many items
- Avoid abbreviations
- Avoid biased or suggestive items or terms
- Number questions to avoid confusion
- Provide secrecy to respondents
- Test the questions on a small sample of respondents.

Types of Questionnaires

Free format questionnaires: the respondent records the answer in the space provided after the question.

- Examples:
  - What reports do you currently receive and how are they used ?
  - Are there any problem with these reports ? If so please explain.

Fixed-format questionnaires contain questions that require selection of predefined responses from individuals:

- Multiple choice questions
- Rating questions
- Ranking questions

Types of Fixed-Format Questions

Multiple choice questions:

Examples:

Do you agree with the collected requirements? Yes ☐ No ☐

Is the current report useful? Yes ☐ No ☐

Types of Fixed-Format Questions

Rating questions:

Example:

The implementation of discounts would increase customer orders:

- ☐ Strongly agree
- ☐ Agree
- ☐ Neutral
- ☐ Disagree
- ☐ Strongly disagree

Types of Fixed-Format Questions

Ranking questions:

Example:

Rank the following transactions according to the amount of time you spend processing them:

_____% New customer orders

_____% Order cancellations

_____% Order modifications

_____% Payments

A Note on Questionnaires

- Although questionnaires are often used and appear scientific because of the opportunity for statistical analysis, a questionnaire is **not a substitute for interviewing** in the requirements discovery context because it has some fundamental problems.
- **Relevant questions** cannot all be decided in advance.
- The assumptions behind the questions bias the answers.
- It is difficult to explore new domains ("What you really should be asking about is ...").
- It is difficult to follow up on ambiguous user responses.
- There is no substitute for the personal contact, rapport building, and free-form interaction of the interview.
- Do the interview first and do it for every new class of problem.

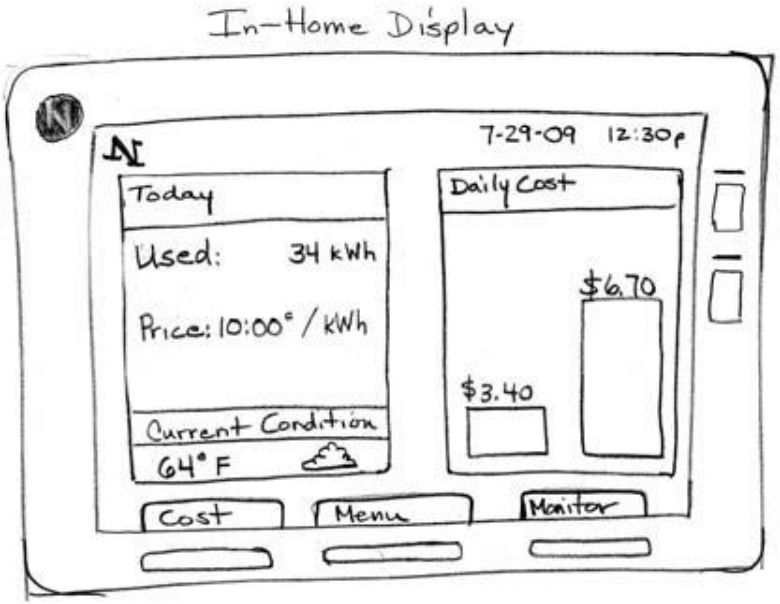
User Experience Mock-Ups

- Developing **user experience interfaces** is always a challenge.
- All aspects of the **presentations**, **dialogues**, and **flows** that make a user interface **easy to navigate** are likely to be largely **unexplored** prior to beginning implementation → **results in real frustration**.
- **"A picture is worth a thousand words"**
- An easy-to-change **drawing** or **mock-up** is an early and effective means of communicating elements of user presentation.
- The **lighter** the presentation, **the quicker the feedback**.

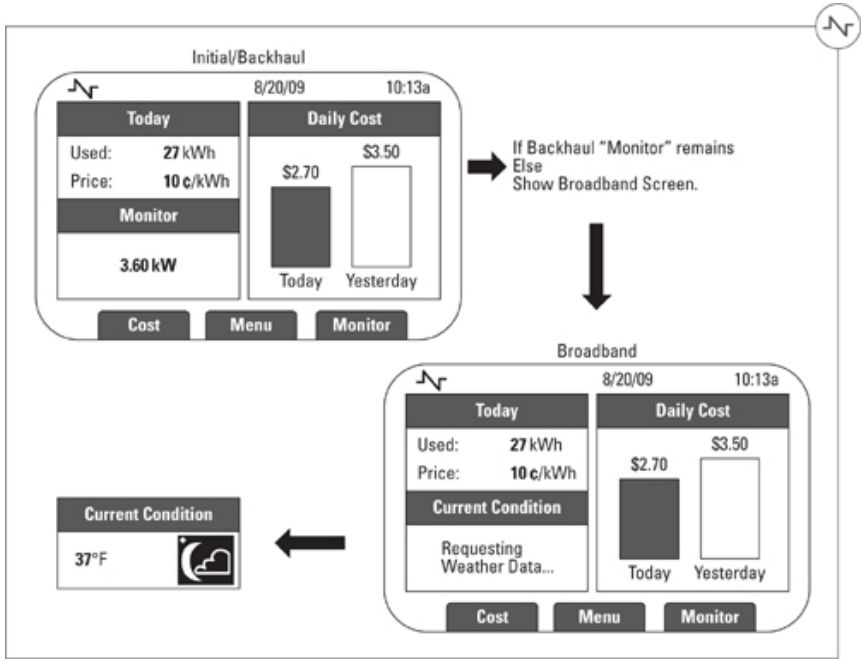
User Experience Mock-Ups

A step up from there is to use a simple drawing tool to illustrate preliminary page design and also flow between pages.

Quick pencil sketch



Preliminary flow design



Ethnography

- Comes from anthropology, literally means "writing the culture"
- **Ethnography is an observational technique that can be used to understand operational processes and help derive support requirements for these processes.**
- Essentially seeks to **explore the human factors** and **social organization of activities** → **understand work**
  - Studies have shown that work is often **richer and more complex** than is **suggested by simple models derived from interviews**

Ethnography

- Discoveries are made by **observation** and **analysis** (**initially workers are not asked to explain what they do**)
  - Collect what is ordinary/what people do
  - Study the context of work and watch work being done
- Next, ask user to explain everything he or she is doing
- Session videotaping
- Can be supplemented later with interviews/questionnaires

Ethnography

Useful to discover for example:

- What does a nuclear technician do during the day?
- What does his workspace look like?

Less useful to explore political factors:

- Workers are aware of the presence of an outside observer

Competitive Analysis

- Who is our competitors, and what features are they offering?
- Hire market survey companies that will research a domain and produce reports outlining prospective benefits of a product proposition.
- Most companies do their own competitive analysis; that way, they gain the knowledge directly.
- In its simplest form, a **competitive analysis** consists of the following:

- Refining the domain of interest/product category
- Identifying competitors
- Studying competitive offerings
- Preparing an analysis of the finding for evaluation

Competitive Analysis

A competitive analysis is usually presented as a matrix.

Example: New product in the web search and browsing domain.

Various Features

	Product	Web Approach	Content Transparency	Page Commenting	"Stumbling"
Our Product	Our product plans	Collects every page visited.	Toolbar tags give "flavor" of a page; embedded tags in search pages allow users to jump to selected pages.	Instant messaging, "sticky notes" on pages	Based on similar useful pages.
Potential Competitors	Search Engines				
	Google, MSN, Yahoo	Finds "all" pages based on search.	Short abstract of discovered pages.	None	The user must supply topic.
	Social Networks				
	YouLicit	The user must bookmark page.	Poor.	None	Based on similar pages.
	StumbleUpon	The user must bookmark page.	Embedded categories (not actually a tag); rating into Google search pages.	Offline comments	Stumbling on category of interest.
	Del.icio.us	The user must bookmark page.	Click bookmark button to see community tags.	Offline comments	No.
	Yoono	The user must bookmark page.	No tagging.	None	User-suggested links.

Customer Change Request Systems

Many companies maintain a customer-facing enhancement request system for just this purpose.

Example customer-facing feature request system

Customer Change Request Systems

- Customers provide feedback on our roadmaps and generate new concepts for products and features.
- Product managers pull ideas from the community into their life-cycle management system.
- Integrate concepts directly into their roadmap—**without having to run traditional focus groups, use off-line surveys, or wait for an annual user conference. It all happens 24×7.**
- Developers can then more efficiently develop the right features, already vetted by customers.
- Customers can participate through the entire process—they track the features they care about, and are notified when they are released.

Customer Change Request Systems

Defect Logs

- Provide a **rich source of product requirements**, because **typically many of the reported "defects" are really enhancement requests**.

- A high number of enhancement requests for a particular capability item is a relatively unbiased form of prioritization (at least from the perspective of *existing* customers).

Use Case Modeling

- When it comes to systems of complexity—systems that are composed of other systems, systems that contain hardware devices and software components, and suites of applications that work together to provide even higher value to the user—neither features nor user stories are sufficiently capable constructs to describe this complex, *aggregate* behavior.

- For these systems, use cases describes the interplay among the actors (users, devices, subsystems) and the various systems that work together to deliver this behavior.