

Requirements Elicitation

Human Computer Interaction Research Group
Department of Computer Science

Finding out what users want

- We know by now (hopefully) that we need to design interactive systems so they do what users want them to and are easy for users to understand
- But how do we find out what users really want?
- Unfortunately, it is still very typical for developers to say “we could be users of this system and we like/understand/can use it, so it must be OK”

Finding out what users really want

- Even if you are part of the target user group, if you have helped develop a system, it is inevitable that you will like and understand it, so trying to establish requirements and then evaluate it on yourself is **invalid**
- You must have independent people, outside the development group, whom you involve in establishing requirements and doing evaluations

Finding out what users really really really want

- Ideally they should cover the full range of users - not just the core group [so the development team probably is not diverse enough anyway]
- So need to think about the range of users for the system - men/women, children/adults, young/middle-aged/old, expert with technology/novice with technology ...
- Make sure the full spectrum is represented in the people you involve in requirements and evaluation

Why do we need requirements?

- In any interactive system, it is important to answer the questions of:
 - What should the system do? And
 - What qualities should it have?
- There is always a deep temptation to design for ourselves – but we are only 1 individual and usually we are separate from the domain of the users for whom we are designing
- Assumptions about requirements lead to massive problems down the road
 - Remember choose and book? Primarily a failure to understand the requirements of doctors and patients

What is a requirement?

- A *requirement* is a statement of something that a system should do or a quality that a system should have.
- In traditional requirements engineering each requirement should be:
 - Correct
 - Atomic
 - Precise and Unambiguous
 - Design Independent
 - Testable
 - Traceable

What is a requirement? (2)

- The statement of testable is an interesting one
 - each requirement should have a *fit criterion* that describes how you know your system meets that particular requirement
 - Requirement: The bank customer must receive their money from the ATM within 15 seconds of completing the transaction.
 - Fit Criterion: 95% of all customers receive their money within 15 seconds.

One view of requirements

- User Requirements
 - What the interactive system should do for a user
 - Example: The **bank customer** must receive their money from the ATM.
- System Requirements
 - How the interactive system will fulfill the user requirement (including technology)
 - Example:
 - The **ATM** will have a dispensing slot which will dispense currency notes for the country.
 - The **ATM** will display a prompt when cash is dispensed.
- Often there are many system requirements to a single user requirement.

Another view of requirements

- Functional requirements – what the interactive system does
 - A bank customer receive cash in denominations of £5, £10 and £20.
- Non-Functional requirements – what qualities the interactive system should have
 - A bank customer will complete their transaction without 90 seconds of the transaction being started.
- Constraints – qualities that the interactive system has regarding restrictions
 - A bank customer will only receive money if they have a valid card and an associated valid PIN.

Representations of Requirements

- Requirements can be expressed in many different forms
- Traditionally, we have lists of user/system requirements that people work their way through – and that still is largely how software is developed outside of agile approaches
- In UCD, we usually have a stage before we reach those sorts of lists – we have a variety of tools in which we represent requirements such as personas and scenarios – more on those in the next lectures

Data requirements

- Lots of interactive systems deal with data and users have requirements about how it is handled
- e.g. mobile phones - data (i.e. calls) must reach us (does your system cope with these requirement?)
- Banking systems - data must be secure, must be available 24/364, must persist over years

Environmental requirements (contexts of use) (1)

Physical requirements

- System must be usable in wide range of lighting conditions - bright glare, evening
- System must be usable in noisy conditions
- System must be usable in extremes of heat and cold (users with gloves on, hot/sticky fingers)
- Compare with office bound, desktop systems

Environmental requirements (contexts of use) (2)

Social environment

- Will user be solo or trying to collaborate with others?
- Laptop computers specifically designed for solo use and to inhibit eavesdropping, but often one wants to share the workspace
- Will collaborators be in the same physical space or remote from each other?

Environmental requirements (contexts of use) (3)

Organizational requirements

- Will the user have technical, training support?
- Is sharing and collaboration encouraged, or is authority and hierarchy important?
- What is the communications infrastructure - broadband, stable, omnipresent
- No good developing a complex, web-based shared workspace if some partners only have a primitive email connection

Usability requirements

- Capture the usability goals and (measurable) criteria
- Might be rather general “easy to use”
- Or very specific
- “95% of users between the age of 18 and 60 should be able to withdraw £50 (effectiveness) within 60 seconds (efficiency) and give a rating of satisfaction of at least 4/5 (satisfaction)”

Usability criteria

- Good to have a specific, testable usability requirements for each component of the definition of usability:
 - Effectiveness
 - Efficiency
 - Learnability
 - Memorability
 - Satisfaction
- Plus something on the range of users to whom these criteria are applicable

Usability criteria for C&B

- **Effectiveness:** the system will allow GPs to book hospital appointments for their patients - providing patients with at least 4 hospitals to choose from, particular consultants if appropriate, and an appropriate range of times and dates for the appointments
- **Efficiency:** the mean time for making an appointment should be 2 minutes; no appointment should take longer than 3 minutes.
- **Learnability:** a GP should be able to make an appointment in 2 minutes after finishing a 15 minute online training package; a benchmark set of 5 dummy appointments is provided for testing purposes.
- **Memorability:** a GP should be able to return to the system after a break of 2 weeks, read a 1 A4 page “reminder sheet” and make an appointment in the 2 minutes
- **Satisfaction:** after 6 months in service, a random sample of 100 GPs should give a mean satisfaction rating on the System Usability Scale (SUS)

Eliciting Requirements

- Whenever we start designing or re-designing an interactive system we need to understand what people want to do with it



Eliciting Requirements Challenges

- We can't just walk in and say "What do you want to do?" – people often cannot express the goals they want to accomplish
- If something is particularly novel then they may have no idea what it should do for them or how it should do it
- We need to ask questions about what people do in the existing domain, what problems people have, what things they like, what they don't like, what happens when they succeed/fail
- Then we analyse the data we collect and try to work out what the possible solutions are

Requirements Elicitation Techniques

- Questionnaires
- Interviews
- Focus groups and workshops
- ~~• Naturalistic observations~~
- ~~• Think aloud protocols~~
- ~~• Indirect observation unobtrusive methods~~

Questionnaires (1)

- Written format, needs very careful work (and piloting and refining a questionnaire) to ensure that questions are absolutely clear to all respondents and that you are collecting all the information you need
- Good when the issues you want to address are well defined (e.g. finding out the problems users are having with a current system which needs improvement)

Questionnaires (2)

- Less good for situations where the questions are not well defined (e.g. when developing a very novel system)
- Good way to get information from a lot of people in a way that can be relatively quickly analysed
- But the response rate may seem low - getting 40% of questionnaires returned is considered good for a postal questionnaire
- Can now do questionnaires via the Web - hard to tell what the response rate is - not an “easy” option
- In a recent study, we send out 500 emails to individuals, organizations, mailing lists - we have 53 responses

Questionnaires (3)

- The more effort you put in, the higher response rate you will get and the better the data will be
- If you write follow up, reminder letters, emails, phone calls, offer raffles ... all improve response rates
- Questionnaire should be too long to complete - length needs to be commensurate with the importance of the topic, but 20 minutes is LONG

Question formats (1)

- Three formats of questions:
- Open ended - completely free response
 - “what did you most like about this system?”
 - Good when you want to elicit all kinds of information, want respondents to be creative (they often enjoy this), don’t know what they might say
 - Difficult to analyse, particularly if you have lots of respondents

Question formats (2)

- Need to develop your own categories to group the answers
- problem then to decide whether two differently worded answers are the same category
 - e.g. memory aid for passwords
 - “name”, “friend’s name” “pet name” - are these similar enough?
 - often need two people to go through the answers

Question formats (3)

- Can use open-ended questions in a pilot version of the questionnaire given to a small number of people, then turn into a list of options →
- CLOSED QUESTIONS - yes/no, or limited set of options, with perhaps a “don’t know” or “other” to catch cases you haven’t thought of (don’t want more than 10% in these categories)

Question formats (4)

- Closed questions are easy for respondents to answer and good for getting lots of data that's easy to process and understand
- But fundamentally categorical in nature (answers “what?” questions), doesn't help with how much people liked something or how easy people found something to use
- These quantitative questions are very useful in designing systems - may want to compare to designs, or compare a design over time

Question formats (5)

- Very simple and neat way of measuring the how much type questions
- Likert or rating scale (after Mr. Likert)
- “The arrangement of icons was easy to remember” (circle one cross on the scale)

+-----+-----+-----+-----+-----+-----+

agree

disagree

- Can use 5, 7, 9 gradations depending on what you think the respondents can discriminate

Questionnaire checklist (1)

- Make questions clear and specific

CLEAR

- Avoid double-barrelled questions where the answers will be ambiguous
- “Do you think it would be helpful to make the icons or buttons larger?”
- Answer: Yes (but is it referring to the icons or the buttons?)

Questionnaire checklist (2)

SPECIFIC

- You do the work to generalize, not the respondent
“How many hours do you use the internet in a week?”
- Better to ask about specifically about yesterday, otherwise respondent is trying to average/add up in their mind - you then x 7 (although then you need to think about weekdays, weekends)

Questionnaire checklist (3)

Think about the ordering of questions

- Ask easy questions first, get the respondent relaxed (lulled into a false sense of security)
- Ask personal questions last (people don't like answering these, but if they've put effort into all your other questions, they probably will do a few more)

Questionnaire checklist (7)

Make scales intuitive, clear and consistent

- If you use numbers, 1 = low agreement, 5 = high agreement is usually intuitive (like an exam score)
- It's easier for the respondents if you don't mix negatives and positives, but you may want to keep respondents on their toes (avoid "blind agreement", make them think!)

Questionnaire checklist (8)

Avoid technical or HCI jargon - speak the language used by respondents (may need a pilot)

e.g. “Is this web site easy to navigate around?”

Do the users of e-shopping web sites actually think about “navigation”, what language should we be using instead?

Interviews

- Interviews can range from the very formal, rather like a questionnaire given face-to-face to very informal, where the interviewer has a set of topics, but has a very free ranging conversation with the interviewee
- A lot of the principles of questionnaires still apply - clear question design, ordering of questions etc etc

Interviews (2)

- Advantage over questionnaires -
 - allows you to develop a relationship with the interviewee,
 - because you are talking, can elicit much more information
 - Can explain things that the interviewee might not understand
 - Take tailor the questions to the individual much more easily

Interviews (3)

- Disadvantages
- Much more time-consuming for the interviewer
- Interviewee may be intimidated, may not reveal personal information
- More prone to “researcher bias” - telling the interviewer what they want to hear, not telling them things they don’t wish to hear

Focus groups

- Bring together 5 - 7 people, usually from a particular user group, to discuss a particular system/problem etc
- Facilitated by a researcher who has a list of questions/topics to be covered, needs to discreetly guide the discussion
- Typically lasts about 2 hours
- Time-efficient way of gathering information
- Not good to eliciting personally sensitive information
- Good for bouncing around ideas - people can spark ideas off each other

Finally: ethics

- If we involve people in research, we have an ethical obligation to inform them that we are collecting data from them and using it
- This is often a problem with observational data, particularly if collected in a public place and this ethical obligation is often ignored
- Unobtrusive measures are attractive specifically because they do not upset the naturalness of behaviour - informing people would definitely do so
- No easy way out of this - but get people's consent whenever you can

Summary

- We need to ensure that we are engaging with users early on – not finding out what users want is a recipe for disaster
- There are lots of ways to elicit data – many of which have advantages and disadvantages
- We will come back to think-aloud protocols when we get to evaluation

Readings

- Cooper et al About Face 3 – Chapter 4 – Qualitative Research and Requirements
- Rogers, Sharp, Preece Interaction Design - Chapter 7 – Data Gathering