Week 3

Pre-test and post-test for basic solution

The essence of **Think-Aloud**

You ask participants to do a set of well-defined and clearly expressed **concrete tasks** (in Asst 1, this is provided to you – the user should work through the teaching materials)

As they do each one, they think-aloud

**You carefully observe and WRITE/RECORD what happens, what user says Easy to state ….**

**Less easy to do well enough to gain benefits**

**BUT even beginners can do it well enough for benefits**

Facilitating think aloud

ONLY WHEN REALLY NEEDED, ask:

• What are you thinking now?

• What do you think that message means? (only after the user has noticed the message and is clearly spending time on it)

• if user appears surprised, Is that what you expected to happen?

DO NOT HELP THE USER

**• except if the user is distressed**

• or if they seem uncomfortable, ask:

• How do you think you can do it?

Things to record for each task

• Did the user succeed

• Time taken in minutes

• Observations (what the user **did, said**, and other observations, especially errors)

eg for a participant studying CUSP to find out who is teaching this subject

• D: Read the menus

**• S: Said “Overview” seemed the place to find the names of the lecturer and tutors**

• O: Smiled on seeing this section

**• S,E: Said there do not seem to be any tutors for this subject**

Recruiting users

• How representative are they?

– similarity to intended user population

– Age

– Gender

– experience in area

– interest/motivation

– computer literacy

• What effect does user population have for conclusions?

How to do a think-aloud study?

“己所不欲 勿施于人”

Why is the think-aloud the most important empirical usability method?

**Benefits of think aloud** #1

It is very effective for evaluating the **learnability** of an interface

You actually **see** them as they learn to use it.

Similarly, you can see how they cope with **errors** – do they recover from them?

**Benefits of think aloud #2**

It is quite **inexpensive** since you do not need to do it with many participants

In fact, if you find really serious problems in the first 1-2 user studies, you can **immediately stop** and fix those

Benefits of think aloud #3

It enables you to understand **WHY** the user had difficulty with an interface.

Benefits of think aloud #4

Since you can tell **where people had difficulties**, you know where changes need to be made to improve the interface.

Benefits of think aloud #5

you can learn how **satisfied** the participants are with the interface.

Benefits of think aloud #6

it can be quite effective even when done by **inexpert evaluators**

What is an in-the-wild usability test? (Also called field, authentic testing)

In-the-wild (versus think-aloud)

**• Study of users making use of an interface in their normal lives and contexts (黑色 = inthewild)**

• in contrast the think-aloud which is a lab-method which is

• in an artificial setting (红色 = thinkaloud),

• where people do things the evaluator asks them to do, even though they may never do them in their normal lives

• Where the user is observed (and so is very likely to behave quite differently from the way they would in their normal lives

• Where the user has to think-aloud, which most people do not normally do and where the talking about the interface alters the time it takes to do the tasks

• Can involve very few … up to many, many users

• think-aloud involves very few users

• Ranging from ethnography … to highly **automated software monitoring** of interface(quantitive)

• think-alouds gold outcomes are the **qualitative** results

**Large scale** study give mainly **quantitative** data about usability

* It is very easy to measure **WHAT** people do but it is really hard to work out **WHY** they did it if you only use in-the-wild monitoring

**In summary:**

**In-the-wild** testing is about studying usability testing in **authentic** settings of use

It is very **widespread** and an important for is on websites - giving **large** amounts of data

For usability testing, it is best **combined with think-aloud** to get the best of both methods

How to address mental model mismatch problems

• **Fix the system** – make it match users' mental models

• eg If people look for something in the wrong place, then move it to the place where they look for it.

• **Fix the user** - improve users' mental models so that they more accurately reflect your system.

• eg explaining things better

• making labels clearer to make the UI more transparent (even though the underlying system remains unchanged)

• Help that insists that the user pays attention

**Core learning objectives for Week 3 evaluation**

• How does evaluation link with analysis?

• What is a **think-aloud usability test**?

• Why is the **think-aloud the most important empirical usability method**(benefits?)?

• What is an **in-the-wild usability test**?

• What are **qualitative** and **quantitative** user study methods?

• What are the **comparative roles** for think-aloud and in-the-wild usability tests?

*• [Lab] Conduct a think-aloud usability study of an interface, with ~5 users, given the interface, and the set of tasks, ensuring that the observations are carefully recorded and analysed to draw conclusions about the overall usability and key problems.*

• How do **mental models link to think-alouds, and in-the-field trials**?

Week4

What is a prototype in HCI?

A prototype is a **draft version** of a product that **allows you to explore your ideas and show the intention** behind a feature or the overall design concept to users before investing time and money into development. A prototype can be anything from **paper drawings (low fidelity)** to something that allows click-through of a few pieces of content to a **fully functioning site (high-fidelity).**

Prototyping: Low-Fidelity (lo-fi)

**Quick** to construct

Easy to **explore many design alternatives**

Have limited or no functionality

Aim to show the general look and feel

Help **communication** and exchange of ideas with users

People may be **more willing to criticise**

**But: limits testing possible**

High Fidelity (Hi-fi) prototypes

Hi-fi prototypes

**Actual working system** from user perspective

eg can interact but no real backend database

Partial to complete functionality.

**Enables users to explore with the system.**

cf. lo-fi enables users to follow just a small set of paths

But

More **costly** to build for sophisticated interfaces

**Reluctance to change** the design

Users involved in evaluation often focus on superficial finish

Management may think they have a full system

A single bug can lead to a complete halt in evaluation

Information architecture (IA)

• This describes the organization of the information in an interface.

• An effective Information architecture will match the user’s mental model of the structure of the information space.

• An effective information architecture is organized so that

• Users can navigate through the interface to meet their needs

• To be effective, the information architecture of an interface needs to take account of

• the users,

• their context

• and the domain

**Week 5**

Cognitive biases

Google effect– or digital amnesia

We tend to forget things that we could readily look up, rather than remember. We essentially delegate some of our memory to digital systems we can search.

Over‐confidence

Interface designers/programmers are confident that their interface is fine They may be reluctant to believe that people cannot use it.

Stereotyping

-Interface designers/programmers may make many assumptions about the users of their system.

-How does the usability engineer avoid stereotyping errors?

Do user research to learn more about users.

Evaluate with think‐aloud to learn more about.

Confirmation bias

People tend to look for, see and remember in a way that maintains consistency with their current beliefs

Social desirability bias

* People tend to answer questions in a way that ensures the questioner will take a positive view of them
* This makes it hard to do analysis based on either asking people  questions or observing them when they know they are being observed

Problems people face in making rational decisions

• **Overload – there is a lot happening in the world around us**: The world has too much information for our brains to take it all in

• **The world is confusing, noisy:** We need to make sense of the information we do take in.

• **We often need to act quickly.** So we may not be able to think carefully through a complex decision such as one involving probabilities and complex evidence.

• **We cannot remember everything** – even from what we do take in ‐ and our memories change over time, both adding and forgetting details.

How people deal with these problems

• **Overload: Humans don’t perceive (see, hear, feel, smell,….) everything** – we only actually see some things – so may miss information that would be useful if we did see it

• **Meaning: Humans search for meaning – potentially confabulating to ensure consistency and confirm assumptions.** (Confabulate means fabricate imaginary experiences as compensation for loss of memory)

• **Speed: Sometimes “quick thinking” can be flawed.** Thinking fast … thinking slow.

• **Memory: Our memory and mental models can reinforce errors**. Some of the stuff we remember for later just makes all of the above systems more biased, and more damaging to our thought processes.

Recasting this for HCI

• Overload: Humans don’t see everything. We only actually see some things on an interface and our mental models play a role in what we see

• Meaning: Humans search for meaning. We try to make sense of interfaces, interpreting interfaces based on our mental models and what we see.

• Speed: Sometimes “quick thinking” can be flawed. We may act as soon as we perceive (usually meaning we see) something relevant, based on our mental, what we saw, how we interpreted it.

• Memory: Our memory and mental models can reinforce errors. Some of the stuff we remember for later just makes all of the above systems more biased, and more damaging to our thought processes.

What does **intuitive** mean（简单易用）

Does not need conscious thought.

So it is effortless for the user to interact based on their current knowledge

All interaction with technology has to be learnt for the first use

Very familiar interactions become intuitive

**Human factor**: develop a mindset that recognize importance of user, work practice, organizational context

**HCI**: study of human interaction with digital tech, includes creating effective interface and system to meet people’s needs.

**Usability**: learnability(how ez first), memorability(how ez after break), efficiency(how quick), errors(how many, how quick recover, how severity), satisfaction(how pleasant)

Why **security** is important: human vulnerability, important for user, design fail in human factor

**Context**: a certain situation. Solution for one isn’t for another, solution not for one is not effective.

**Think like user**: understand domain, context of use, user knowledge, user experience, what to see

To **design test**: define what they need/want, need analysis of user goal

**Aesthetic**: may not be effective/efficient. Screen design is gud for aes/use but limited

**Mental model**: world view, current knowledge, detailed goals

**SSM:**  approach to solve problem involving soft system, start by identify **user needs, key factors, stackholders**. We consider **situation of concern, CATWOE(client, actor, transformation, worldview, owner, environmental constraint)** ,to get over challenge of **describing** and **understanding** problem.

**Soft system**: no clear defined solution, we don’t find for optimal solution, we find good enough solution created by iterative approach(create&test), introducing change will alter system.

**Concern for us**: phishing and social engineering, spoofing techniques to impersonate a trusted entity to conduct highly deceptive phishing attacks, by careful design and timing of a message, use people’s curious and interest.

**Penetration test, Security awareness training**, : learn which types of users pose the most risk for specific types of attacks while also identifying which employees require additional training. If people know what forms social engineering attacks are likely to take, they will be less likely to become victims.

**Spear phishing**: send a specially crafted email to very select individuals, by research targets, create customised email, for high value target.

**SSM2222**: concern, transformation(new system objective), stackholders’ worldview(mental model, what they think), context, environmental constraint.

**3 key methods for user need&pref**: study behaviour, what they say, trustworthy previous work

**Behaviour**: automated monitor- track behaviour, used for embedded training|ethnographic-observe in context and observer immerses themselves in small number ppl to get deep qualitied result,give richer data to auto monitor, works only if observers can present on time, need training|lab study- bring ppl into lab.

**Say**: in context-questionaire, out context-after lab, small group, hybrid

**Asking**: self awareness, self censorship, politeness, joke or lie

**high level user goals**: identify the goal based on all methods

**persona**: a representation of user group, need&expectation, how likely to use system, universal feature&functionality, real ppl with background, goal, value. Used as communication tool, discussion, design trade off. Do user need research, study similar&diff, identify group.

**Thinkaloud**: ask users to do concrete task and record what they say. dont help unless they are unhappy. Need to consider representative. Record success, time, did, said. Tells mental model.

**Why important**: effective with learnability and error, inexpensive, why they feel hard, where they feel hard, how satisfied, effective for inexpert evaluators

**In the wild**: study of ppl daily live. Have many user, produce quantity result, easy to know what but hard to know why

**Qualitive and quantitive**: non-numbers and numbers.

**Link to mental model**: think aloud tells mental model, fix the system or fix the user(mental model)

**Prototype**: draft that allow u to explore idea and show intention behind feature or design concept. Do evaluation to inform next version(iterative)

**Lofi**: quick, many alternative, help communicate, willing to criticize, limit test possible

**Hifi:** actual working system, enable explore, costly, reluctance to change

**IA:** organization of information, match mental model, take account of user, context, domain. eg. Alphabedical, chronological, geographical, topical, task oriented, metaphor-driven, sitemap.

**Card sorting:** group cards. Open for creating own category, closed for pre-defined category.

**Mood board**: collect ideas useful, like comic, games, videos, to explore way to teach

**Screen design guidelines:** line length-ppl scanread cuz retina range is small, recommend 60 char&ppl may resize->responsive design. Fonts-> no more than 3, standard cuz more familiar, read faster. Use mixed case, line spacing, white space, text-background contrast, color blind, avoid blinking text-distract

**Ppl’s attribute**: mental&physical, which can be mostly similar or individual differ. And context which one ppl differ at different time.

**Physical**: perceive: see, hear, feel --- action: type, touch, speak

**Visual perception**: retina range, color sensitivity, color blindness-5%

**Cognitive bias**: ways human reasoning may not be rational.

**Google effect**- forget things easy to search on google

**Over confidence**- about performance, designers can be over confident about usability, think aloud can help them

**Stereotyping**: gud-> simplify complex stuff, reason based on incomplete evidence, make fast decisions. Interface designers/programmers may make many  assumptions about the users of their system. Do more user research and evaluate think aloud can help.

**Confirmation bia**s: ppl tend to believe what they already believe. it may be hard to change people’s belief, and in some case, even contradict evidence to their belief will help strengthen it. It will be hard for designers to give people information in some case, like when that information is contradicting to users’ mental model.

**Social desirability bias**: People tend to answer questions in a way that ensures the questioner will take a positive view of them. this makes it hard to observe/ask when they know they are being researched. Field trail will help.

**Problem ppl may meet:** information overload, world is confusing, we need to act quickly, we cant remember everything.

**How cog-bias help**: ppl don’t perceive everything, ppl search for meaning, ppl think quick, memory can make error

**Intuitive**: ez2learn,ez2remember,ez4errors,pleasant to use->ppl need background knowledge2use

**Usability for all**: focus on special need, but also include all requirement. Eg: physical, cognitive, social, cultural

Week7

**Privacy:** claim of individuals groups or institutions of **when, how, and to what extent** information about them is communicated to others he **balances** the desire for **privacy** with the desire for **disclosure and communication.**Have legislation controlling it,like personal data on computers

For below (a,b): x-axis: accuracy of personal information. Y-axis: amount of information

**Secrecy:**keep things hidden(1,1).**confidentiality**:keep someone else data hidden(2,1).**Availability**:oppose to secrecy.**Anonymity**:keep the identity of the person associated with the data secret(1,2).**privacy**:principles/rules to control availability of data. **Auditability**:able to keep a record of data made available. **Access control**:the rules for controlling privacy

**privacy involves a balance of secrecy and availability**

**ppl vary in privacy**:by their worldview, mental model and other factors. Diff privacy pref with diff ppl and diff things.ppl show similar comparative assessments for certain classes of personal data

**privacy paradox**: actual behavior is inconsistent with beliefs.eg. register without read terms.

**privacy by design**: methodology let privacy be built in architecture of information systems, business processes and networked infrastructure.aim to consider privacy before and through the initiatives that involve the collection and handling of personal information.

**Features**:Proactive not Reactive,Preventative not Remedial;Privacy as the Default Setting;Privacy Embedded into Design;Full Functionality-Positive‐Sum, not Zero‐Sum;End‐to‐End Security-Full Lifecycle Protection;Visibility and Transparency;Respect for User Privacy-Keep it User‐Centric

**right to be forgotten**: delist from search results articles relating to one’s past crimes.

WEEK8 Security Mindset, Terminology, Threat Models

**Security**: build and maintain systems that remain safe to use and dependable in the face of malice, error, and mischance.

**Mindset**:understand the big picture and specific detail.require:1.to think like the attack for what,how,why,and for money/fame.2.to think like user for get job done and low usability like complex warnings. Security is by design not afterwards.

**Map security space**: Policy: What supposed to achieve; Mechanisms:how to implement policy; Assurance: Amount of reliance placed on a control; Incentive: How to motivate those policy.

**Systems:ppl** use **technology**(software, hardware) to perform **activities**.security is keep work gud

**Ppl**:security depend on behaviours.ppl are not machines, not rational.ppls differ in many ways.

**Risk management**:quantity\*chance;human is bad on this;fear, uncertainty, doubt->cloud thinking

**Adages**:hide don’t make things safe,only frustrate attker.assume he can find,never rely on hide.

**Def in depth**:only raise cose/time for attker,avoid eggshell model->one ctrl fail->not full sys fail

**Vulnerability**:Weakness in system at implementation level;known or unknown;eg.buffer overflow

**Flaws**: Weakness in the system at design level; cannot always be addressed after sell product.

**Threat**:diff meaning depend on context;all high-level environmental factors a system expose to.

**Attack**: Activity that intends to cause harm to the system;can happen even no vulnerability.

**Controls**: A measure to defend the system by avoiding risk or attack, detecting it, or mitigating it;can be Preventive, Detective,corrective;can be technical, but may not Physical, Operational, Regulatory.

**Trust**:trusted system will operate as intended. trusted component if its failure would compromise the system's security. trustworthy component if we can safely assume that it is not going to fail

**Confidentiality**: the obligation to protect secrecy;is a security goal.

**Integrity**: be able to verify that information has not been(illegaly)altered

**Auditability**:find out what happened(who did);legal requirement may exist;useful after failure.

**Availability**: provide service for the intended users;require: process&acceptable time.

**Anonymity**: do things without being identified;may conflict with other aspects;perfect anonymity is impossible

**Non-repudiation**:identity/action/belonging is not able to be challenged/disguised.

**Trade-off**: control has costs as well as benefits;both harder for attkers&users.

**Threat modelling**:context matters,eg:what’s protecting?who’s attacking?how long?attacker’s motivation,organisation;social engi can be used to breach from human side.

Week9 Operational Security and Authentication

**Operational security**: defend against attacks against procedures and processes, not technology;asymmetrical, no tech required;bypass tech defense,attack psychological weakness that are normally beneficial in a social context, by social engineering.

**critical information**: your own intentions and capabilities that enemy need to act against you

**Analyze threat**: Systematic analysis potential opponents:who, motivation and goals, capabilities, determination.

**Analyze weaknesses**: assume the opponent's role and think like them: Which parts of your own organization show weakness? Which practices could be exploited? What equipment do you use could potentially harbour a weakness?

**Assess risks**: Assign a risk level to each identified weakness;help decide investment.

**Apply countermeasure**: prioritize the risks; mitigation plan:tech&non-tech measures; security is process, not state(threat&countermeasure change);business consideration.

**Pretext**:pretend to be someone else;policy against:call back.

**Phishing**: policy:disallow email for high value transaction.

**Baiting**: Tricking someone into accepting/using a malicious gift.policy:education,security culture.

**Tailgating**:walk behind authorised ppl to restricted area. Only in physical space;policy:guard.

**Quidproquo**: promise of a benefit in exchange for some information;defense: ensuring critical information flows are need to know rather than need to withhold.

**AAA (Triple-A)**: Authentication, authorization, accounting

**Authentication**: "what you know, what you possess, what you are"

Week10 Software security

**Memory**: **Stack**: local variables, function return addresses;**Heap**: dynamically allocated memory; **Global**: global variables; **Code**: program instructions;

**buffer overflow**: program defect triggered by crafted input, Overrun the part of a data structure with data that will cause execution of malicious code.**steps**: Find the overflow, Change the control flow(memory loc), Insert foreign program, Execute program.

**Atker**: Need binary machine instructions to overwrite stack; Code must have absolute entry address; once found successful attack vector , it can be reused.

**Defer**: Understand how memory is accessed; check validate inputs&sanity check

**Heap overflow**: memory pointer was changed, could overwrite files.

**Time of check&time of use**:atker change status of file between system check the file and open it.

**Def methods**:use safe version of program language;static code check(no run);dynamic code check(run);unit test;no-exec mem area;aslr(random mem);stack canaries(place value in stack and check periodically);shadow stack(use copy of stack);fuzzing(random input test);

**Online context**:information belong to a session,including cookies, javascripts, data, plugin, etc.

**Malware**:virus: Attach to programs or data , get loaded into memory,replicate and infect other programs;worm: Spreads stand-alone copies of itself via a network;trojan: Disguise as a useful software;keyloggers:record keyboard input;backdoor:debug admin features;

**Def principle**: user:never work at higher privilege;developer:check input, test, clear memory, failsafe default;operator:avoid monoculture, config manage(update).