HCI Definition: study of interaction between ppl & computer based sys, concerned with phys, physiological & theoretical aspects of this proc, about designing computer sys that support ppl so that they can carry out their acts productively & safely
User: individual user, group of users working together or a seq of users in org dealing with some part of proc/task
Computer: tech ranging from desktop to large scale sys, or control/embedded sys
Interaction: comms between user & comp
What is involved: study of humans using intf, dev of new apps/sys to support user's acts, new devices & tools for users, develop usable prod (easy to learn, effective to use, provide enjoyable & satisfying xp)
Interdisciplinary (HCI): compsci & sys design are central concerns, but not possible to design effective interactive sys from one discipline in isolation
Contributing Disciplines: cognitive psych, compsci, anthropology, engg, ergonomics & human factors, design, social & organizational psych, sociology, philosophy, AI, linguistics
Importance: how to make sys usable, evaluate usability of bespoke (custom) sys, understanding how users interact with computers & enabling users to do so effectively, matter of law (is suitable to task, easy to use & adaptable to user's knowledge & xp, provides feedback on perf, displays info in format & pace adapted to user, conforms to principles of software ergonomics)
Factors in HCI: organisational, environmental, health & safety, the user, comfort, UI, task factors, constraints, sys func, productivity factors, more:
Dev Process: design approaches, impl techniques & tools, eval techniques, example sys & case studies
Problems with Software: excessive & unwanted share dealing in stock market, error in dosage given to patients receiving rad therapy, erratic behaviour of military & civil aircraft, difficulties in controlling nuclear power plants during sys failures, delays in dispatching ambulances to accidents
Avoiding Problematic Design + what to design: take into account who users are, what acts are being carried out, where interaction is taking place, optimise interactions to they match users' acts & needs
3U: Useful (accomplish what is required), Usable (do it easily & naturally, without danger of error), Used (make ppl want to use it, be attractive, engaging, fun)
Principles for supporting HCI + understanding users' needs: take into account what ppl are good & bad at, consider what might help ppl with the way they currently do things, think through what might provide quality user xp, listen to what ppl want & getting them involved in design, using tried & tested user based techniques during design proc
Science or Craft: bit of both (artistically pleasing & capable of fulfilling tasks required), innovative ideas lead to more usable sys (understand not only that they work but how & why they work), creative flow underpinned with science, scientific method accelerated by artistic insight
Bad Design: button & label look the same, buttons on different sides laid out different, need to push button first to activate instead of inserting bill (against convention)
Good Design: marble answering machine (based on how everyday objs behave, easy, intuitive & pleasure to use, one step acts to perform core tasks), TiVo remote (peanut shape to fit in hand, logical layout, color coded distinctive buttons, easy to locate buttons)
Interaction Design (ID): designing interactive prod to support the way ppl comm & interact in their everyday & working lives (Preece, Sharp, Rogers, 2015), design of spaces for human comms & interaction (Winograd, 1997)
ID Goals: develop usable prod (easy to learn, effective to use & provide an enjoyable xp), involve users in design proc
Interdisciplinary Contributor (ID): academic (psych, social sci, computing, engg, ergonomics, informatics), design (graphic, prod, artist, industrial, film industry)
Interdisciplinary Fields Doing ID: HCI, ubiquitous computing, human factors, cognitive engg, cognitive ergonomics, computer supported cooperative work, info sys
Working in Multidisciplinary Teams: many ppl from different backgrounds involved, different perspectives & ways of seeing & talking about things, more ideas & designs generated, but difficult to comm & progress forward the designs being created
Professionals in ID: interaction designers (design of interactive aspects), usability engineers (evaluate prod using usability methods & principles), web designers (develop & create visual design of websites, such as layouts), info architects (ppl who come up with ideas of how to plan & struct interactive prod), UX designers (all before + field studies to inform prod design)
UX: how prod behaves & is used by ppl in real world, way ppl feel about it & their pleasure & satisfaction when using it, looking at it, holding it, opening/closing it, cannot design UX, only design for UX
iPod: quality UX from start, simple, elegant, distinct brand, pleasurable, must have fashion item, catchy names, cool
ID Process: establishing reqs, dev alts, prototype, evaluate
ID Characteristics: users should be involved through dev of the project, specific usability & user xp goals need to be identified, clearly documented & agreed at the beginning of the project, iteration needed through core acts
Help Designers: understand how to design interactive prod that fit with what ppl want, need desire, appreciate that one size does not fir all, identify incorrect assumptions about particular user groups (not all old ppl want/need big fonts), be aware of both ppl's sensitivities & capabilities
Cultural Differences: date format
Accessibility: degree to which prod usable & accessible by as many ppl as possible, focus on disability (mental or phys impairment, adverse effect on everyday life, long term)
Usability Goals: effective, efficient, safe to use, have good utility, easy to learn, easy to remember how to use
Design Principles: visibility (of act, invisible auto controls can be more difficult to use), feedback (sending info back to user), constraints (help prevent user from selecting incorrect options), consistency (easier to learn & use, but can break down, ex: control + first letter but problem if multiple commands with same first letter, internal is in app, external is across apps & devices), affordances (attribute of obj that allows ppl to know how to use it, virt intf better conceptualized as perceived affordances, learned conventions of arbitrary mappings between act & effect)
Understanding Problem Space: what you want to create, assumptions, will it achieve what you hope it will
Assumption: taking something for granted when it needs further investigation (bad: ppl want to watch TV while driving, wouldn't mind payng a lot more for 3D TV, ok: wouldn't mind wearing 3D glasses in living rooms, enjoy enhanced clarity & color detail from 3D)
Claim: state something to be true when it is still open to question (voice commands for GPS is safe)
Framework for analysing problem space: are there probs with existing prod/UX, why there are probs, how do you think proposed design ideas might overcome these, if designing for new UX how proposed design ideas support, change, extend current ways of doing things
Benefits of Conceptualising: orientation (enable design teams to ask specific qs about how conceptual model will be understood), open minded (prevent design teams from becoming narrowly focused early), common ground (allow design teams to establish set of commonly agreed terms)
From Problem to Design Space: understand problem can help inform design (what kind of intf, behaviour, func to provide), important to develop CM before
Conceptual Model (CM): high level desc of how sys organized & operates, enables designers to straighten out thinking before they start laying out widgets (Johnson, Henderson, 2002), desc in terms of core acts & objs, also in terms of intf metaphors
CM components: metaphors & analogies (understand what prod is for & how to use it for act), concepts ppl are exposed to through prod (task-domain objs, attributes, ops: saving, revisiting, organizing), rel & mappings between concepts
First steps in formulating CM: what will users be doing when carrying out tasks, how will sys support tasks, what kind of intf metaphor (if any) will be appropriate, what kinds of interaction modes & styles to use, always keep in mind when making design decisions how user will understand underlying conceptual model
Interface Metaphors: conceptualizing what we are doing (surf web), conceptual model instantiated at the intf (desktop metaphor), visualizing op (icon of shopping cart for placing items into), designed to be similar to phys entity but also has own properties, can be based on act, obj or both, exploit user's familiar knowledge to help then understand unfamiliar, conjures up essence of unfamiliar act, enabling users to leverage thos to understand more aspects of unfamiliar func ex: material (card, familiar form factor, material properties added, giving appearance & phys behaviour, like surface of paper)
Interface Metaphor Benefits: makes learning new sys easier, helps users understand underlying conceptual model, can be very innovative & enable realm of computers & apps to be made more accessible to greater diversity of users
Interface Metaphor Problems: break conventional & cultural rules (recycle bin on desktop), can constrain designers in way they conceptualize problem space, conflict with design principles, forces users to only understand sys in terms of metaphor, designers can inadvertently use bad existing designs & transfer bad parts over, limit designers' imagination with coming up with new conceptual models
Interaction Types (CM): hybrid often used, support different ways to do same thing, can take longer to learn, more below
Instructing: issue command, select option, quick & efficient interaction, good for repetitive kinds of acts performed on multiple objs, ex: word processor, vending machine
Conversing: underlying model of conversation with human, range from simple voice recog menu driven sys to more complex natural lang dialogs, virt agents, toys, robots designed to converse, ex: timetables, search engines, advice giving sys, help sys, allows users (especially novice & technophobes) to interact with sys in way that is familiar (make them feel comfortable, at ease, less scared), but misunderstanding can arise when sys cannot parse what user says
Manipulating: involves drag, select, open, zoom on virt objs, exploit user's knowledge of how they mode & manipulate in phys world, can involve acts using phys controllers (Wii) or air gestures (Kinect) to control movements of on screen avatar, tagged phys objs that are maniplulated in phys world result in phys/digi events (animation)
Direct Manipulation (DM): continuous representation of objs & acts of interest, phys acts & button pressing instead of issuing commands with complex syntax, rapid reversible acts with immediate feedback on obj of interest
DM Pros: novice can learn basic func quickly, xp users can work rapidly to carry out wide range of tasks (even defining new functions), intermittent users can retain operational concepts over time, error msgs rarely needed, users can immediately see if acts are furthering goals & if not do something else, users xp less anxiety, users gain confidence & mastery & feel in control
DM Cons: some ppl take metaphor too literally, not all tasks can be described by objs & not all acts can be done directly, some tasks better achieved through delegating (spell checking), can become screen space gobblers, moving mouse can be slower than pressing func keys to do same acts
Exploring: involves users moving through virt or phys environments (with embedded sensor tech)
Interface Types (CM): kind of intf used to support mode, ex: command, speech, data entry, form fill in, query, graphical, web, pen, VR/AR/Mixed, gesture, brain, when choosing need to determine reqs & user needs, take budget & other constraints into account, also depend on suitability of tech for act being supported
Paradigm: inspiration for CM, general approach adopted by community for carrying out research (shared assumptions, concepts, values, practices), ex: ubiquitous computing, pervasive computing, wearable computing, tangible bits, AR, attentive environments, ambient computing
Visions: driving force that frames R&D, invites ppl to imagine what life will be like in 10, 15, 20 years time (Apple 1987 Knowledge Navigator, Smart Cities, Smart health), provide concrete scenarios of how society can use next gen of imagined tech, also raise qs concerning privacy & trust
Theory: explanation of phenom (info processing that explains how the mind or some aspect of it is assumed to work), can help identify factors (cognitive, social, affective, relevant to design & eval of interactive prod)
Models: simplify HCI phenom, intended to make it easier for designers to predict & evaluate alt designs, abstracted from theory from contributing discipline (psych, keystroke model)
Framework: set of interrelated concepts &/or specific qs for 'what to look for', provide advice on how to design
Why Need to Understand Users: interacting with tech is cognitive (need to take into account cognitive processes involved & cognitive limitations of users, provides knowledge about what users can & cannot be expected to do, identifies & explains nature & causes of probs users encounter, supply theories, modelling tools, guidance & methods that can lead to design of better interactive prod)
Cognitive Processes: below
Attention: select things to focus on at point in time from mass of stimuli around, allow to focus on info relevant to what we are doing but limits ability to keep track of all events, involves audio &/or visual senses, info at intf should be struct to capture attention (use perceptual bounds like windows, colour, sound, ordering, spacing, underlining, sequencing, animation), multitaskers easily distracted & find it hard to filter irrelevant info
Design Implications: make info noticeable when it needs attending to, use techniques that make things stand out, avoid clutter intf with too much info (search engine & form fill in that have simple & clean intf easier to use)
Perception: how info acquired from world & transformed to xps, text should be legible, icons should be easy to distinguish & read, group info
Design Implications: icons should enable users to readily distinguish meaning, bordering & spacing are effective visual ways of grouping info, sounds should be audible & distinguishable, speech output should enable users to distinguish between set of spoken words, text should be legible & distinguishable from background, tactile feedback should allow users to recognize & distinguish different meanings
Memory: involves first encoding (more attention paid, more it is processed in terms of thinking & comparing with other knowledge, more likely to be remembered), then retrieving knowledge, don't remember everything (involves filtering & processing what is attended to), context important in affecting memory (where, when), recognize better than recall, remember less about photographed objs than actually seen (Henkel, 2014), ppl can scan list to find one they want
Digital Content Management: memory involves 2 processes (recall directed & recog based scanning), file mngt sys should be designed to optimize both (search box & history), help users encode files in richer ways (colour, flagging, image, flexible text, timestamp)
Design Implications: don't overload mem with cmplx procedures, recog vs recall, provide various ways of encoding info
Learning: prefer to learn by doing rather than read manual, rely more on internet to look things up, expecting to have internet reduces need & extent to which we remember, enhances memory of knowing where to find it online (Sparrow et al, 2011)
Design Implications: design intf that encourage exploration, constrain & guide learners, dynamically linking concepts & representations can facilitate learning of complex material
Reading, Speaking, Listening: many prefer listening to reading, reading can be quicker, listening requires less cognitive effort, dyslexics have difficulties understanding & recognizing written words, speech recog, speech output, natural lang sys (type in qs & give text based responses)
Design Implications: speech based menus & instructions should be short, accentuate intonation of artificial voice, provide opportunities for making text large on screen
Problem Solving, Planning, Reasoning, Decision Making: involves reflective cognition (thinking about what to do, what the options are, consequences), often involves conscious processes, discussion with others or self, use of artefacts (maps, books, pen, paper), may involve working though different scenarios & deciding which is best option
Design Implications: provide additional info/functions for users who wish to understand more about how to carry out an act more effectively, use simple computational aids to support rapid decision making & planning for users on the move
App Mentality: developing in psyche of younger generation is making it worse for them to make their own decisions because they are becoming risk averse (Gardner, Davis, 2013), all desires & qs should be satisfied/answered by app, so less thinking?
Mental Model: how to use the sys (what to do next), what to do with unfamiliar sys or unexpected situations (how sys works), used to make inferences, involves unconscious & conscious processes (images & analogies activated), deep (how to drive car) vs shallow (how car works), errorneous ex: turn up thermostat to heat up quicker
Gulf of Exec & Eval: exec is from user to sys, eval is opp
Info Processing Steps: encode, compare, response sel, exec
Distributed Cognition (DC): info transform through diff media (comp, disp, paper, head) instead of in mind only
DC Involves: distributed problem solving, role of verbal & on verbal behaviour, various coordinating mechanisms used (rules, procedures), comms that takes place as collaborative act progresses, how knowledge shared & accessed
External Cognition: concerned with explaining how we interact with external representations (maps, notes, diagrams), what are cognitive benefits, what processes involved, how they extend cognition, what computer based representations can we develop to help even more
Externalizing to reduce memory load: remind that we need to do something, remind what to do, remind when to do, ex: diaries, reminders, calendars, notes, shopping & todo lists, post-its, piles, marked emails
Computational Offloading: when tool is used together with ext representation to carry out compute (pen & paper)
Annotation: modify existing representations through making marks (cross off, tick, underline)
Cognitive Tracing: externally manipulating items into different orders/structs (playing Scrabble, cards)
Design Implication: provide external representations at intf that memory load & facilitate computational offloading (info visualizations to allow ppl to make sense of & make rapid decisions about big data)
What is Proto: screen sketches, storyboard, ppt, vid simulating use, lump of wood (for size), cardboard, limited function SW written in target lang or other
Why Proto: eval & feedback central to ID, stakeholders can see, hold, interact with proto more easily than doc/drawing, team members can comm effectively, test ideas, encourages reflection, answer qs and support designers in choosing between alts
Proto Filtering Dimensions: appearance, data, func, interactivity, spatial struct (layout)
Proto Manifestation Dimensions: material, fidelity, scope
What to Proto: technical issues, workflow, task design, screen layouts & info display, difficult, controversial, critical areas
Low-F Proto: use medium unlike final, quick, cheap, easily changed, ex: storyboards (sketch series, early), card based (often for web), wiz of oz (dev acting as sys, hidden)
High-F Proto: use materials expected to be in final, look more like final, can be dev by integrating existing HW & SW components, danger that users think they have complete system
Proto Compromises: SW based proto may have slow response, sketchy icons, limited func, horizontal (wide range of func, little detail), vertical (lot of detail, few func)
Conceptual Design: transform user req/needs to CM, consider alts (proto helps)
Metaphor Eval: how much struct does it provide, how much is relevant to prob, is it easy to represent, will users understand, how extensible is it
Expanding Init CM: what func will prod do (what prod do & what user do), how func related to each other (seq or parallel, categories), what info needed (data req for task, how data transformed by sys)
Concrete Design: many aspects (colour, icons, buttons, interaction devices), user characteristics & context (accessibility, cross-cultural design), successful products are bundles of social solutions, inventors succeed in particular culture because they understand values, institutional arrangements, economic notions of culture
Using Scenarios: express proposed or imagined situations, used in various ways (basis for overall design, script for user eval of proto, concrete ex of tasks, as means of coop across prof boundaries), to explore extreme cases
Explore UX: use card based proto or stickies to model UX, called design/customer/XP/user journey map, can be wheel or timeline
Proto Construction: phys (electronics, Arduino), SDK
Emotions and UX: HCI traditionally been about designing efficient & effective sys, now more about how to design sys that make ppl respond in certain ways (happy, trusing, learn, motivated)
Emotional Interaction: what makes us feel stuff, why ppl become emotionally attacked to certain prod (virt pet), can social bot help reduce loneliness and improve wellbeing, how to chg human behaviour through use of emotive feedback, emotional state changes how we think (if frightened/angry more likely to be less tolerant, if happy more likely to overlook minor prob & be more creative)
Expressive Intf: provide reassuring feedback that can be both informative & fun (but can also be intrusive, annoying or making ppl angry), colour, sounds, icons, graphics & anim used to make look & feel of intf appealing (convey emo state), can affect usability (ppl prepared to put up with certain aspects of intf if end result is appealing & aesthetic, ex: large graphics slow dl)
Friendly Intf: 3D metaphors based on familiar places (living rooms), agents (bunny, dog) incl to talk to user, make user feel more at ease & comfortable
Frustrating Intf: app doesn't work properly/crash, sys doesn't do what user wants it to do, unmet expectations, sys does not provide enough info to enable user to know what to do, vague/obtuse/condemning error, appearance of intf garish, noisy, gimmicky, patronizing, sys req users to do too many steps for task, then discover mistake made earlier and need to start all over
Gimmick: amusing to designer but not to user (site under construction)
Error Msg: avoid fatal, invalid, bad, audio warnings, uppercase and code numbers, vague, provide context-sensitive help
Emotional Tech: measure facial expr, gesture, body movement, aim to predict user's emotions and aspects of behaviour (what user most likely to buy online when feeling sad, bored, happy)
Emo Data Use: adapt content to match user's emo state
Indirect Emotion Detection: beginning to be used more to

infer/predict behaviour
Persuasive Tech: designed to change attitude & behaviour (Fogg, 2003), referred to as nudging (pop up ads, warning msg, reminder, prompt, personalized msg, recom, Amazon 1-click, pocket Pikachu, tracking devices, visualize electricity usage)
Anthromorphism: attributing human like qualities to inanimate objects, much exploited in HCI (make UX more enjoyable, more motivating, make ppl feel at ease, reduce anxiety)
Anthromorphism Pros, Cons: positive impact, more willing to continue with kind feedback, but deceptive, make ppl feel anxious, inferior, stupid, many prefer impersonal, make users feel less responsible for actions
Virt Chars: sales agents, game char, learning companions, wizards, pets, newsreaders, provides welcoming persona, has personality and makes user feel involved, but can lead ppl into false sense of belief (confide personal secret), annoying and frustrating, may not be trustworthy
Virt Agent Believable: extent to which users believe agent's intentions and personality, appearance and behaviour
Global Intf Issues: individual's response affected by factors (age, gender, race, sexuality, class, religion, political persuasion), exported SW needs modifications to suit local customs, laws, conventions, dev of multiple intf costly, so need to make generic & easily modifiable intf
Intf Specialization Lvl: globalisation (cultureless std), internationalisation (design base struct with intent of layer customisation), localisation (dev specific intf for paritcular market, translation, gov regulations, business practices, brand mngt, cultural elements, seamless integration)
Effective Design: recognize cultural elements in given app, cultural diversity makes it unrealistic for designers to rely on intuition/xp for intf design, adaptation of shared intf requires identification of user factors, incl objective (gender, age, ethnic background, mother-tongue), subjective (cognitive style)
Approaches: adoption of user centred dev approach (users and devs work together on identification of factors affecting usability and perf), effective use of iterative & parallel prorotyping (facilitate user participation and maximise effectiveness of intf eval process while minimise time req for dev), integration of Taguchi techniques (to provide rigour for identification of optimum intf, minimise variation as main means for inproving quality), systematic and logical integration of techniques (so method can be applied by commercial intf designers)
Cultural Factors: power distance (degree of dependence between boss and subordinate), collectivism-individualism (integration into cohesive groups or being expected to look after self), femininity-masculinity (extent gender roles distinct or overlap), uncertainty avoidance (extent feel threatened)
Cultural Model: help identify levels of issues being involved in cmplx prob by using international var (categories that org cultural data) or dimensions of culture, meta models help understand how and where culture comes to influence lives in profound way
Iceberg Model: only 10 percent of cultural characteristics of target audience is easily visible to observer, surface (number, currency, time, date format), unspoken rules (obscured, need context of situation), unconscious rules (difficult to study)
Multicultural Intf Design: charset (alphabet), collating seq (character sorting order for list), format (number, date, time, currency), layout (address, tel no), icons (red cross not recognisable as hospital/health), symbols (x means not wanted, but sometimes used to fill box), colours, screen text, menu accelerators (on translation maybe two commands start with same letter)
Cultural Design Considerations: identify lang & country, address gov regulations, support charsets, international db design, code form so data not corrupted, display info in culturally correcy ways, use appropriate currency, design graphical img with care, provide alts for natural lang proc and audio/video elements
Single App Localisation: handle multi lang within same app, consistent but localised UI
Avoid: hard code text, refer to culture specific std, use slang, jargon, humor, sarcasm, colloquialism, metaphors, form plurals by adding (s) (use both forms if needed)
User Customisation: keep sentences short and simple, allow users to select date and time format, calendar format, numeric and monetary format, paper size
Usability Testing: goals & qs focus on how well users perform tasks with prod, comparison of prod or proto common, focus on time to complete task & type of errors, testing central, recording perf of typical users doing typical tasks, controlled settings, users observed and timed, data record on vid, log key press, data used to calc perf time identify & explain errors, user satisfaction eval using qs and interviews, field observations may be used to provide contextual understanding
UT vs Research: UT: improve prod, few participants, results inform design, usually not completely replicable, cond controlled as much as possible, procedure planned, results reported to devs; Research: discover knowledge, many participants, results validated statistically, must be replicable, strongly controlled cond, experimental design, sci report to sci community
Testing Conditions: usability lab or other controlled space, emphasis on selecting representative users and developing representative tasks, tasks usually around 30 min, same test conditions for all, informed consent form explains procedures and deals with ethical issues
Types of Data: time to complete task (also after time away from prod), n & type of errors per task, n of errors per time, n of times online help/manuals accessed, n of users making error, n of users successfully completing task
N of Participants: dep on sched for test, availability of participants, cost of running tests, typically 5 to 10, some argue to continue until no new insights
Experimental Designs (Participants): diff (single group alloc randomly to experimental cond, no order eff but indiv diff problem), same (all appear in both cond, no indiv diff, but need to counter balance order eff), matched (in pairs, based on expertise, gender, no order effect and indiv diff reduced, but cannot be sure of perfect match)
Field Studies: natural settings, aim to understand what users do naturally and how tech impacts them
Field Study Uses: identify opportunities for new tech, determine design req, decide how best to introduce new tech, eval tech in use
Data Presentation: aim to show how the prod being appropriated and integrated into surroundings, ex: vignettes (brief evocative description), excerpts, critical incidents, patterns, narratives
Gathering Data Key Issues: setting goals (how to analyze), identify participants (who to get data from), relationship with participants (clear & professional, informed consent when appropriate), triangulation (look at data from more than 1 perspective, collect more than one type of data), pilot studies (small trial of main)
Data Recording: notes, audio, video, photo can be used individually or in combination (notes + photo, audio + photo, video)
Interviews: unstructured (no script, rich but unreplicable), structured (tightly scripted, like qnaire), semi-structured (guided by script bit interesting issues can be explored in more depth, can provide balance between richness and replicability), focus groups (group interview)
Interview Qs: closed (predetermined answer format, easier to analyze, may be analyzed by computer) vs open (no predetermined format), avoid long qs, compound sentences (split them), jargon & lang that interviewee may not understand, leading qs that make assumptions (why do you like), unconscious bias (gender stereotype)
Running Interview: intro (introduce self, explain goals, assure about ethical, ask to record, give informed consent form), warmup (first qs easy and non threatening), main body (present qs in logical order), cooloff (include few easy qs to defuse tension), closure (thank, signal end, turn recorder off)
Enriching Interview Proc: props (devices for prompting interviewee, use proto, scenario)
Qnaires: can be administered to large populations, disseminated by paper, email, web
Qnaire Design: impact of q can be influenced by order, may need diff ver of qnaire for diff pops, provide clear instruction on how to complete, strike balance between using white space and keeping qnaire compact, avoid very long qnaires, decide whether phrases will be all positive, negative or mixed
Q Format: yes/no, checkboxes with many opts, rating scales, open ended responses
Encouraging good response: make sure purpose of study is clear, promise anonymity, ensure qnaire well designed, offer short version, if mailed include stamped addressed envelope, follow up with emails, phone calls, letters, provide incentive, 40% response good, 20% acceptable
Online Qnaire Pros: realtively easy & quick to distribute, responses usually received quickly, no copying and postage costs, data can be collected in db for analysis, time required for data analysis is reduced, errors can be corrected easily
Online Qnaire Cons: sampling problematic if pop size unknown, preventing individuals from responding more than once can be a problem, also known to change questions in email qnaires, server privacy laws maybe different
Observation: direct observation in field (structuring frameworks, degree of participation (insider ot outsider), ethnography), direct observation in controlled environment, indirect observation (tracking users' activities, diaries, interaction logging, video & photo collected remotely by drones or other equipment)
Observation Framework (Robson, 2014): space (what physical space is like, layout), actors, activities, objects, acts, events, time, goals, feelings
Planning & Conducting Observation in Field: decide on involvement (passive oberver to active participant), how to gain acceptance, how to handle sensitive topics (culture, private spaces), how to collect data (what data, what equipment, when to stop)
Ethnography: philosophy with set of techniques that include participant observation and interviews, ethnographers immerse themselves in culture they study, degree of participation can vary from inside to outside, alanyzing vid & data logs can be time consuming, collection of comments, incidents, artefacts made, coop of ppl beting observed is req, informants useful, data analysis continuous, qs get refined as understanding grows, reports usually contain examples
Online Ethnography: virtual worlds have persistence that physical worlds do not have, ethical considerations & presentation of results different
Ethnography Observations & Materials: activity/job descriptions, rules & procedures that givern particular activities, descriptions of activities observed, recordings of talk between parties, informal interviews with participants explaining detail of observed activities, diagrams of physical layout, photos/videos/descriptions of artefacts, workflow diagrams showing seq order of tasks, proc maps showing conn between acts
Observation in Controlled Env: direct (think aloud), indirect (track user activities, diaries, interaction log, web analytics), both types use video, audio, photo, notes
Web Analytics: sys of tools and techniques for optimizing web usage by measuring, collecting, analsing, reporting web data, tytpically focus on n of web visitors and page views
Choosing Techniques: dep on study focus, participants involved, nature of techniques, resources & time available
Quantitative & Qualitative: quanti as numbers, quali hard to measure sensibly as numbers (count n of words to measure dissatisfaction), quanti analysis use numerical methods (mean, median, mode, percentage, graphs, charts) to ascertain size, magnitude, amount, quali analysis expresses nature of elements, represented as themes, patterns, stories, critical incidents, categorize data
Tools for support of Data Analysis: spreadsheet, statistical packages, quali data analysis (categorization and theme based, analysis of text based)
Qualitative Analysis Framework: below **Grounded Theory**: derive theory from systematic analysis, based on categorization (open: identify categories, axial: flesh out & link to subcategories, selective: form theoretical scheme), researchers encouraged to draw on theorethical backgrounds to perform analysis
Distributed Cognition: ppl, environment, artifascts regarded as one cognitive sys, used for analysing collaborative work, focus on info propagation & transformation
Resource Info Structs: plans (action to be performed), goals, current state (of world/interactive sys), history (actions, properties), action-effect model (effect actions have on sys), affordances
Resource Configuration: collection of info structs that can be defined in each step of interaction and which can be used to inform action, can be enternal in intf or represented in user's mind
Interaction Strategies: link resource config to support decision making on actions, resource alloc as part of UI design and identification to support evaluation, plan following (plan, history, current state), plan construction (goal, affordances, action-effect and current state), goal matching (goal, affordances, action-effect), history-based choice (goal, affordances, history)
Activity Theory: explains human behaviour in terms of practical activity in world, provides framework that focuses analysis around concept of activity and help to identify tension between diff elements of sys
Presenting Findings: only make claims data can support, best way to present findings dep on audience, purpose, data gathering and analysis undertaken, graphical representations may be appropriate, rigorous notations (UML), using stories to create scenarios, summarizing findings
Evaluation: why (check user req, that they can use prod, if they like it), what (conceptual model, early proto, later more complete proto), where (natural & lab settings), when (throughout design, finished prod can be eval to collect info to inform new prod)
Eval Types: controlled settings involving users (usability test & experiments in labs & living labs), natural settings involving users (field studies & in the wild studies to see how product used in real world), settings not involving users (predict, analyze, model aspects of intf analytics)
Living Labs: can be used to eval people's use of tech in everyday life (too hard to do in usability lab)
Usability Testing + Field Studies: FS to eval initial design ideas and get early feedback, make design changes, UT to check specific design features, FS to see what happens when used in natural env, make final design changes
Crowdsourcing: if large number of participants needed, let ppl perform online tasks for small reward, similar results to lab exp and lower cost
Eval Methods: observing (controlled & natural settings), ask users (controlled & natural settings), ask experts (natural & no user settings), testing (controlled setting), modeling (without users)
Participant Rights and Consent: participants need to be told why eval being done, what they will be asked to do, rights (know goals, know what will happen to findings, privacy of personal info, leave when they wish, be treated politely), informed consent form have them, design of form, eval proc, data analysis & data storage methods are typically approved by authority (both institution and external, for specific groups of participants possibly, such as children)
Things to Consider When Interpreting Data: Reliability (does method produce same result on diff occassion), Validity (does method measure what it is intended to measure), Ecological Validity (does env of eval distort results), Biases (may distort results), Scope (how generalizable are the results)
DECIDE Framework: for eval
Determine Goals: high level goals, who wants it and why, goals influence methods used, goal ex: identify best metaphor, check user req met, check consistency, investigate how tech affects working preactices, improve usability of existing prod
Explore Qs: qs help guide eval, goal can be broken down to qs, ex: paper vs eticket, what are cust attitude to eticket, are they conerned about security, if intf for obtaining them poor
Choose eval approach & methods: eval method influences how data collected, analysed, presented
Identify practical issues: how to select users, find evaluators, select equipment, stay on budget, stay on schedule
Decide about ethical issues: develop informed consent form
Evaluate, Interpret & Present Data: consider same stuff as when interpreting above
Inspections: experts use knowledge of users & tech to review SW usability, expert critiques can be formal/informal, walkthroughs involve stepping through pre planned scenario noting potential problems
Heur Eval: review guided by heurs, design guideleines form basis for developing heurs
Nielsen's Heurs: visibility of sys status, match between sys & real world, user control & freedom, consistency & std, error prevention, recognition rather than recall, flexibility & efficiency of use, aesthetic & minimalist design, help users recognize, diag, recover from error, help & doc
N of Evaluators: Nielsen suggests on avg 5 evaluators identify 75 to 80% of usability prob, n of users needed for same lvl dep on context and nature of prob (Cockton, Woolrych, 2001)
Web Heurs (Budd, 2007): clarity, minimize unnecessary complexity & cognitive load, provide users with context, promote positivs & pleasurable UX
Ambient Disp Heurs: visibility of state, peripherality of disp
Heur Eval Stages: briefing to tell experts what to do, eval period (1 to 2h, each expert work separately, 1 pass to get feel for prod, next pass for foxus on specific features), debriefing (experts work together to prioritize prob)
Heur Eval Pros, Cons: few ethical and practical issues (no users), can be difficult and expensive to find experts, best experts have knowledge of app dom & users, important prob may get missed, trivial prob often identified, experts have biases
Cognitive Walkthrough: focus on ease of learning, designer presents aspect of design & usage scenarios, expert is told assumptions about user pop, context of use, task details, one or more expents walk through design proto with the scenario, guided by 3q (will correct action be suffifiently evident to user, will user notice that correct action is available, will user associate & interpret response from action correctly)
Pluralistic Walkthrough: variation on cognitive walkthrough theme, performed by carefully managed team, panel of experts begins by working separately, then there is managed discussion that elads to agreed decisions, approach lends itself well to participatory design
Eval Using Analytics: eval user traffic through sys or part of sys, times of day & visitor IP addr
Predictive Models: provide way of eval prod/designs without directly involving users, less expensive than user testing, usefulness limited to sys with predictable tasks (tel answer sys, mobiles, cell & smartphones), based on expert error-free behaviour
Fitts' Law: predicts that time to point at an obj using device is function of dist from target obj & obj size, further away & smaller obj, longer time to locate and point to it
Human Factor: humans important as part os sys dependability & successful use, operator errors are largest single source of failures in many sys, human interact using Ui with sys, human error inevitable
Human Element: one aproach to sys is to amke them simple (reduced complexity aids understanding, not possible with addition of humans as highly complex entities), humans as components (often unreliable, unpredictable/nondeterministic, vary enormously, sex, age physical and cognitive aspects, cultural, issues with sys can sometimes be dirrectly attirbuted to human error, developer vs user distinction)
Alloc of Func: computers good at speed and following predefined set of instructions, ppl good at judgment calls, flexible & adaptable
Sys Complexity: having humans in system loop adds complexity and shift responsibility (sys integrity requirements), example: train driver control train speed, respond to signal and env cond, supported by some auto features (dead man's handle for auto brake), removing driver then dep on computer based sol (consequence of failure high, wider implications for connecting sys — cost), many sys built on human strength (would like to build intf that build on them, but then can become weakness)
Humans in Sys: operators as part of sys, humans design, build, install, maintain, certify (uses large amount of subjective judgment calls) sys
Human Error: slip (right action but fail to exec), mistake (wrong action), happens despite xp, skill based (error of inattention or misplaced attention, slip)
, rule based (inappropriate rule, misdiagnose state of sys, deficient rules, wrong recall of procedure), knowledge based (incomplete/inaccurate understanding of sys, overconfidence, cognitive strain, misdiagnosis)
Dangers of Automation: skill and rule based usually, leave operator to do knowledge based, but under stress ill suited, can hinder unserstanding and mental modelling (decrease sys visibility, increase cmplx, operator no hands on control xp)
Coping with Human Error: fault tolerant sys, undo (familiar model for recovery, enbale trial and error, can be for system state, must encompass all hard state, incl net & HW config, flexible, now overhead, transparent to end user)
Humans in Eval: verification (output of phase fulfills req of prev phase), validation (spec of phase or complete sys appropriate and consistent with cust req), testing (proc to verify/validate sys or component)
Anxiety: ease of use (natural UI, forgiving sys, user control)
Alienation: socially active designers, comms support sys can indicate emotions (emoticons), multimedia sys incl video & voice
Impotency: when simple task put incontact with comp may feel powerless, frustrated
Complexity & Speed: comp speed up & add cmplx to life, should try to simplify & slow down some
Org & Societal Dep: dependence on sys make it easier to bring down, need robust fault tolerant design
Unemployment & Displacement: responsibility of org, offer retraining & other jobs
Value Human Diversity: flexible, adaptable intf, neutral, non-biased interaction
Ethical Concerns: accessibility, privacy, cyberbully, accountability, accuracy, online predator, intellectual property, user testing (perf anxiety, feel like intelligence test, compare self & compete with others, feel stupid in front of observer)
Formative vs Summative: F used throughout, S at end, F for improvement, S for grades/check of goals met at end