HCI According to ACM: the discipline concerned with design, eval, impl of interactive computer sys for human use & with study of major phenom surrounding them

HCI Definition: study of interaction between people & computer based sys, concerned with phys, physiological & theoretical aspects of this proc, about designing computer sys that support people 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 & computer in direct/indirect manner

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:

Human: human info processing, lang, comms & interaction, ergonomics

Computer: I/O devices, dialogue techniques, dialogue genre, computer graphics, dialogue arch

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 people want to use it, be attractive, engaging, fun)

Principles for supporting HCI + understanding users' needs: take into account what people are good & bad at, consider what might help people with the way they currently do things, think through what might provide quality user xp, listen to what people 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 people 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 people 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

ID in Business: help companies enter age of consumer, design human centered prod & services, from research & prod to goal related design, provides wide range of design services, in each case targeted to address prod dev needs at hand, creates prod, services, environmments for companies pioneering new ways to provide value to customers

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 (people 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 people in real world, way people 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 people want, need desire, appreciate that one size does not fir all, identify incorrect assumptions about particular user groups (not all old people want/need big fonts), be aware of both people's sensitivities & capabilities

Cultural Differences: date format

Accessibility: degree to which prod usable & accessible by as many people 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 people 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: people 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 description of how sys organized & operates, enables designers to straighten out thinking before they start laying out their widgets (Johnson, Henderson, 2002), describe 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 people 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 meethphor, 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 Advantages: novice can learn basic func quickly, experienced users can work extremely 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 Disadvantages: some people 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 people 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: simplification of 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 concentrate 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 structured 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 for Attention: 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 for Perception: 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), people 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)

SenseCam: intermittently takes photos without user intervention while worn, can improve memory from Alzheimer's

Design Implications: don't overload memory with complicated 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), erroneous ex: turn up thermostat to heat up quicker

Gulf of Execution & Evaluation: exec is from user to sys, eval is other way

Info Processing Steps: encoding, comparison, response select, exec

Distributed Cognition (DC): info transformed through different media (computer, display, paper, head) instead of in mind only

DC Involves: distributed problem solving, role of verbal & non 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 in conjunction with external representation to carry out computation (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 people 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 interaction design, stakeholders can see, hold, interact with proto more easily than doc/drawing, team members can comm effectiely, 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, resolution/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), plus and minus scenarios 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