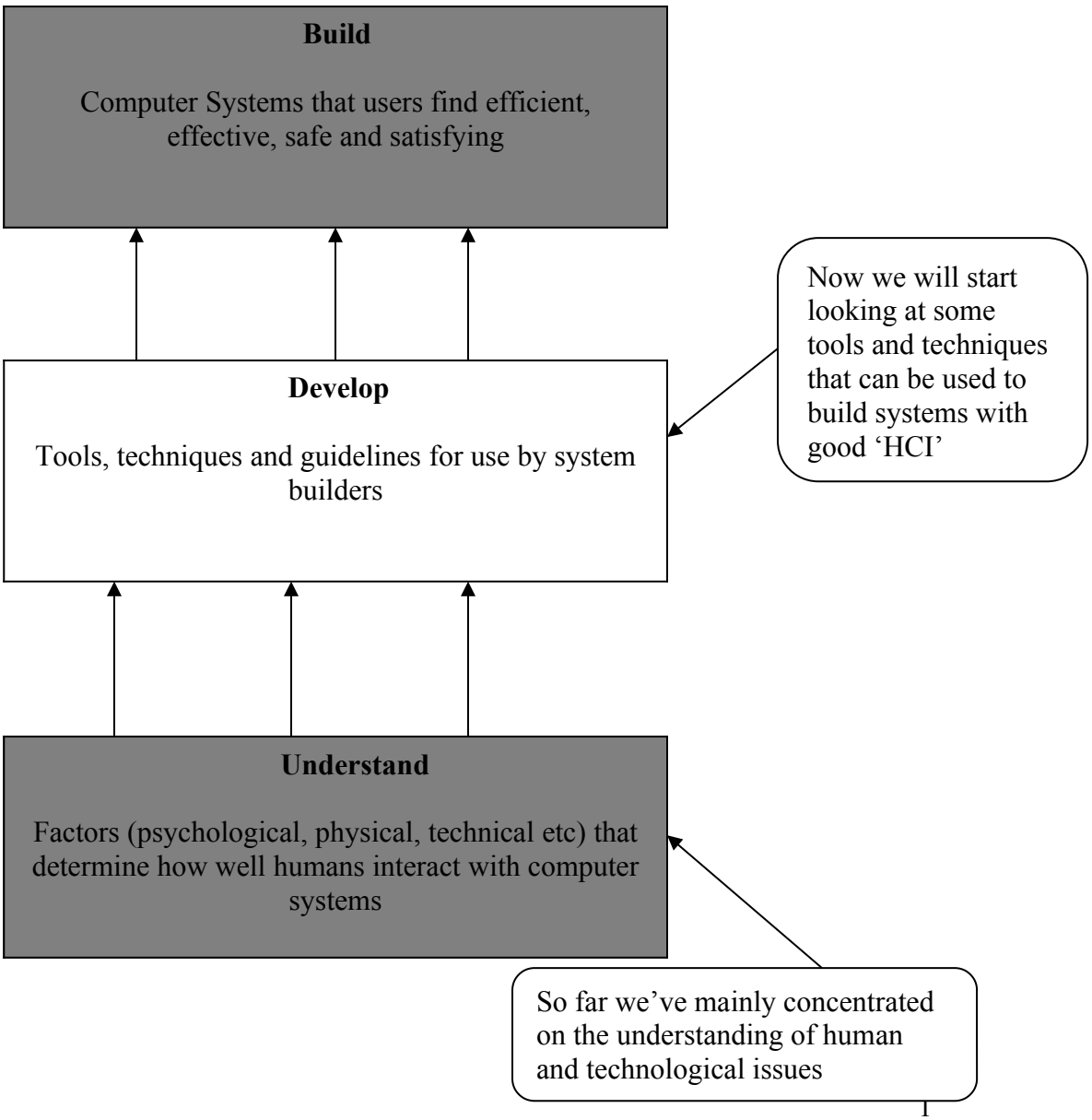
HCI-13-APPRAOCHES TO DESIGN

Aims and Objectives: this is the first of two lectures that look specifically at HCI design techniques and issues. This lecture starts by looking at where the consideration of HCI fits into the process of system development and then goes on to discuss three design techniques that are applicable to HCI design. The three techniques are the *use of standards and guidelines*, an approach called *usability engineering* and the use of *prototyping*. The next lecture looks in more detail at a technique called Hierarchical Task Analysis (HTA) and demonstrates how it can be used.

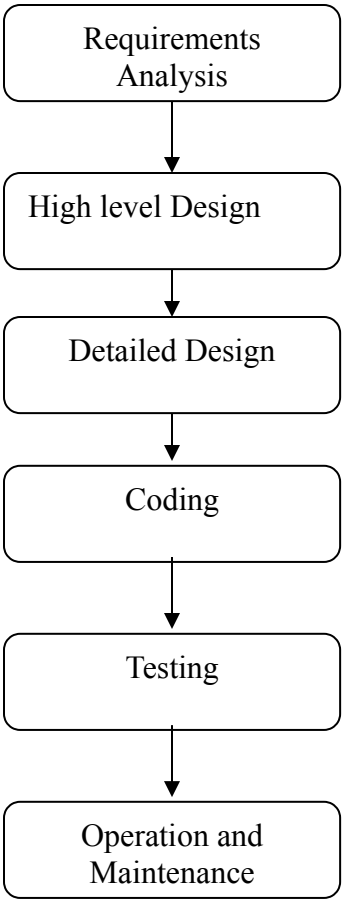
Lecture Notes

Introduction

Remember the Goals of HCI that we looked at in the first HCI lecture.



Where should HCI design fit into the software development lifecycle?



When trying to find a particular phase to fit HCI design into they all seem too early or too late or both e.g.

- Requirements analysis – good place to start thinking about usability but **too early** for details of screen designs
- Detailed design- possibly where detailed screen designs will be finalised but too late if the overall design of the system compromises its usability.

‘’Traditional’’ software development (esp. 70’s and early 80’s) gave little explicit consideration to HCI. Often HCI problems don’t surface until system testing or even after delivery of the system to the users.

The probably unsurprising conclusion is:

- That HCI cannot fit wholly within any one lifecycle stage, and
- Good HCI cannot be achieved with a single technique.

Ideally HCI should be considered throughout the development process and approached using a number of different tools, techniques and guidelines.

We will briefly consider three approaches (not mutually exclusive) in this lecture.

- Guidelines and Standards
- Usability engineering
- Prototyping

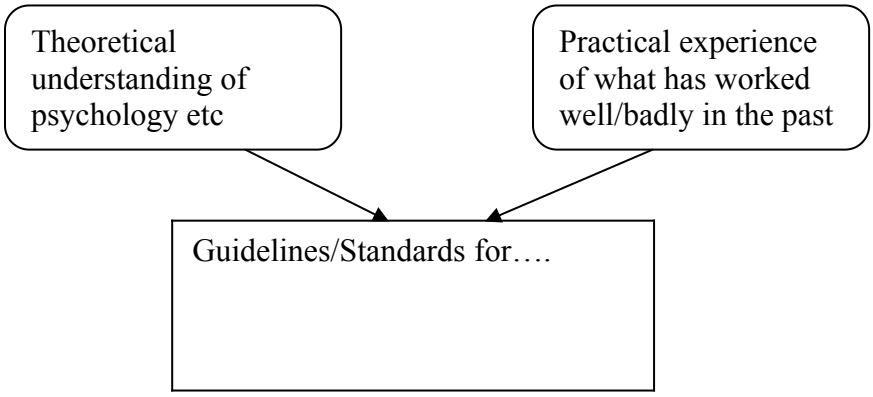
The next lecture will look at a technique called Task Analysis in a little more depth.

Design technique 1 – Guidelines and Standards

Often these terms are used interchangeably however they can be distinguished.

Guidelines	:	tend to be hints and advice that can be overridden
Standards	:	tend to be more rigid rules which should be obeyed. Some standards have the force of law.

Guidelines and Standards are derived from two sources.



Guidelines and Standards vary in a number of ways e.g.

- Generality
- Areas covered
- Originator

Generality

On the whole standards tend to be more specific than guidelines. They can range from very general e.g.

- The content of displays will be presented in a consistent manner
- The sequence of tasks a user is expected to perform should follow a logical sequence to more specific e.g.
  - Users will always be given a method of undoing their last action
  - Use audible cues to call the user's attention to urgent messages

to very specific e.g.

- all screen error messages will be of no more than 10 words in length.
- The company name in blue text must appear at the top left of all web pages
- All menus will contain 2 to 9 options

Areas covered

Some guidelines and standards are specific to certain areas e.g. I/O devices, Screen layouts, Documentation (Online help, Manuals, Error messages etc), web pages etc.

Originator

Guidelines and standards are developed by many different bodies. It can sometimes be bewildering for a developer to know which to apply in a given situation.

Many organisations, especially large ones, have their own **in-house** guidelines and standards. These are often used to give a consistent '' house'' style to software produced by a company. Software companies and other large developers (e.g. banks) may have whole quality and standards departments who develop, maintain and maybe monitor the application of guidelines and standards.

Another category of guidelines and standards are what may be called **platform based**. Companies such as Microsoft, Apple, and Lotus produce standards for software developed for their platforms or using their products. Try visiting **htt://devworld.apple.com/dev/techsupport/** using their products

Some guidelines and standards can be considered as **public domain**. For example researchers and authors may publish guidance on the web.

Professional bodies (e.g. the British Computer Society in the UK) may develop guidelines and authors and standards for use by their members and other professionals.

The final category we shall consider are standards developed by **national and international standards** bodies e.g. ISO 9241 covers the ergonomic requirements for work Visual display Terminals – both hardware and software.

What are the advantages of using Guidelines and Standards?

- They can be way of passing on experience and knowledge. This can prevent the need for people to continually ‘’reinvent the wheel’’.
- They are good for encouraging consistency
- They can be particularly helpful for inexperienced staff.
- Adherence to guidelines and standards can be checked by reviews at appropriate points in the development lifecycle.

What are the possible problems of guidelines and Standards?

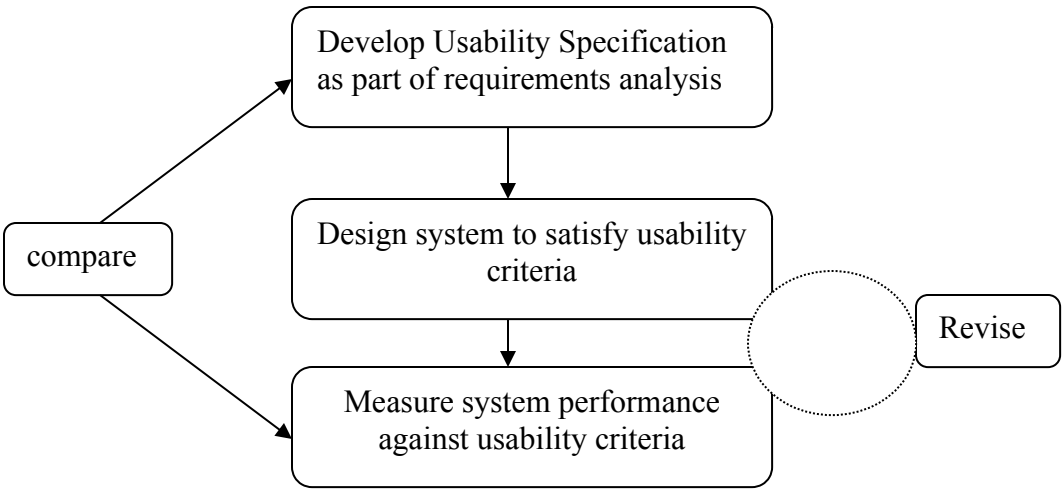
- Finding the appropriate one(s) to apply in a particular situation can be a real problem. Consider the following scenario: - professional body guidelines, results of research found on the web, my company’s guidelines, international standards, national laws, my customer’s company’s guidelines!
- When you try to combine several sets of guidelines and standards you may find that there are conflicts.
    - within a single document (this suggest bad design of the guidelines and standards)
    - between guidelines/standards being applied e.g.
        - your in-house guideline say that the company log must appear on each screen
        - the guidelines for the platform you are designing for says do not use a graphic on screen unless it has an essential functional purpose.
- Guidelines and standards may be constricting to originally
- May quickly become out of date. For example some specific guidelines for web page design may quickly become out of date as the technology changes.

Quick Quiz Question

Standards and guidelines originate from a number of different sources. Identify five types of originators

Design technique 2 – Usability Engineering

Usability Engineering is a formalised approach (hence the term ‘’engineering’’) originated in the late ‘80s within large manufacturers (e.g. IBM and DEC). The basic approach is illustrated below.



Early on in the development process (ideally during the requirement analysis phase) a set of usability criteria is developed. The characteristic of these criteria is that they are measurable. Later on in development process the system will be measured against the criteria. If it is found to be lacking in some way it will need to be amended until it meets the criteria.

The Usability Specification

The usability specification contains a number (possibly large) of measurable attributes. For each attribute **six** items are identified related to its measurement. The six items are:

Attribute name - a brief name for the attribute

Measuring concept – a brief description

Measuring method – the way in which this attribute of the system can be measured. This is crucial as the whole point of usability attributes is that they can be measured.

Now level - the current measurement for this attribute. This maybe the measurement for the current version of a system that is to be replaced or perhaps for a manual system. If the system is usually new it may not be possible to provide a meaningful value for this.

Worst case – the worst possible measurement that would be acceptable
Planned case – the best measurement that can be imagined

Best case – the best measurement that can be imagined

Below are two example entries from a Usability Specification

Attribute Name	Installability/Adaptability,Reliability,Recoverability
Measuring Concept	Ease of installation
Measuring Method	Elapsed time taken to install successfully
Now Level	Current version takes 90 minutes
Worst case	90 minutes
Planned case	60 minutes
Best case	45 minutes

Attribute Name	Preference over XXX
Measuring concept	User satisfaction with our package compared with package XXX
Measuring method	Ratio of user questionnaire scores for the two packages
Now level	Package XXX 10% preferred over current version of our package
Worst case	Two packages level
Planned level	Our package 20% preferred over package XXX
Best case	Our package 50% preferred over package XXX

Some other possible measuring methods for different attributes are:

- Number of errors made
- Number of times the help is consulted
- Number of keystrokes required to perform a task
- Reported level of user satisfaction

Advantage of Usability Engineering

- It promotes early consideration of usability. This is clearly better than the situation where usability is not considered until late in the lifecycle.
- It provides measurable criteria rather than pious hopes.

Possible Problems of Usability Engineering

- It may be difficult to come up with realistic detailed criteria so early in the lifecycle. This is easier where a new release of an existing system is planned but less easy for a completely new system.

- There is a danger that developers may work towards meeting the criteria at the expense of other aspects of the system. If the criteria are too narrowly focused this may result in a system that meets the criteria but still faults.

- If the built system fails to meet the criteria then reworking it may be expensive.

- The technique only provides detailed goals but little guidance about how to achieve them. This suggests that perhaps this technique should be used in conjunction with other techniques that offer more help and guidance (e.g. guidelines and standards).

Quick Quiz Question 2

What do you think accounts for the term 'engineering; in the title Usability Engineering?

Design Technique 3 – Prototyping

Prototyping involves building a **working model** of part, or some aspect of, the system for evaluation before building the full system. It is a similar approach to that followed by engineers – e.g. when a new car design is being developed.

Figure: Formula 1 Car

When prototyping for HCI it is the system interface that gets prototyping

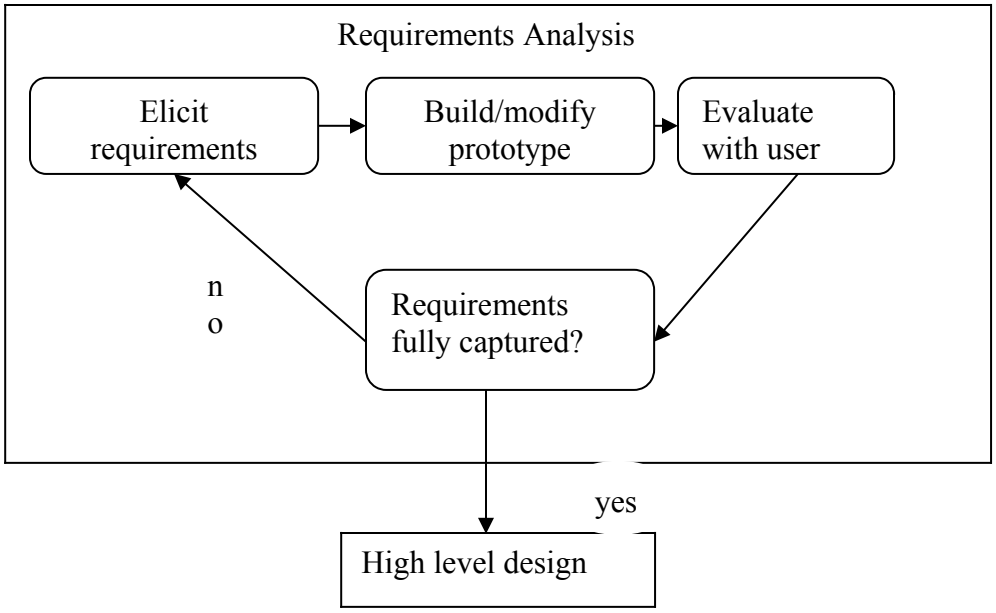
There are two main types of prototyping

- **Throw-away** - the prototyping is done within the requirements analysis phase to elicit user requirements. Then system development proceeds as normal.

- **Evolutionary** - the final system gradually evolves from a series of prototypes.

These two techniques are described in more detail below.

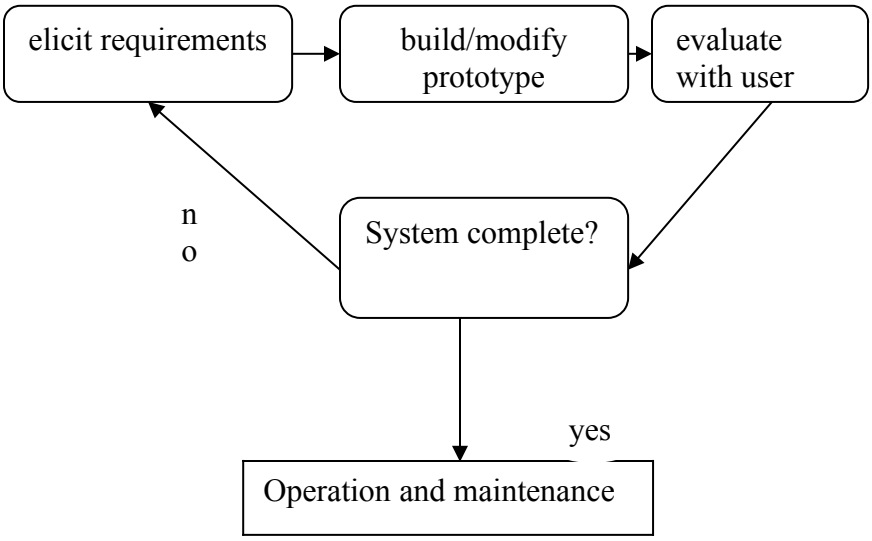
Throw-away Prototyping



The whole process takes place within the requirements analysis phase of software development lifecycle. The purpose is to use the prototype to capture the user's requirements. An initial prototype is developed based on a first understanding of the user's requirements. The prototype is then evaluated in conjunction with the user. If the prototype completely captures the user's requirements (unlikely on the first attempt) the prototyping process is finished and development moves on to the next phase. The whole iterate process continues until the requirements are completely captured.

This process has a less radical effect on the lifestyle than evolutionary prototyping. Prototyping may vary from primitive animations showing screen layouts (e.g. using PowerPoint slides) to simulated system functionality (e.g. using visual basic) The final system may be delivered using a completely different tool (e.g. Java or visual C++).

Evolutionary Prototyping



This approach revolutionises the traditional software development lifestyle. It completely replaces the traditional phases with the iterative prototyping process. For this type of prototyping, unlike throwaway prototyping, the prototype development tool (e.g. programming language) used must be suitable for the development of the fully operational system

Advantages of Prototyping

- Users are often unable to define their requirements fully without being able to experience how the system will work.
- It involves users early (and possibly throughout) the development process.
- It aids communication between users and developers and within the development team.
- Possible HCI problems can be corrected early when it is relatively easy and cheap.

Possible Problems of Prototyping

- *Management can be difficult* – how many versions of the prototype will be required, how long will the whole process take?
- *Can be time consuming*. It is possible to get into a situation where the system never seems to be completed. It always needs ‘’ one more change’’ to perfect it or meet some new requirement.

- The choice of users to be involved is important. You don't want to design a system that just suits one person.
- Sometimes systems developed as a result of evolutionary prototyping may be poorly documented and difficult to maintain. With the traditional approach to software development a written specification is produced to be fed into the next stage. This may not happen with evolutionary prototyping.

Quick Quiz Question 3

What do you think are the main criteria for choosing a tool to use for throwaway prototyping? How does this differ from evolutionary prototyping?

Summary:

This lecture has examined at the place of HCI design in the system development and has described three possible approaches, namely the use of guidelines and standards, usability engineering and prototyping. Some conclusions are that.

- There is no simple way to build systems that can be guaranteed to exhibit good HCI i.e. systems that users find efficient, effective, safe and satisfying to use.
- HCI needs to be considered throughout the software development process.
- There are a number of technique available to help developers with the HCI aspects of the systems

On completing this lecture you should be able to:

- explain were HCI design fits into the system development process
- describe three approaches to HCI design and discuss their strengths and weakness

Another design technique (task analysis) will be covered in the next lecture.

Tutorial Exercise

1. Search the web to find several sets of guidelines about designing web pages.

   - Can you find any that seem out of date?
   - Can you find any conflicts between different sets of guidelines?
   - How can you tell which guidelines are worth adhering to?

2. Imagine that you assigned to project to develop a voice input facility for a popular spreadsheet package. Identify three measurable attributes that you might include in the usability specification. Draw up the usability specification entries for each of them.

Answers to the Quick Quiz questions

1. five types of originator are:

   - in-house
   - platform specific
   - public domain
   - professional bodies
   - standards bodies

2. The ‘’ engineering’’ signifies that this is a formalised approach that is based on the use of measurable criteria.

3. For throwaway prototyping the main criteria is that the tool should be quick and easy to use for creating and amending the prototype. For evolutionary prototyping there is the additional requirement that the tool needs to be suitable for building the full system, not just the user interface.