UNIT 4: THE SOFTWARE DESIGN PROCESS OF HUMAN COMPUTER INTERACTION

Table of contents

1.0 INTRODUCTION

Software engineering is the discipline for understanding the software design process, or life cycle. Therefore, this unit looks at the software design process of human computer interaction by analysing the pros (linearity) and cons (non linearity) of the water fall model that comprises the design life cycle. The usability engineering process that measures the user's experiences is weighed against the ISO usability standards 9241. For a successful and effective design, management issues concerned with interactive design and prototyping are considered along the relevant design rationale.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Understand the distinct activities that constitute the software engineering life cycle
- Understand the concepts of the water fall model in relationship to the software life cycle
- Differentiate between software engineering life cycle usability engineering
- Explain the concepts of interactive design and prototyping

3.0 MAIN CONTENT

3.1 The software process of Human Computer Interaction

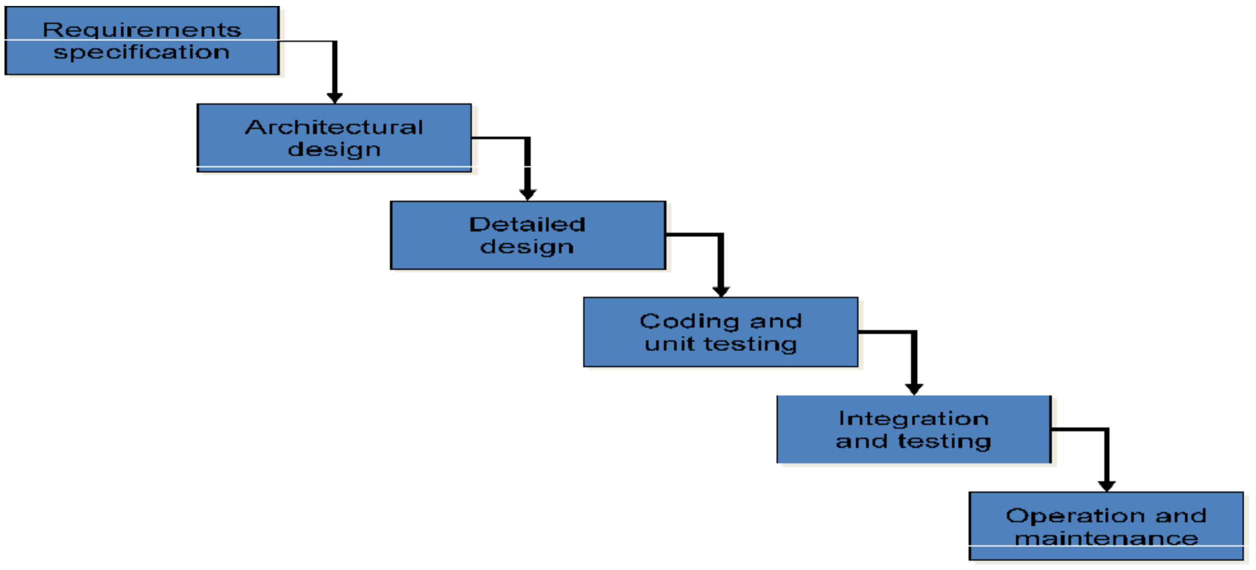
The software process comprises the following:

- Software engineering and the design process for interactive systems
- Usability engineering
- Iterative design and prototyping
- Recording the design knowledge using the design rationale

Software engineering is the discipline for understanding the software design process, or life cycle. The design for usability occurs at all stages of the life cycle, not as a single isolated activity.
Usability engineering is the ultimate test of usability based on measurement of user experience.
Iterative design and prototyping overcomes inherent problems of incomplete requirements
Design rationale is information that explains why a computer system is the way it is.

3.2 The waterfall model

The waterfall model depicts the software life cycle.
The pictorial illustration that follows reflects a mountain top from where water f lls towards the bottom of the mountain, hence called a waterf ll. It shows the commencement of the life cycle (the requirements specification) through the design, co ding and testing processes to its ultimate ter mination (the operation and maintenance).



Activities in the software lifecycle are :

Requirements specification:
Here, the designer and client try to capture what the system is expected to provide and can be expressed in natural language or more precise languages, such as a task analysis would provide.

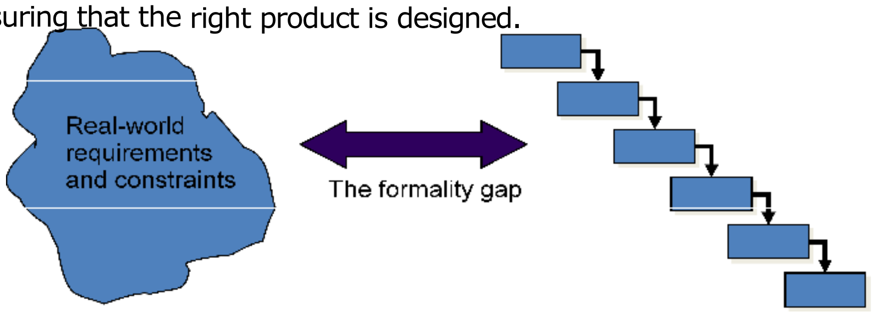
Architectural design:
This is a high-level description of how factor system into major component
the system will provide the services required. It describes how to d. s of the system and how they are interrelate It shows the needs to satisfy both functional and non-functional requirements

Detailed design
This concerns a refinement of architectural implemented separately. The refine

components and their interrelations to identify modules to ent is governed by the non-functional requirements.

Verification and Validation

Verification: This is ensuring that the product is designed right Validation: This is ensuring that the right product is designed.

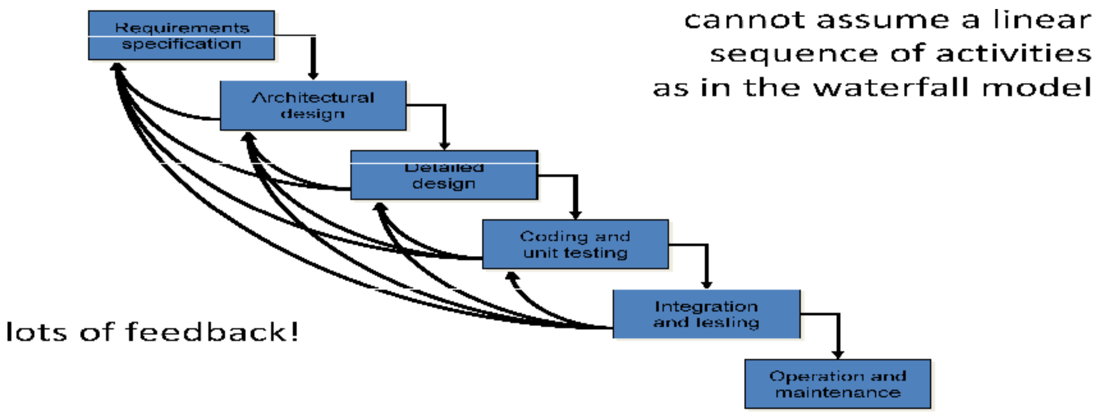


The formality gap shown in the diagram above indicates that validation will always be subjective means of proof.

s rely to some extent on

Management and contractual issues
describes design in commercial and legal co
ntexts

. 3 The life cycle for interactive systems


Requirements specification
Architectural design
Detailed design
Coding and unit testing
Integration and testing
Operation and maintenance
cannot assume a linear sequence of activities as in the waterfall model
lots of feedback!

The life cycle for interactive systems cannot assume a linear sequence of activities as in the waterfall model because there are lots of feed backs occurring within the initial requiremets specification, the designs, the coding and testing processes. See the illustration above.

3.4 Usability engineering:

This is the ultimate test of usability based on measurement of user experience. Usability engineering demands that specific usability measures be made explicit as requirements.

Usability specification comprises the usability attribute and/or principle, the mea suring concept and the measuring method. It also depicts either the present level the worst case level, t e planned level or the best case level.

The problems associated with usability specifications are:

i. Usability specification requiresel of detail that may not be possible in lev process.

ii. It does not necessarily satisfy usabthe e ility.

arly life of the design

Example of a usability specification Attribute: Backward recoverability
Measuring concept: Undo an errone us programming sequence
Measuring method: Number of explicit user actions to undo current program Present level: An undo is not allowed presently
Worst case: This considers as many actions as it takes to program in mistakes Planned level: A maximum of two explicit user actions are allowed
Best case: One explicit cancel action is allowed

3.5 ISO usability standard9241:

ISO usability standard 9241 adopts the following traditional usability categories:Effectiveness: This is achieving what you want to.
Efficiency: This is doing it without wasting effort.
Satisfaction: This is showing whether or not you enjoy the process. Some metrics from ISO

9241 are :

Usability objective	Effectiveness measures	Efficiency measures	Satisfaction measures
Suitability for the task	Percentage of goals achieved	Time to complete a task	Rating scale for satisfaction
Appropriate for trained users	Number of power features used	Relative efficiency compared with an expert user	Rating scale for satisfaction with power features
Learnability	Percentage of functions learned	Time to learn criterion	Rating scale for ease of learning
Error tolerance	Percentage of scale for errors corrected handling successfully	Time spent on correcting errors	Rating error

3.6 Iterative design and Prototyping

Iterative design overcomes inherent problems of incomplete requirements while prototypes simulate or animate some features of intended system.

Different types of prototypes can be identified as:

(i) throw-away,

(ii), incremental and

(iii) evolutionary

Management issues concerned with Interactive design and Prototyping: The management issues are:

- Time allocated to the design,
- Planning the design,
- Non-functional features of the design and
- Contracts of the design.

Prototyping Techniques:
These include the following:

- Storyboards which need not be computer-based but can be animated
- Limited functionality simulations in which some part of the system functionality is provided by designers. Such tools like HyperCard are common.
- Wizard of Oz technique
- Warning about the iterative design. These show concerns about the design inertia in which early
  bad decisions stay bad. It is better to diagnose the real usability problems
  in prototypes; and not just the symptoms.

4.0 CONCLUSION

The life cycle for interactive systems cannot assume a linear sequence of activities as in the waterfall model because there are lots of feedbacks!

5.0 SUMMARY

The software engineering life cycle consists of distinct activities and the consequences for interactive system design. The waterfall model depicts the software life cycle.
Usability engineering makes usability measurements explicit as requirements and is the ultimate test of usability based on measurement of user experience.
Iterative design overcomes inherent problems of incomplete requirements.
Prototyping techniques comprise storyboards, limited functionality simulations or animations of intended system, and Wizards.
Design rationale records design knowledge through the process and structure orientations.

6.0 Tutor Marked Assignment

1. Briefly explain the four tasks of the software process of Human Computer Interaction.

   2 (a) What do you understand by the term "the Waterfall model" in the software design process of
   human computer interaction?
   (b) With the aid of a related diagram, briefly explain the activities depicted by the "waterfall model"
   3. What do you observe as the major departure of the interactive system life cycle from the "waterfallmodel"? What do you think as the reason behind
   this major departure?

4. (a) Explain the value of usability engineering in the Software design process

(b) What are the major problems associated with usability specifications? © Give any three examples of a usability specification

5. What are the usability objectives of the International Standards Organisation (ISO) 9241? Mention itsusability categories

6. (a) Differentiate between the objectives of interactive design and prototyping. (b) What are the management issues concerned with interactive design and prototyping?

7.0 Further Readings / References

- Swinehart, D., *et al.*, "A Structural View of the Cedar Programming Environment." *ACM Transactions on Programming Languages and Systems*, 1986. 8(4): pp. 419-490.
- Swinehart, D.C., *Copilot: A Multiple Process Approach to Interactive Programming Systems.* PhD Thesis, Computer Science Department Stanford University, 1974, SAIL Memo AIM-2 30 and CSD Report STAN-CS-74-412.
- Teitelman, W., "A Display Oriented Programmer's Assistant." *International Journal of Man-Machine Studies*, 1979. 11: pp. 157-187. Also Xerox PARC Technical Report CSL-77- 3, Palo Alto, CA, March 8, 1977.
- Newman, W.M. "A System for Interactive Graphical Programming," in *AFIPS Spring Joint Computer Conference.* 1968. 28. pp. 47-54.
- Shneiderman, B., "Direct Manipulation: A Step Beyond Programming Languages." *IEEE Computer*, 198 3. 16(8): pp. 57-69.
- Smith, D.C., *Pygmalion: A Computer Program to Model and Stimulate Creative Thought.* 1977, Basel, Stuttgart: Birkhauser Verlag. PhD Thesis, Stanford University Computer Science Department, 1975.