COMP3008 Notes

William Findlay

January 11, 2019

Contents

1 Good Design and User Experience (UX)

1.1 What is User Experience?

- how a product **behaves** when **used in the real world**
  - how do people **feel** about it
  - **every** product used by someone has a UX
    - even ketchup bottles
- we **can't design a user experience**
  - we have to design **for** the user experience

1.2 How Do We Design for a Good User Experience?

- we must take into account:
  - **who** are our users?
  - **what** tasks are they trying to accomplish?
  - **where** is the interaction taking place?

1.3 Usability Goals

1. effectiveness
   - effective to use
2. efficiency
   - efficient to use
3. safety
   - safe to use
4. utility
   - does it do what it's supposed to do?
5. learnability
   - is it easy to learn how to use?
6. memorability
   - is it easy to remember how to use it?

1.3.1 Example: Password Scheme

1. effectiveness
   - number of errors permitted
2. efficiency
   - entry time should take 5-10 seconds
3. safety
   - secure from brute-force and shoulder surfing attacks
4. utility
   - password can be used in basic computer applications
5. learnability
   - use should learn how to use the password scheme in 5 minutes or less
6. memorability
   - password recall should take less than 5 seconds

1.4 Understand User's Needs

- consider what people are **good and bad** at
- how do people **currently** do things?
  - how can we improve it?
- listen to what people want?
  - not always easy to find out

  - ask **the right questions**
- tried and tested **user-centered** methods

1.5 Design Principles

1. visibility
    - show the **state** of the system
    - show **possible actions**
2. affordance
    - **suggest** consequences of actions
3. constraints
    - **steer** actions
4. consistency
    - establish **similarities** between analogous parts of the UI
5. feedback
    - **show** consequences of actions

1.5.1 Visibility

- show the current state to the user
- show **all possible options** to the user
- **make it clear** what they need to do

1.5.2 Affordance

- **actions suggested** by the **design** of an object
- **actual** affordance
  - on actual objects
- **perceived** affordance
- how do we represent the actual in the virtual world?
- problems:
  - **false affordance**
    - perceived affordance does not match what the user was intended to do
  - when **simple things** need to be **explained**

1.5.3 Constraints

- **opposite** of **affordance**
- **limit** perceived **potential actions** based on the object's appearance
- **physical** constraints
  - a large peg doesn't fit in a small hole
- **cultural** constraints
  - red is bad
  - green is good
- **logical** constraints
  - what should **probably** happen

1.5.4 Consistency

- for **analogous elements**
  - analogous operation
  - analogous design
- **easily** learn new UIs
- **less consistency** $\implies$ **more burden** on users

1.5.5 Feedback

- **communicate** to the user what has been done
  - successful?
  - unsuccessful?
  - results?
- types
  - auditory
  - visual
  - haptic/tactile
  - combination
- type to choose depends on **context**

2 Design Process

2.1 Involve Users

2.1.1 Authenticity

- users have **domain expertise**
  - what are their common tasks?
  - exceptions?
  - what are the **roles** in the domain?
- we need their **perspective**

2.1.2 Pragmatics

- expectation management
  - realistic expectations
    - no surprises
    - no disappointments
  - training
  - communication
- ownership
  - make users **active stakeholders**

2.2 Degrees of User Involvement

- member of the **design team**
  - full time
  - part time
  - short term
  - long term
- dissemination devices
  - **social media**
  - increase outreach
  - two-way communication
- post-release
- **combination** of these approaches

2.3 User-Centered Approach

- **early focus** on **users** and **their tasks**
  - **study** characteristics
    - cognitive

    - behavioral
    - anthropomorphic
    - attitudinal
- empirical measurements
  - reactions
  - performance
- **iterative design**

2.4 Four Main Activities of Interaction Design

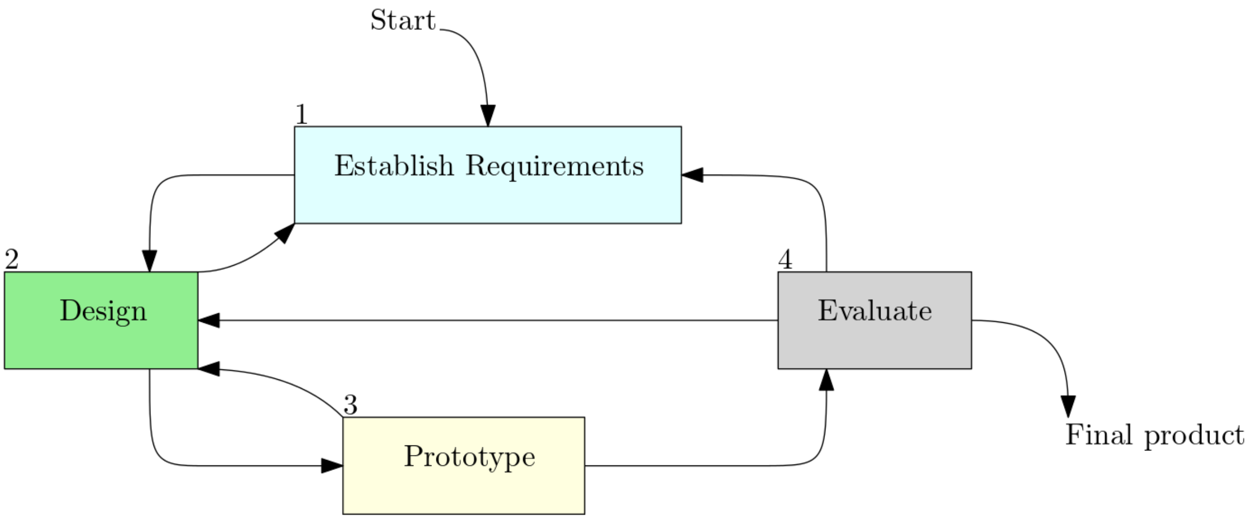

Figure 1: The four activities of interaction design.

2.5 Who are the Users/Stakeholders

- not as obvious as you think
  - there might be other users besides who you initially think
- those who:
  - interact directly with product
  - manage direct users
  - receive output from product
  - make the purchasing decision
  - use competitors' products
- **three categories**
  - primary
    - frequent hands-on
  - secondary
    - occasional
    - or through someone else
  - tertiary
    - affected by the product's introduction
    - will influence the product's purchase
    - e.g., customers at a store where a new cash system is introduced

2.6 User Needs

- users don't always know **what is possible**
- they can't always **tell you** what they need

    - we need to **ask the right questions**
    - we need to **conduct studies**
- look at **existing tasks**
    - context
    - information they require
    - who collaborates?
    - why is the current method used?
        - what might be wrong with it?
- consider **envisioned tasks**
    - what might they want to do?
        - why can't they currently do it?

2.7 ISO 9241-210 Human Centered Design for Interactive Systems

1. The design is based on an **explicit understanding of:**
    - users
    - tasks
    - environments
2. Users are involved throughout
    - design
    - development
3. Design is driven by and influenced by **user-centered** evaluation
4. Design process is **iterative**
5. The design addresses the **whole UX**
6. The design team includes **multi-disciplinary skills and perspectives**

3 Interaction Design vs Traditional Software Engineering

3.1 Integration of ID with SE Models

3.1.1 Royce Waterfall

- not great
- **not iterative**

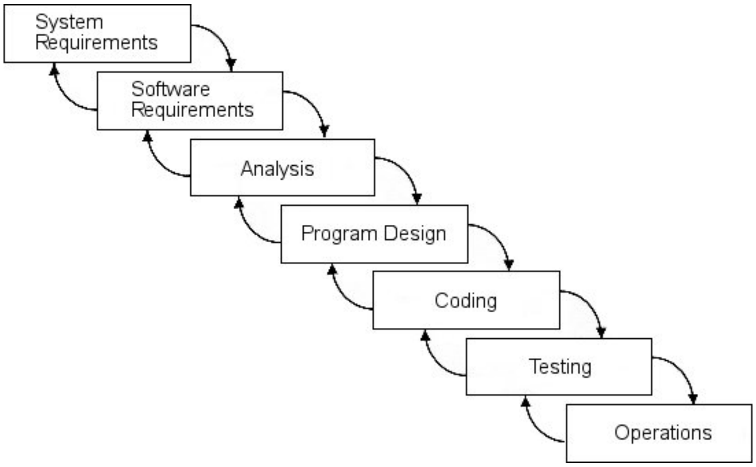

Figure 2: The Royce Waterfall model.

3.1.2 Rational Unified Process

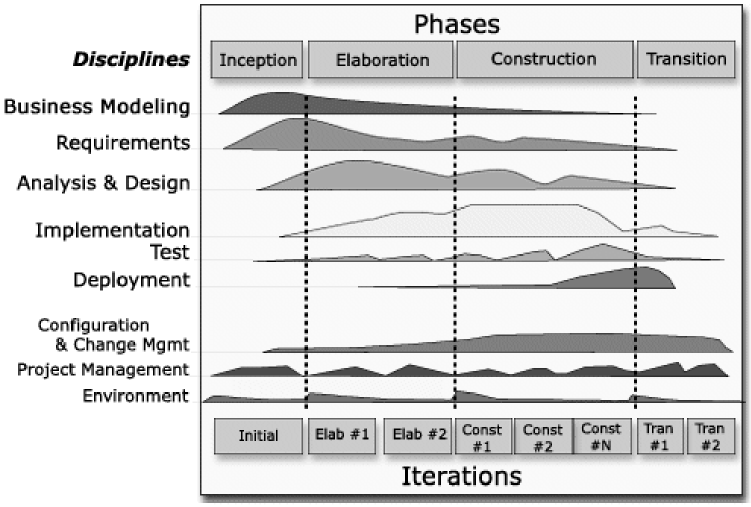

Figure 3: The Rational Unified Process model.

3.1.3 Agile

- this is a great candidate for ID
  - iteration-focused model

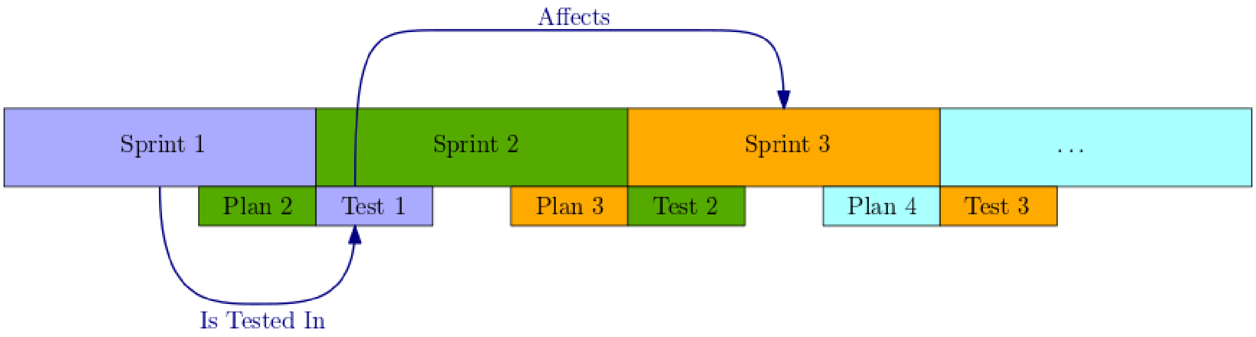

Figure 4: The agile development process. Design 1 is tested in sprint 2. The outcome of this test affects sprint 3. The topic of each sprint is different.

3.2 Integration with Other Models

- **start** with some **early developer involvement**
- **continue** with designer involvement **during implementation**

3.2.1 Early Developer Involvement

- developers can help determine **viable solutions**
  - avoid technical limitations
- help with knowledge transfer
  - relate to client

3.2.2 Continued Designer Involvement

- design reviews
- verify deliverables for good design
- design coach
- limit refactoring as a result of bad UI implementation