SWE3053
Human Computer Interaction

Lecture 23

Interface Design Lifecycle

Projective Augmented Reality

Projecting computer images to reality

- https://www.youtube.com/watch?v=yBJEP4lsRFY

Assignment #9 – Data Analysis

Submit on iCampus before **(Wednesday) May 25 <u>23:59 pm</u>**.

CSE3040 – Human Computer Interaction

1. Theories, Principles and Guidelines
   - Conceptualizing Interaction
   - Sensational and Perceptual Aspect
   - Anthropometric Aspect
   - Cognitive Aspect
   - Emotional and Social Aspect
   - Interface Design Principles and Guidelines

2. Academic Research
   - Research Ethics
   - Research Methodology
   - Gathering Data
   - Data Analysis, Interpretation and Presentation

3. Design and Evaluation
   - Design Lifecycle
   - Establishing Requirements
   - Design, Prototyping and Construction
   - Evaluating Design

Structured Evolutionary Prototyping Model

- Developers build a prototype during the requirements phase
- Prototype is evaluated by end users
- Users give corrective feedback
- Developers further refine the prototype
- When the user is satisfied, the prototype code is brought up to the standards needed for a final product.
-

Structured Evolutionary Prototyping Steps

- A preliminary project plan is developed
- A partial high-level paper model is created
- The model is source for a partial requirements specification
- A prototype is built with basic and critical attributes
- The designer builds
    the database
    user interface
    algorithmic functions
- The designer demonstrates the prototype, the user evaluates for problems and suggests improvements.
- This loop continues until the user is satisfied
-

Structured Evolutionary Prototyping Strengths

- Customers can “see” the system requirements as they are being gathered
- Developers learn from customers
- A more accurate end product
- Unexpected requirements accommodated
- Allows for flexible design and development
- Steady, visible signs of progress produced
- Interaction with the prototype stimulates awareness of additional needed functionality
-

Structured Evolutionary Prototyping Weaknesses

- Tendency to abandon structured program development for “code-and-fix” development
- Bad reputation for “quick-and-dirty” methods
- Overall maintainability may be overlooked
- The customer may want the prototype delivered.
- Process may continue forever
    - Scope creep: continuous and uncontrolled growth in project’s scope

When to use Structured Evolutionary Prototyping

- Requirements are unstable or have to be clarified
- As the requirements clarification stage of a waterfall model
- Develop user interfaces
- Short-lived demonstrations
- New, original development

Iterative Model

- Start with implementing some of the requirements first
- Iteratively enhances further versions releases with additional features and implementation of requirements
- Additional functionalities and capability were added in each iteration
- Design modification can be made in each iteration
- The overall system is developed through repeated cycles (iterations) by building smaller portion at each iteration

Iteration Model

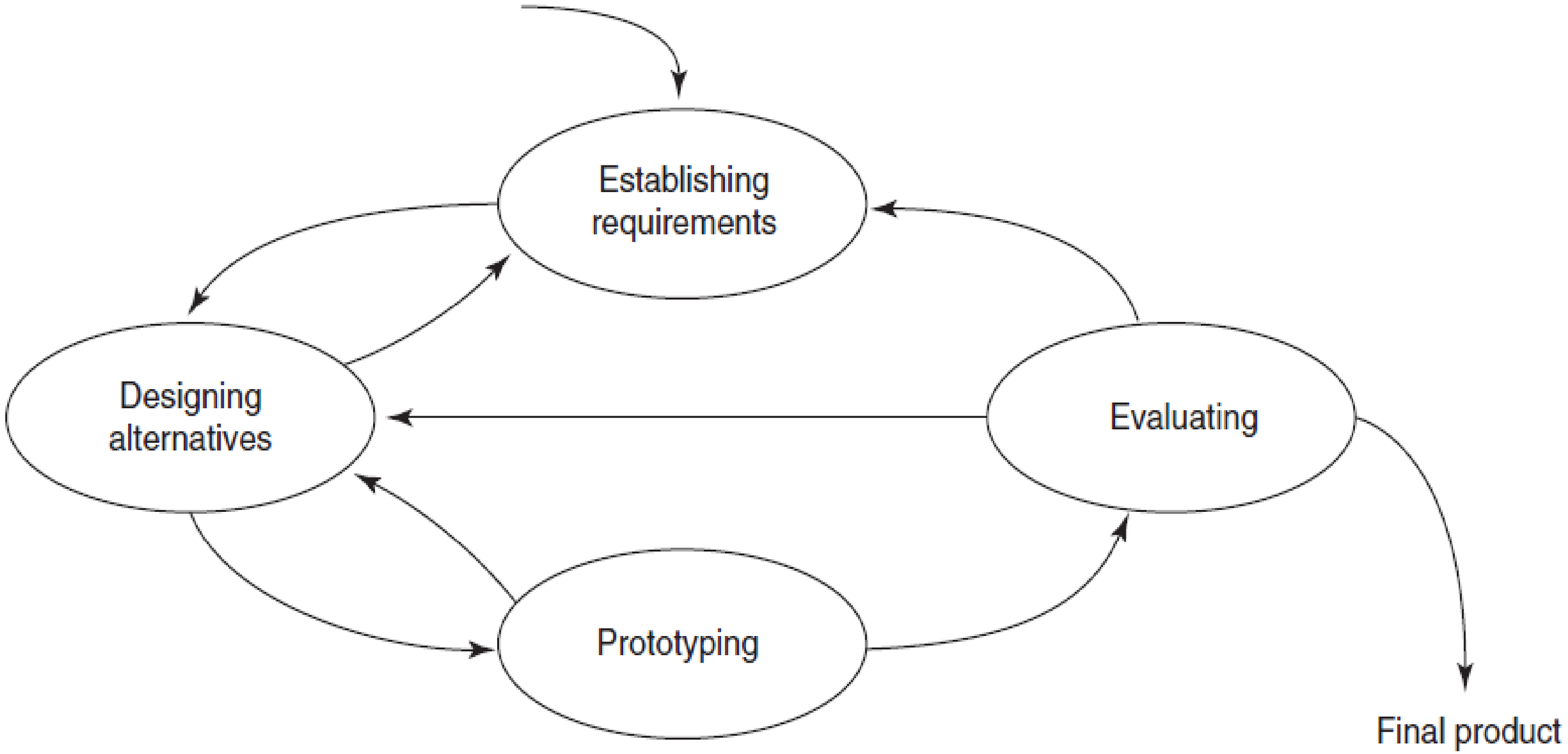


Figure 9.3 A simple interaction design lifecyle model

- Similar to the Waterfall, you just put the Waterfall into a circular model!
- In Waterfall, you're using your customers to evaluate your product!

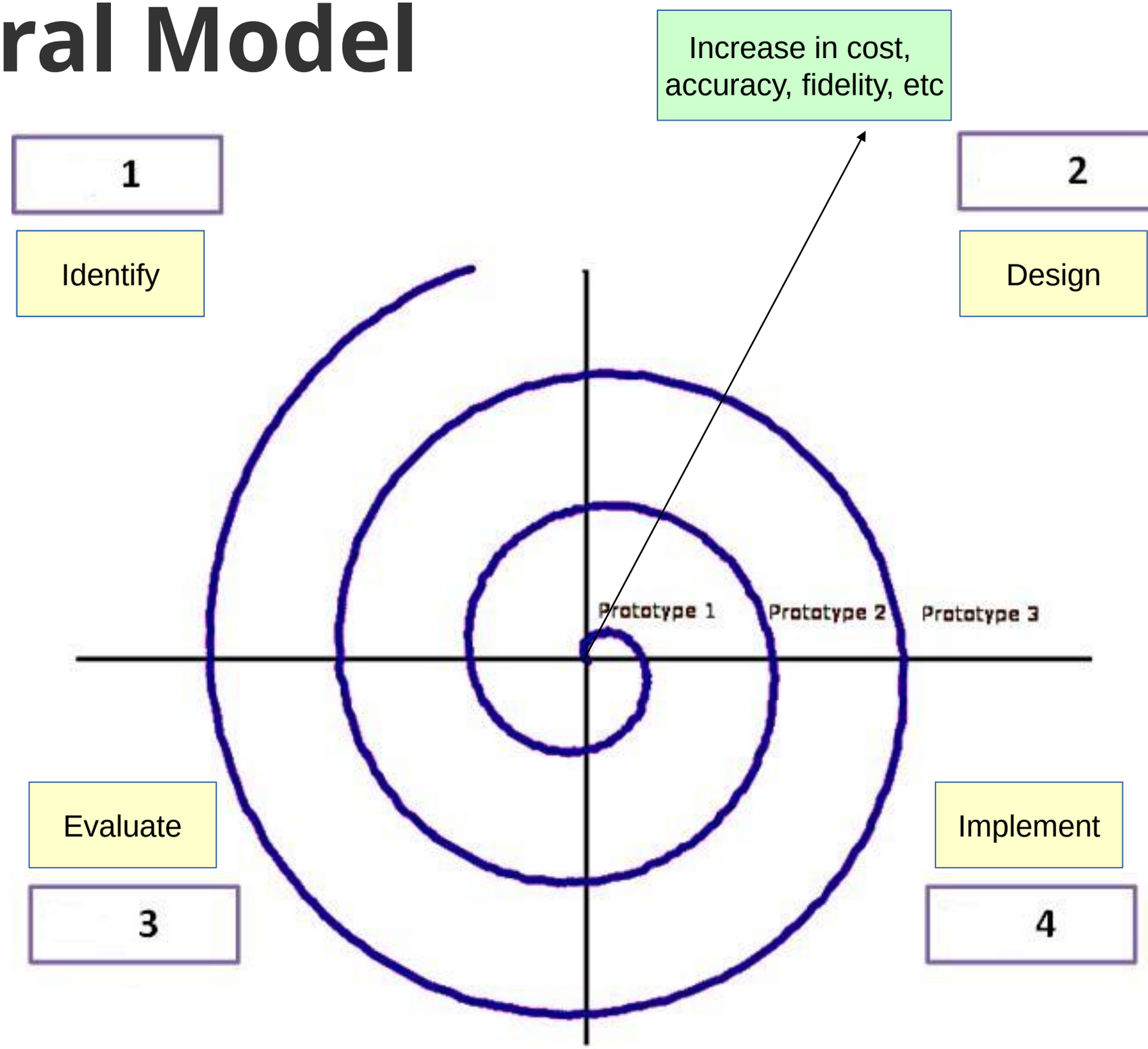
Iteration Strength

- Some working functionality can be developed in very early stage
- End users' responses can be obtained early and periodically
- Parallel development
- Measurable progress
- Easier and less costly to change requirement
- Risk management

Iteration Weaknesses

- Requirement for resources could be high
- High level of management requirement and complexity
- Working without the entire requirement established is risky

Spiral Model



Spiral Model

- Early iterations use cheap prototypes
  - May also build multiple cheap prototype alternatives for evaluation
  - Risk is greatest in early iterations
  - Put less resources and commitment in early stages
- Later iterations use richer implementations
  - Risk is much lower after a few iteration
  - Ready to spend more money on more detail
- Every iteration is evaluated by the end user
  - Users involved in all iteration
- More iteration generally means better quality

Spiral Model Strengths

- Provides early indication of insurmountable risks, without much cost
- Users see the system early because of rapid prototyping tools
- Critical high-risk functions are developed first
- The design does not have to be perfect
- Users can be closely tied to all lifecycle steps
- Early and frequent feedback from users
- Cumulative costs assessed frequently

Spiral Model Weaknesses

- Time spent for evaluating risks too large for small or low-risk projects
- Time spent planning, resetting objectives, doing risk analysis and prototyping may be excessive
- The model is complex
- Risk assessment expertise is required
- Spiral may continue indefinitely
- Developers must be reassigned during non-development phase activities
- May be hard to define objective, verifiable milestones that indicate readiness to proceed through the next iteration

When to use Spiral Model

- When creation of a prototype is appropriate
- When costs and risk evaluation is important
- For medium to high-risk projects
- Long-term project commitment unwise because of potential changes to economic priorities
- Users are unsure of their needs
- Requirements are complex
- New product line
- Significant changes are expected (research and exploration)
-

Which model to use...

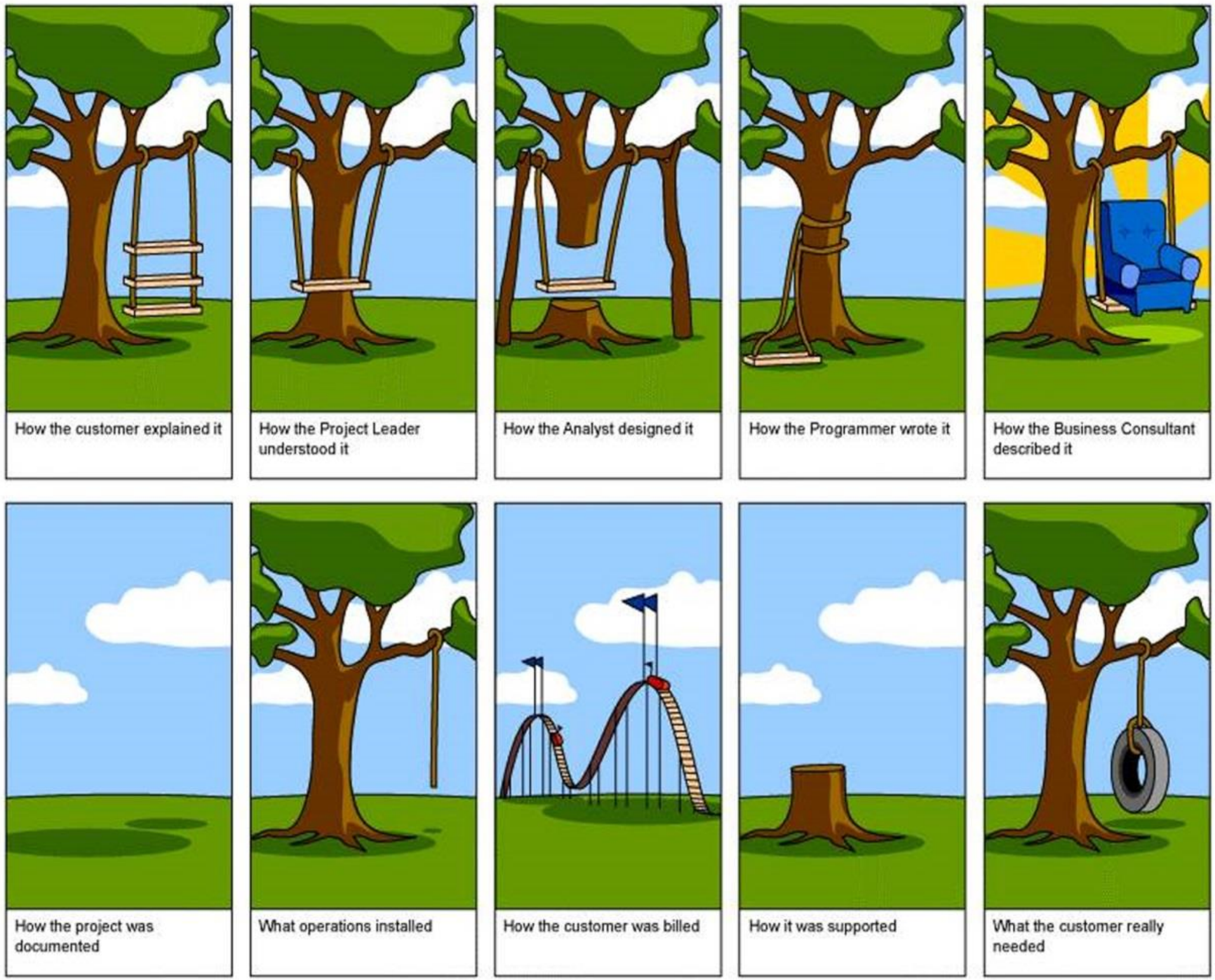
- It doesn’t have to be a one or the other decision
- Many systems are a blend
  - With some parts are prototyped to elicit requirements
- There isn’t one ‘best way’
- Nor is there a ‘silver bullet’

Requirement Analysis

- Overview of user requirements
- Data gathering for requirements
  - Task Analysis
    - Hierarchical Task Analysis
    - Observation
    - Ethnographic Study
  - User Analysis
    - Survey
    - Interview and Focus Group
  - Data analysis and presentation
-

What, how and why?

- Why bother? Requirements definition is the stage where failure occurs most commonly



Getting requirements right is crucial

Establishing requirements

- What do users want? What do users ‘need’?

  Requirements need clarification, refinement, completion, re-scoping

  Input: Requirements document (maybe)
  Output: stable requirements

  - Why ‘establish’?

    Requirements arise from understanding users’ needs

Requirement Analysis Objectives

1)Clarify customer's needs

- Customer's requirements are ALWAYS unclear
- Evaluate feasibility (i.e. feasibility study)

2)Establish Agreement

- Define scope of project
- Establish schedule and constraints.
- Generate Requirement Document
  - Serve as an agreement (contract) for the rest of the project

3)Create system definitions

- Terminologies
- For effective communication
- Prevent any potential miscommunication

Management Questions

- How much effort put towards analysis?
- Who does the analysis?
- Clarify responsibility and liability
- Why is it so difficult?
- Bottom line - who pays for it?

Feasibility Study

- Economic feasibility
  cost/benefit analysis
- Technical feasibility
  hardware/software/people, etc.
- Time feasibility
- Legal feasibility
- Alternatives
  there is always more than one way to do it

What is Requirement

- Requirement
  features of system or system function used to fulfill customer’s needs
- Focus on customer’s needs and problem, not on solutions
-

- **Requirements Definition Document**

  - End product of User Requirement Analysis
  - System background and objectives.
  - Description of the problem.
  - Detailed characteristics of the system (data & functionality).
  - Description of operating environment.
  - Requirement document may be splitted into 2 parts: Internal and External:

    Requirements definition document (written for customer).

    Requirements specification document (written for programmer; technical staff).

- **System Analyst**

  - The bridge between customers and the technical staff
  - Talk to the customers using layman terms
  - Talk to the technical staff using technical terms

- **Steps for User Requirement Specification**

- Data Gathering
    - Task Analysis
        - Hierarchical Task Analysis (HTA)
        - Observation / Ethnographic Study
    - User Analysis
        - Survey
        - Interview
        - Focus Group
- Data Analysis
- Produce User Requirement Document

Different kinds of requirements

- Functional:
  - What the system should do
- Non-functional:

  Performance (how well is system functioning).
  Documentation.
  Data (qualitative stuff).
  Resources (finding, physical space).
  Security (backup, firewall).
  Quality assurance (max. down time, MTBF, etc.).

- Data:
  - What kinds of data need to be stored?
  - How will they be stored (e.g. database)?

Different kinds of requirements

Users: Who are they?

- Characteristics: nationality, educational background, attitude to computers
- Age, gender, culture, language
- Education (literacy? numeracy?)
- Physical limitations
- Computer experience (typing? mouse?)
- Motivation, attitude
- Domain experience
- Application experience
- Work environment and other social context
- Relationships and communication patterns with other people

- System use: novice, expert, casual, frequent
  - Novice: prompted, constrained, clear
  - Expert: flexibility, access/power
  - Frequent: short cuts
  - Casual/infrequent: clear menu paths

Different kinds of requirements

Different class of users

- Many applications have several kinds of users
- By role (student, teacher)
- By characteristics (age, motivation)

  - novice, expert, casual, frequent

    - Novice: prompted, constrained, clear

    - Expert: flexibility, access/power

    - Frequent: short cuts

    - Casual/infrequent: clear menu paths

Personas

- Capture a set of user characteristics (user profile)
- Not real people, but synthesised from real users
- Should not be idealised
- Bring them to life with a name, characteristics, goals, personal background
- Develop a small set of personas

Example Persona

CAPLIN

Ginnie

BACKGROUND

- 15, Female
- Ongoing Private Education
- Ambitious
- Comfortable using technology to communicate

MOTIVATIONS

- Keeping in touch with her network
- Fashion/street cred
- Keeping up with peers.

FRUSTRATIONS

- Sad people trying to be 'friends' on Facebook
- Having to be in bed @ 11pm
- Being swamped in friends updates
- Missing important status updates

Receives private tutoring in Maths and English as these are not her strong subjects. Enjoys playing for the school's 2nd teams for netball and Lacrosse and is good at art.

She loves recording her favourite shows: ER and Sun Valley High on Sky+ and spends some of her time on her Laptop that Daddy bought her watching videos on YouTube, downloading music, keeping up to date with her friends on Facebook and chatting via MS IM to her cousin who is at University in Leeds.

She loves Ugg boots and Abercrombie & Fitch and uses the Internet to shop and find the cheapest prices.

"I want to easily hook up with my friends whilst watching TV"

Stakeholders

- Everyone who is “affected” by your system
  - Customers of your client
  - Suppliers of your client
  - Maintenance staff
  - Business Partners
  - Consultants
  - etc ...