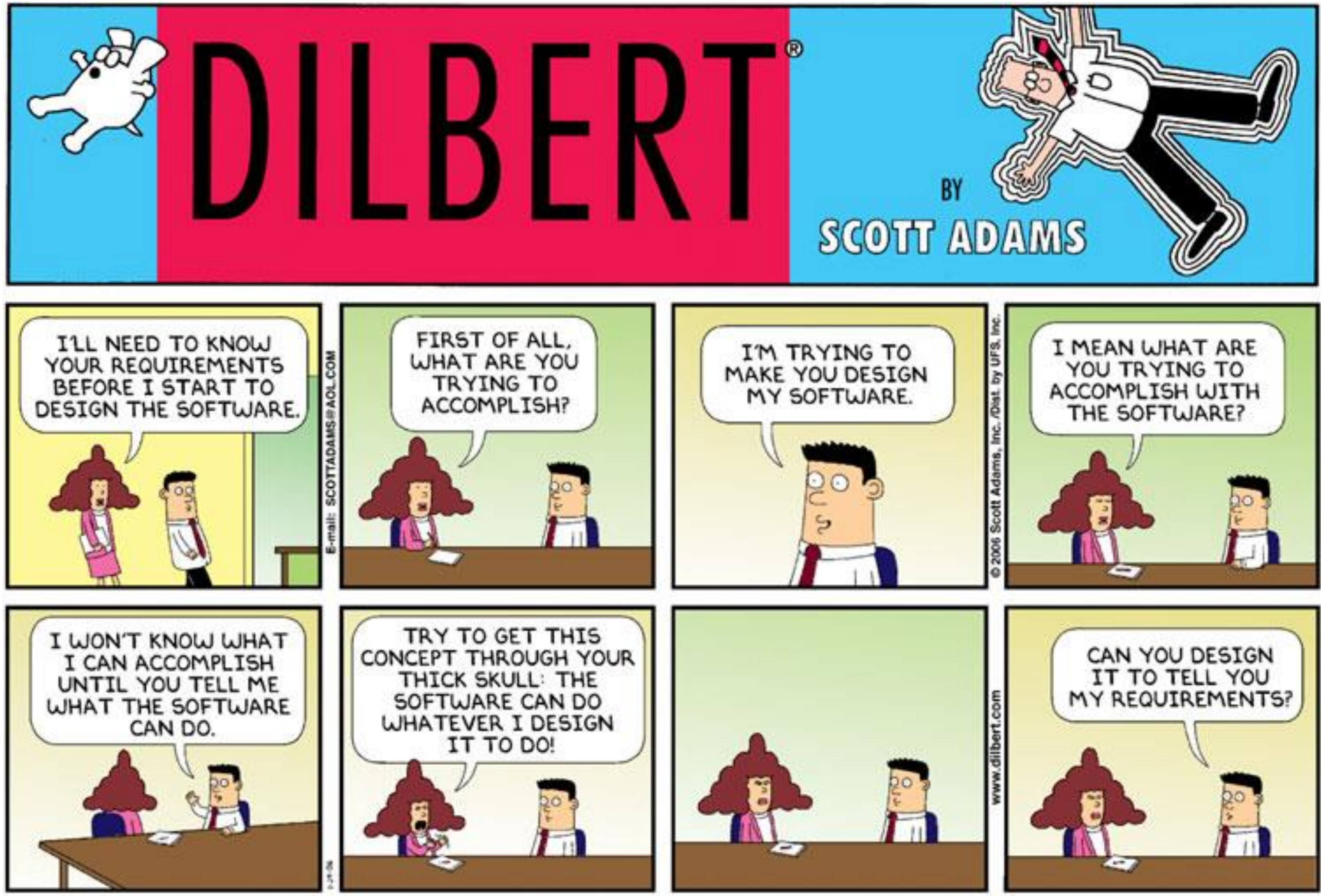
User Interface Design


DILBERT®
BY SCOTT ADAMS
I'LL NEED TO KNOW YOUR REQUIREMENTS BEFORE I START TO DESIGN THE SOFTWARE.
E-mail: SCOTTADAMS@AOL.COM
FIRST OF ALL, WHAT ARE YOU TRYING TO ACCOMPLISH?
I'M TRYING TO MAKE YOU DESIGN MY SOFTWARE.
© 2006 Scott Adams, Inc. /Dist. by UFS, Inc.
I MEAN WHAT ARE YOU TRYING TO ACCOMPLISH WITH THE SOFTWARE?
I WON'T KNOW WHAT I CAN ACCOMPLISH UNTIL YOU TELL ME WHAT THE SOFTWARE CAN DO.
TRY TO GET THIS CONCEPT THROUGH YOUR THICK SKULL: THE SOFTWARE CAN DO WHATEVER I DESIGN IT TO DO!
www.dilbert.com
CAN YOU DESIGN IT TO TELL YOU MY REQUIREMENTS?

© Scott Adams, Inc./Dist. by UFS, Inc.

UI Design: Wed Aug 31 (Day 5)

- Dr. Jillian Aurisano

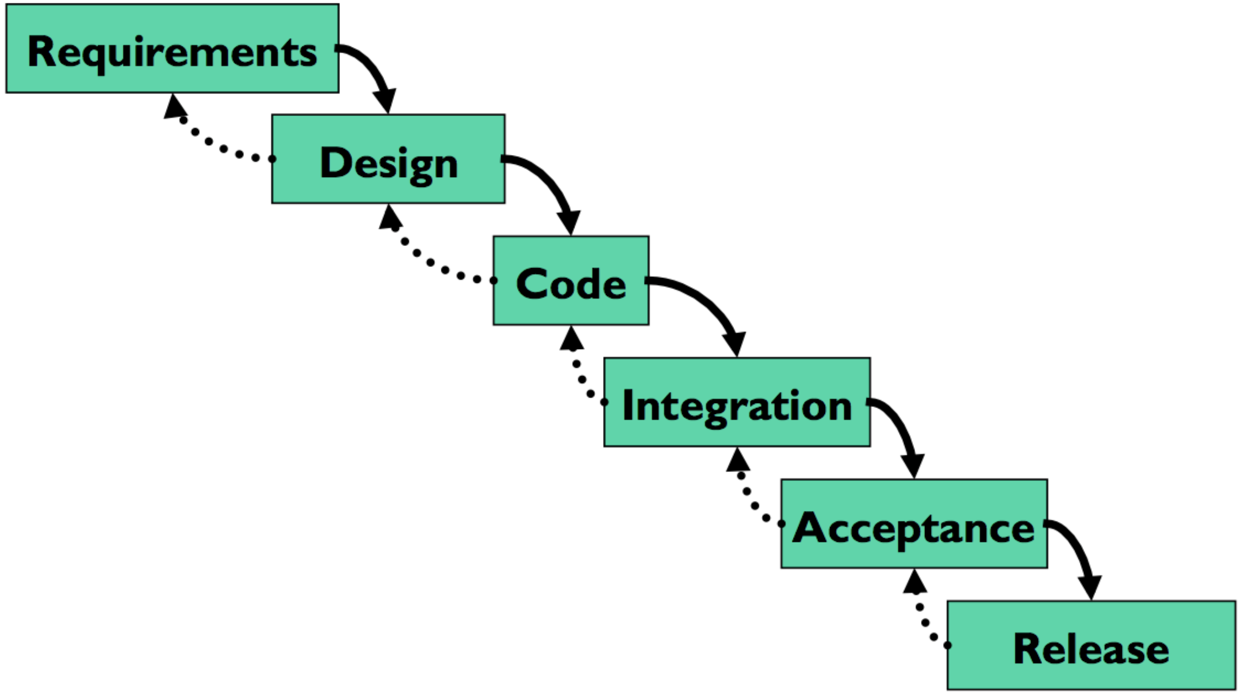
- Plan for today:
  - Reminders, course logistics
  - Finish design lightning presentations
  - Intro user centered design
  - Needs finding / requirements gathering
    - Concrete tasks vs essential goals
    - One method: interview
  - Do an interview activity

Course logistics

- Assignment 1 turned in
- Shift in schedule:
  - This week- user centered design, continuing to build our project ideas
  - Next week- sketching week
  - Then 2 technical weeks
- New project schedule:
  - Project 1 out week 3 (next week), due Week 7 (5 weeks)
  - Project 2 out week 8, due week 11 (4 weeks)
  - Project 3 out week 12, due week 15 (4 weeks)

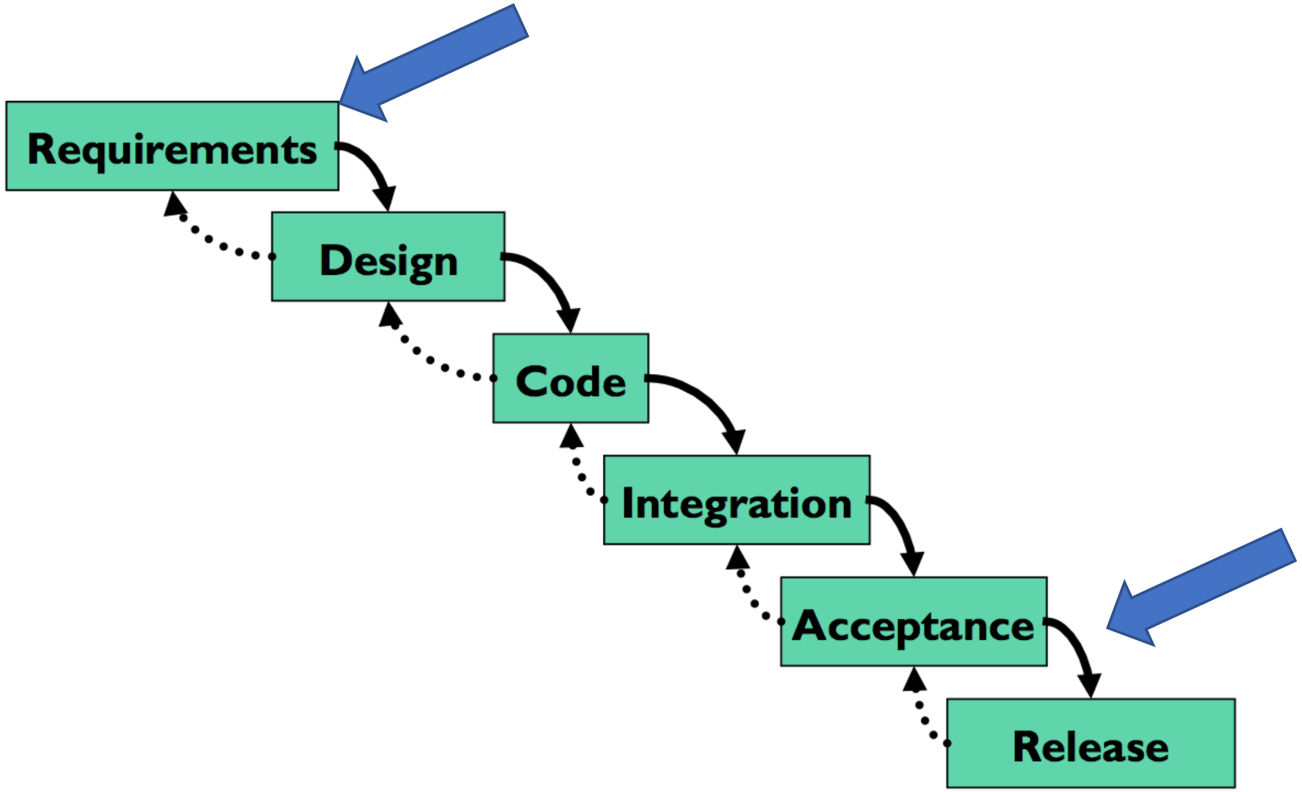
Traditional software development process

- Waterfall model
  - Anyone encountered this in class before?
  - Can you describe it?
- Is this a good model for user-centered design?
  - Where is the user?



Not enough user engagement in the process

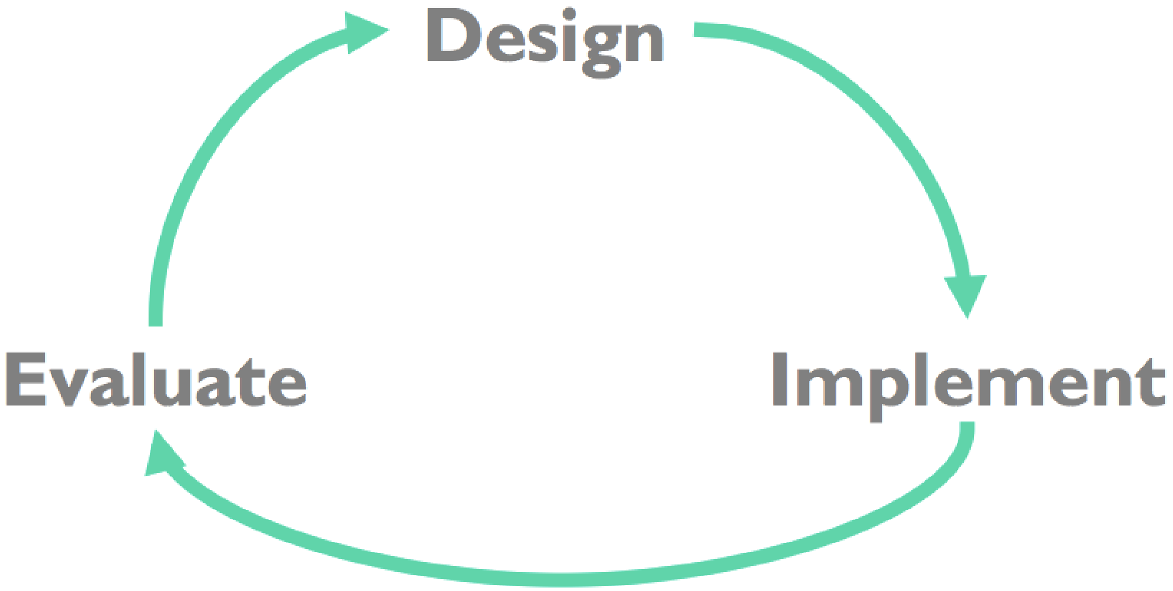
- User only included in the first step: capturing requirements and the last step- acceptance and release
- You can go back up the waterfall
  - But integrating user feedback in not considered at all stages
- What can go wrong?
  - Don’t find out until the end if user’s like it
  - Then you are too far along to make many changes
  - Or have to throw out well written code, because it isn’t useful



Hence, User-Centered Design is iterative

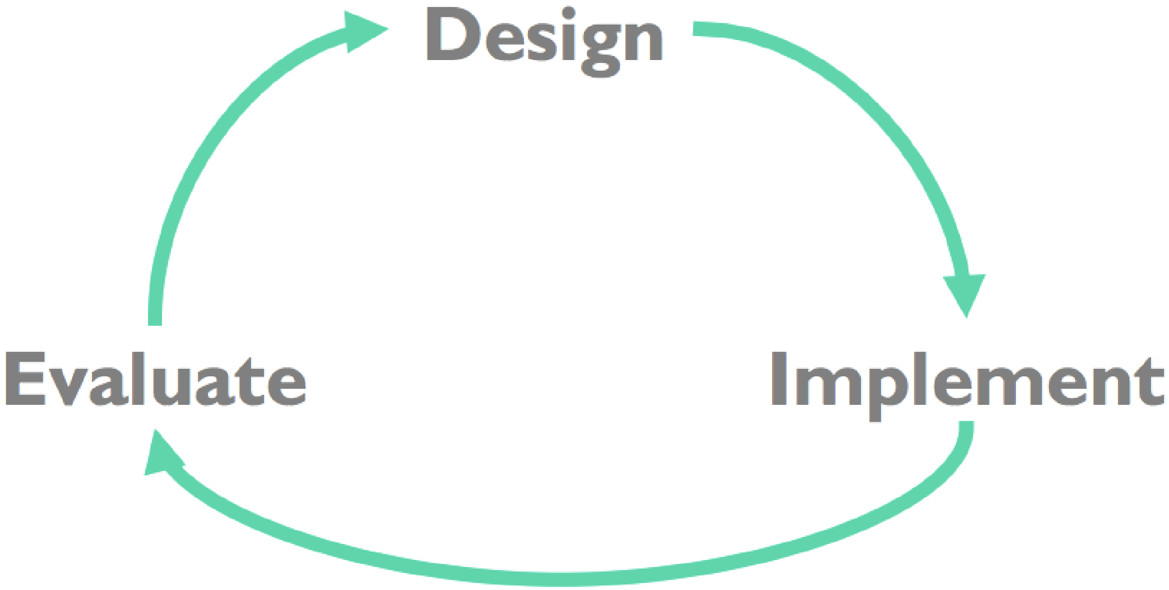
- Iterative design
- Early focus on users and tasks
- Constant evaluation to inform new design choices

- We will get into this but first….



How do we enter the cycle?

- Needs finding, understanding out users



Before we design: Needs-finding

- Need-finding or Analyzing User Requirements

- We need to understand our users
  - Who are they?
  - What are their goals?
  - What should the design help them do?

Know your user

- Things to learn
  - 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
- Pitfall
  - describing what you ***want*** your users to be, rather than what they actually are
    - “Users should be literate in English, fluent in spoken Swahili, right-handed, and color-blind”

Know your user

- There can be multiple classes of user
  - Primary users, Secondary users
  - Or other divisions (expert users, novice users….)
- These distinctions can help you think about different sets of goals, or different expectations or levels of experience

How to go deeper than a user profile? Do a user analysis, working with real users

- Requirements gathering, establishing design requirements
  - What is it that our design is going to support
- Not about systems requirements (yet)
- Not about design of the UI (yet)
- All about what people need

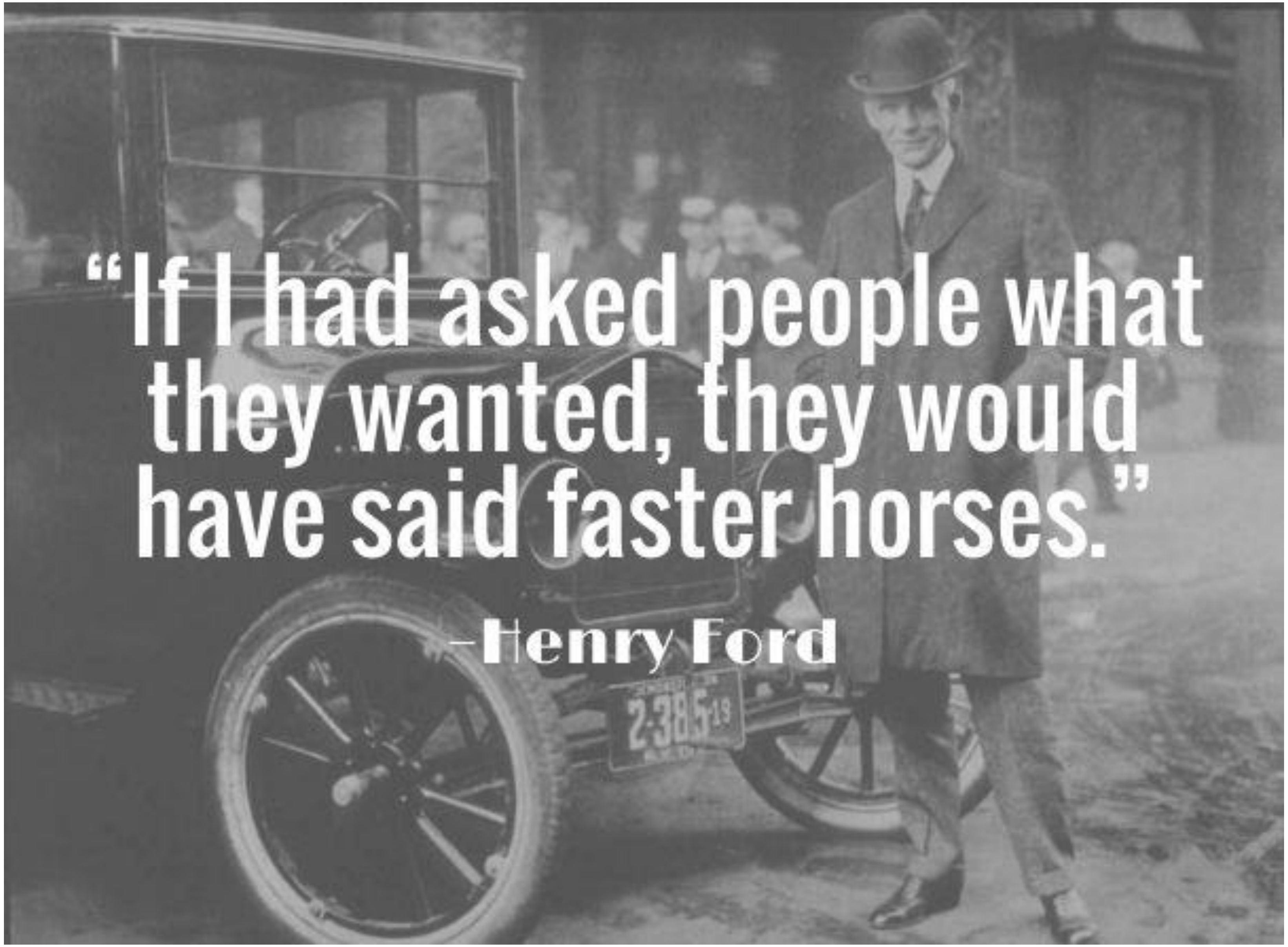
- Methods:
  - Observation (we will cover later….)
  - Surveys (we will discuss later…)
  - Interviews- we will begin working on today

Why not think first about the system or UI design?

- Pitfall: Begin with the system you want to build
  - System: “Notify the user about an appointment”
  - User need first: “Get notified about my appointments so I don’t miss them”
- Pitfall: Begin with UI design goals
  - UI design idea: “The system bell will ring to notify the user about an appointment…”
  - User needs first: I need to be alerted in a way that it can be detected by me even if I am not looking at a device
- Why does this matter?
  - Solution focus BEFORE you fully understand the needs and have tested alternate designs

Concrete tasks vs essential goals

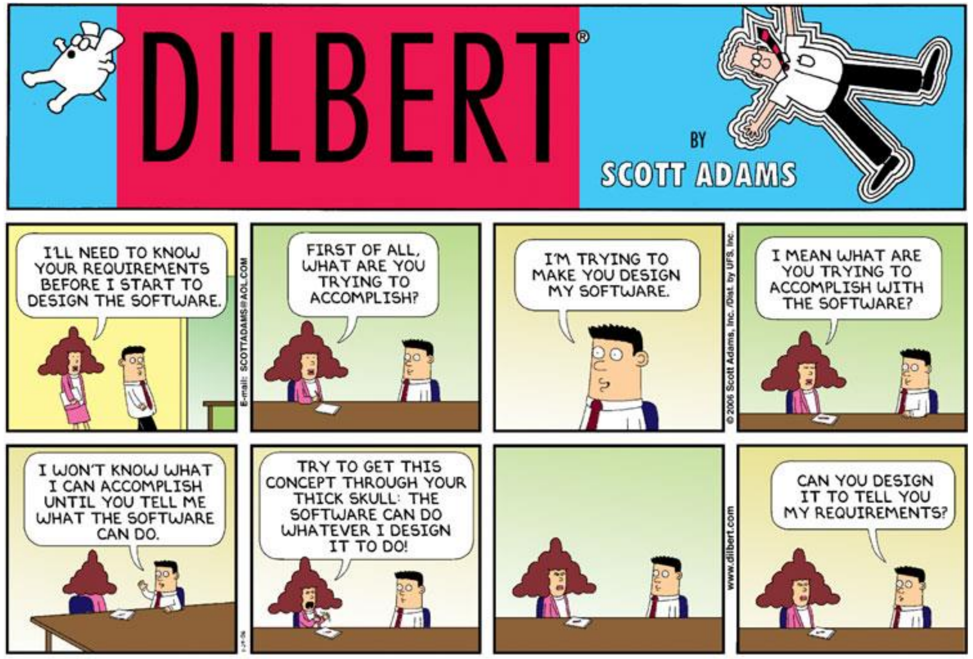
- Sometimes users may express something they want, good to ask yourself:
  - is this based on what they currently do? (Concrete task they perform)
  - Is it based on what is behind this current practice (essential goal)
- Example:
  - User : ”I need to save the file to disk” (Concrete task)
  - Requirement: I need to preserve my work.
- Why:
  - As before, be open to alternate solutions not based on current practices
  - What might we come up with instead??



“If I had asked people what they wanted, they would have said faster horses.”

–Henry Ford

Need finding pitfall: Expecting the users to tell you what to design


DILBERT®
BY SCOTT ADAMS
I'LL NEED TO KNOW YOUR REQUIREMENTS BEFORE I START TO DESIGN THE SOFTWARE.
E-mail: SCOTTADAMS@AOL.COM
FIRST OF ALL, WHAT ARE YOU TRYING TO ACCOMPLISH?
I'M TRYING TO MAKE YOU DESIGN MY SOFTWARE.
© 2006 Scott Adams, Inc./Dist. by UFS, Inc.
I MEAN WHAT ARE YOU TRYING TO ACCOMPLISH WITH THE SOFTWARE?
I WON'T KNOW WHAT I CAN ACCOMPLISH UNTIL YOU TELL ME WHAT THE SOFTWARE CAN DO.
TRY TO GET THIS CONCEPT THROUGH YOUR THICK SKULL: THE SOFTWARE CAN DO WHATEVER I DESIGN IT TO DO!
www.dilbert.com
CAN YOU DESIGN IT TO TELL YOU MY REQUIREMENTS?

© Scott Adams, Inc./Dist. by UFS, Inc.

- You can't just ask the users what you should be designing
  - Ask them what they need to get done
  - Ask about their goals
  - Ask about their problems

Take note:

- User analysis is different than market analysis

- Not- will users use my product
- Not- is there a market for this product
- Not- what will people pay

- These are important questions, but often approached with different techniques

What is the end result?

- Purpose :
  - The requirements analysis must produce a readable and clear, documented set of user requirements that informs the design activity

- Focus:
  - Requirements should be expressed at a level of detail which is appropriate for leaving room for alternative design exploration (within the frame established by the requirements)

- **NO:** user sees a pop-up asking to register, he enters username and password, click on submit and the pop-up close, showing a page with the top navigation bar and in the center a dashboard with all the personalized information displayed in a grid metaphor….
- **YES:**
  - What user needs to do ultimately: The users' goal is to access their assignments and lecture notes, so they can keep up in their class
  - What problems do they face: The user struggles to find what they need, and it delays their work
  - What do they want?
    - The users wants to have personalized info in the app, saved for every time they use it
    - User needs to be able to see their high priority tasks quickly as soon as they start using it, and not have to scroll through lots of options
    - User wants to have it be a familiar layout, without too many options
    - ….

How can we do need finding?

- Observation (we will learn some methods later)
- Surveys (we will discuss later)
- Interviews- we will begin working on today

- (Note- interviews and observations can ALSO be used for evaluation….)

Interview Guidance: how to do it

- Attitude:
  - YES: Learning, understanding, identifying, discovering, unveiling, clarifying (= open to change)
  - NO: confirming, demonstrating, re-assuring, achieving consensus, discussing opinions (= closed to change)
- Key test of a good user interview:
  - What are 3-4 things you have learned that you did not know before
- Focus your questions on the **experience**
  - Not projection, prediction, extrapolation, “thoughts”
  - NO: Is this a useful application?
  - YES: Is this application valuable to the work you do right now? How? Why?

Interview Guidance: how to do it

- Focus on *immediate* experience
  - Capture current behavior
  - **NO**: Is this product <interesting> for you?
  - **YES**: If it were available right now, would you use it?

    Why? How?

- Non-judgmental attitude
  - Do not convey in any way that you are expecting a certain answer…
  - **NO**: Don’t you think this feature would be better if available also as a iPad application?
  - **YES**: Is there any other way you’d like to use a feature like this in your current work? How? Why?

User interview guidance

- Keep each question focused on a *single topic*
  - Avoid using AND / OR in your questions (linking more statements together)
  - NO: How would you use this app at school or at work, for example?
  - YES:
    - How would you use this app at school?
    - How would you use this app at work?

- Provide a *way out* from your options
  - What do you often use to get the news? Smartphone or tablet? *Or something else?*

- Even in close-ended questions, give the possibility of expressing things outside the available options
  - NO: which of the following feature is most important to you? [*what if no one is important?*]
  - YES: Rate from 1 to 5 how important each feature is for you, where 1 is least important, 5 is most important. Put 0 if a feature is irrelevant for you.

User Interview Guidance

Avoid *binary* questions

- Do not force black/white commitments on the “whole”
- Elicit “analytical” feedback on specific elements
- **NO**: Is this product useful?
- **YES**: What, if anything, do you find useful about this product? Why?

Activity: Lets try to do an interview

- First planning
- Then we will go into small groups and interview each other
- And then reflect