

Pre-Spec Work

Part 1 - Know Thy Users

CSE 112
Winter 2016
Prof. Powell

Sad Truths

- The majority of software you work on will be poorly specified
- Truth: Writing specs are hard
- Truth: People don't always know what they want
- Truth: Specs often require speculation about an unknown topic or future action

Fuzzy Spec Reactions

- With fuzzy specs we have two reactions from our previous discussion
 1. Spend time on big think
 2. Jump in and try to figure it out and adjust
- Both are fraught with peril but currently choice two is vogue for at least “progress” appears to be made and it appears less risky

It's Partially Our Fault

- We enable this activity when we aim to be ~~programmers~~ coders rather than engineers or architects
 - We are indeed one of the two parties to allow insanity in our projects
 - Either don't grumble or do something about it - you must realize the spec
- Interestingly Agile with its user stories is forcing you to do this as is the big think "discovery process"

User Acceptance

- Is paramount ... this will be formally called verification
 - "Are we doing the right problem even?"
- It trumps the how we accomplish things
- Users ultimately are the overall acceptor/rejector of the app in nearly all cases
- Sadly "customers don't know what they want. Stop expecting them to know what they want"

<http://www.joelonsoftware.com/articles/fog0000000356.html>

What to do...

- It is our job to figure out what users want
 - Both business owners and the end user
 - We have to be an advocate for both
- Beware: General design tension of control



Control

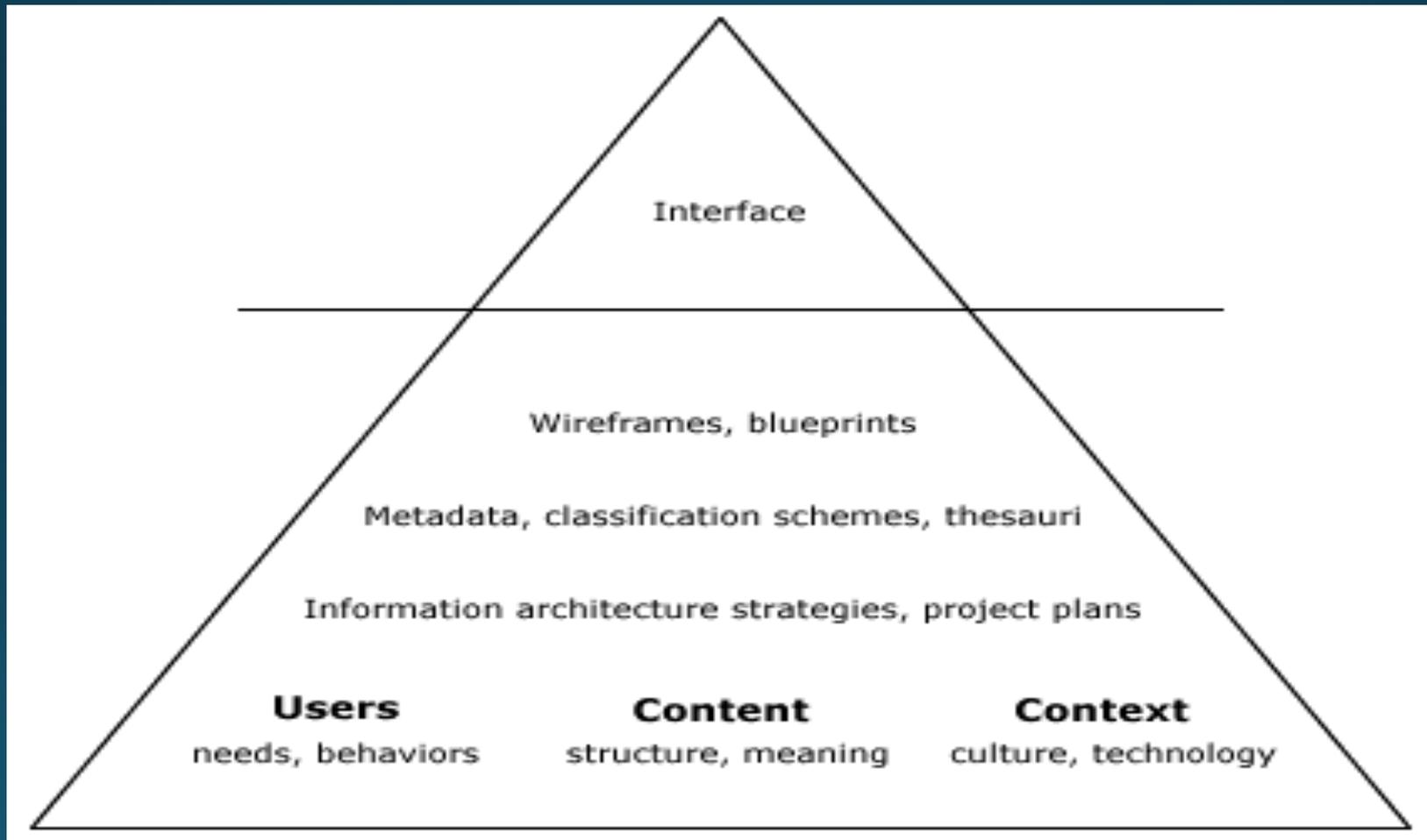
- Who is in control is a key idea of the vision phase of requirements
 - Too much user control - trouble, chaos, etc.
 - Too much owner control - missing real needs
- Again we see something
 - User control ~ bottom up
 - Owner control ~ top down

Practice Las Vegas Design

- Balance of power
- Illusion of control
- Benevolent Dictator
- Measure, measure, measure

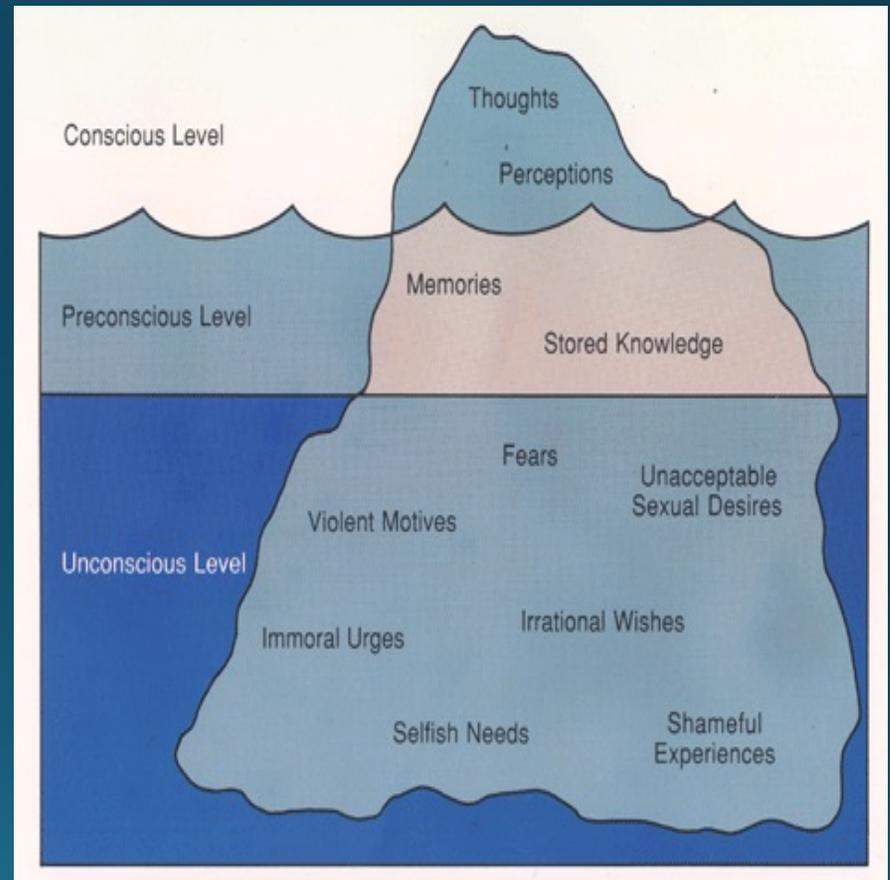


Accept the App Iceberg



This iceberg thing...

- Might go deeper if we applied it to the people themselves
- Freud thinking as iceberg
- Why they think the way they do about our app?



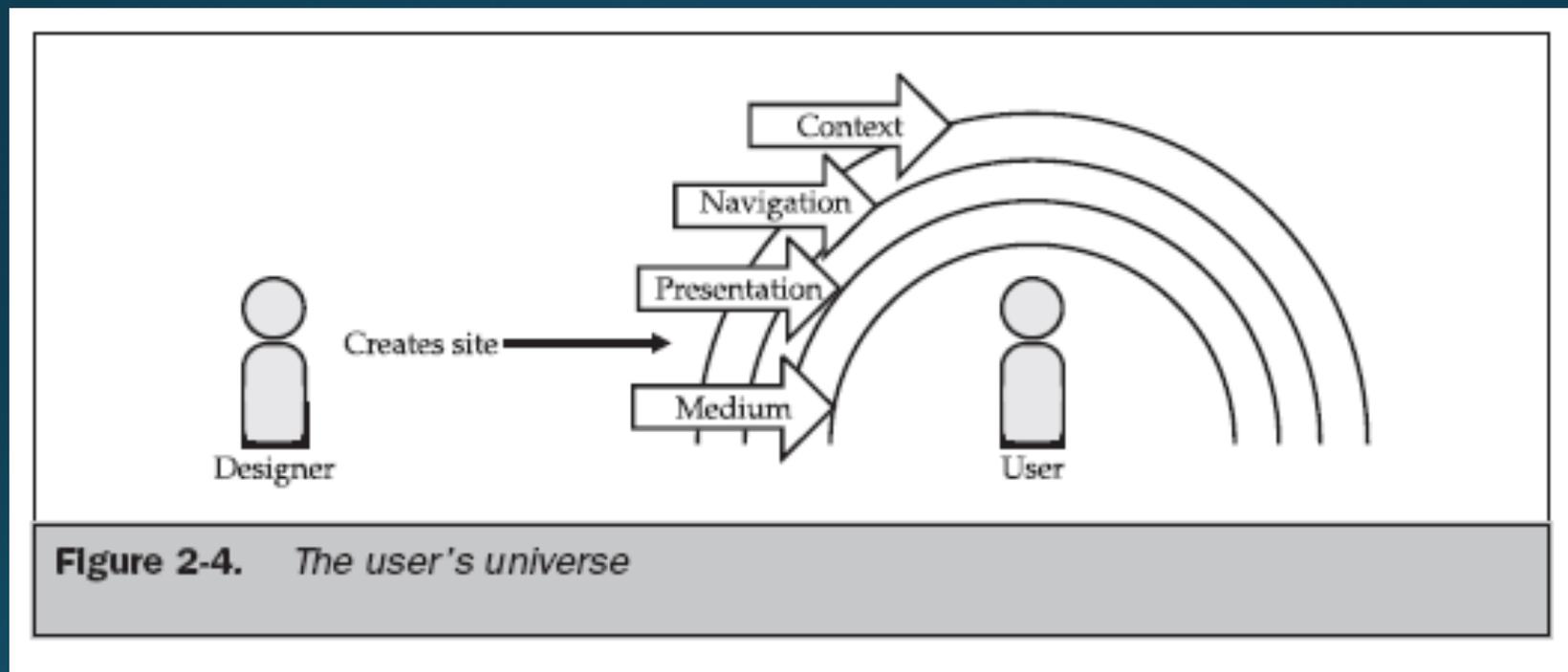
What do users want?

- Law: You are not the user*
- Reaction: Ask the users!
- Law: Users are not designers
 - "Sure you gave us what we asked for but it isn't what we want"
-- your users
 - They aren't aware of themselves as much as you think - it's up to you

User Naturalist

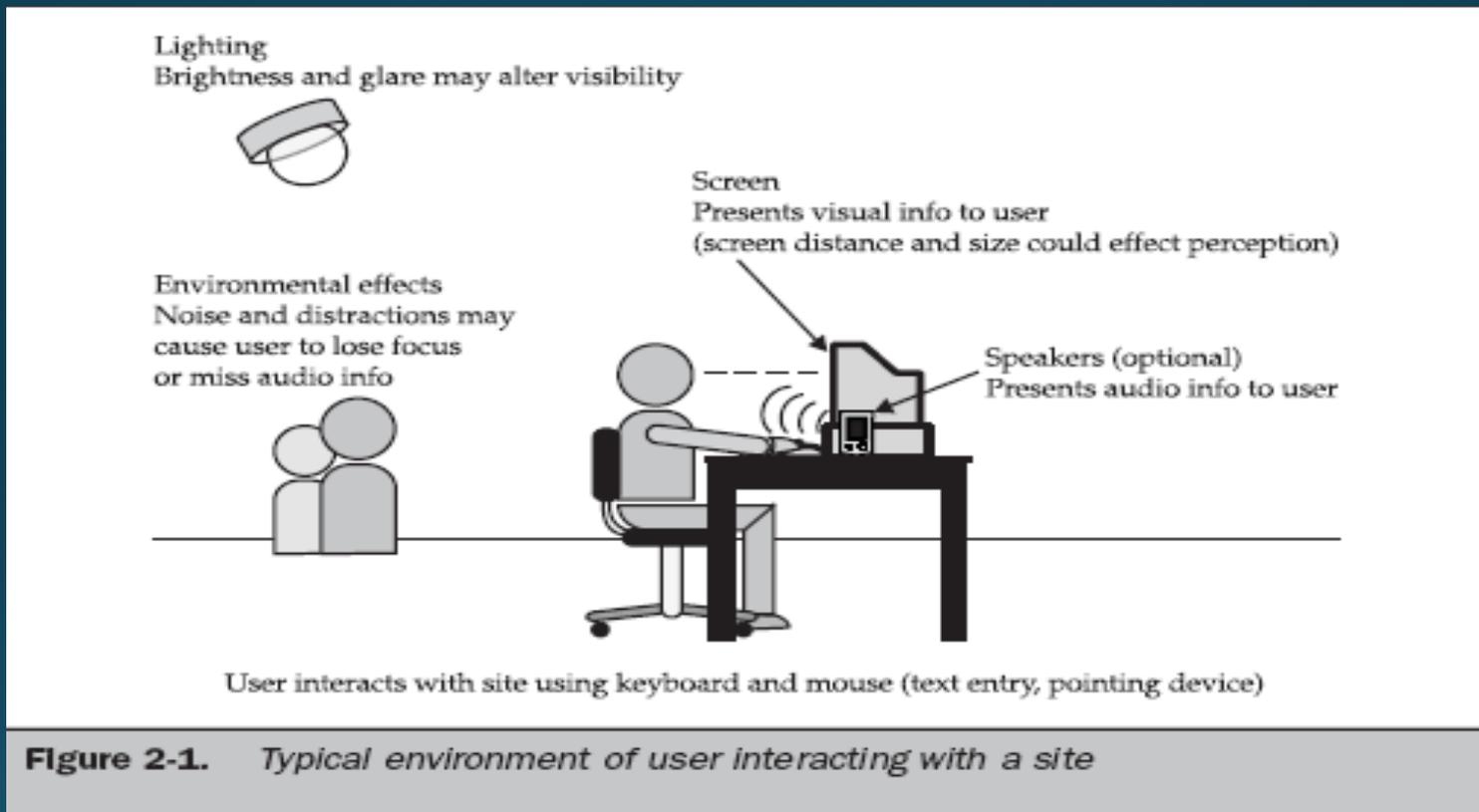
- Study your ~~prey~~ users in their habitat
- Do so with as little interference as possible
- Try to think like they do with empathy
 - No robots nor stereotypes, real people with real problems and personalities
 - And know that you **can't** please them all

We live in our own worlds



We go to them not the other way around really

A Common User Environment



There are many others

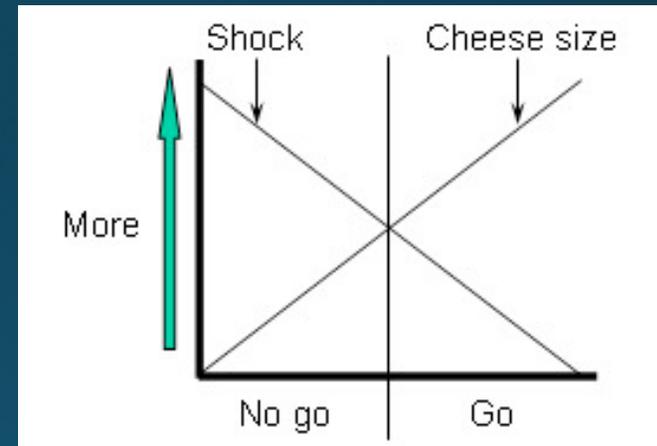
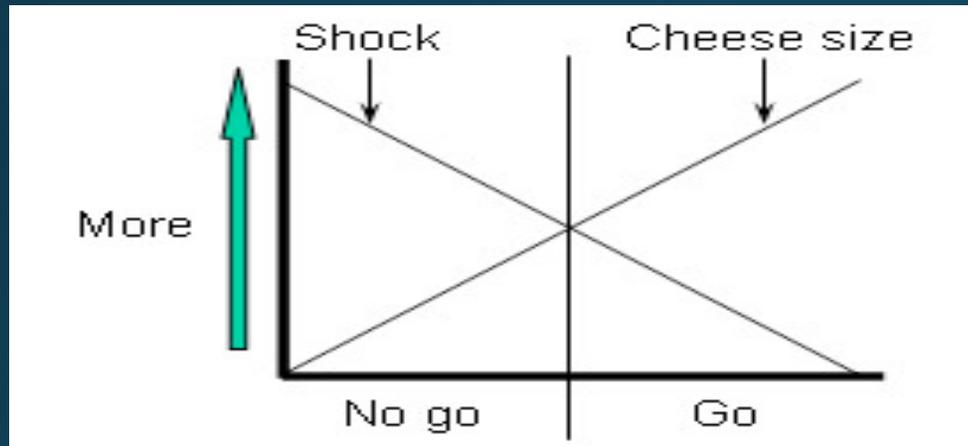
Watching Users

- Go meet them and listen
 - Avoid focus groups
- Watch out for testing artifacts
- Employ the power of passive monitoring
- Ultimately - “Don’t spook your prey!”

Meeting Those Users

- Video - “What is a browser”
- Truth: Most of them don’t care how you build something and are quite surface-y
 - No: ZOMG Facebook used PHP!!?!?
 - Yes: I don’t like how it looks.
- Might be tough to engineer against the irrationality...even when it can be predictable

Our App and Users?



- Mostly our cheese will not overcome even mild shock
- Exceptions? *Fear Factor*
- Equate electric grid to ease of use

Usability

- Definition: Usability is the extent to which an app can be used by a specified group of users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.
- From Nielsen 5 aspects of usability
 1. Learnability
 2. Rememberability
 3. Efficiency of use
 4. Reliability in use
 5. User satisfaction
- Q: Does one matter more than others ultimately?

Usability

- Rule: There is no absolute idea of a usable application.
- Rule: Usability varies by the user
- Rule: Usability depends on medium of consumption
- Rule: Usability is highly influenced by app type and familiarity with app

Avoid Sporknifes



Law: Good usability and user satisfaction are directly related

Humans to Tech not the Reverse



Discuss the right and wrong of this idea

Ok if I am wrong ... this will be us in the "future"



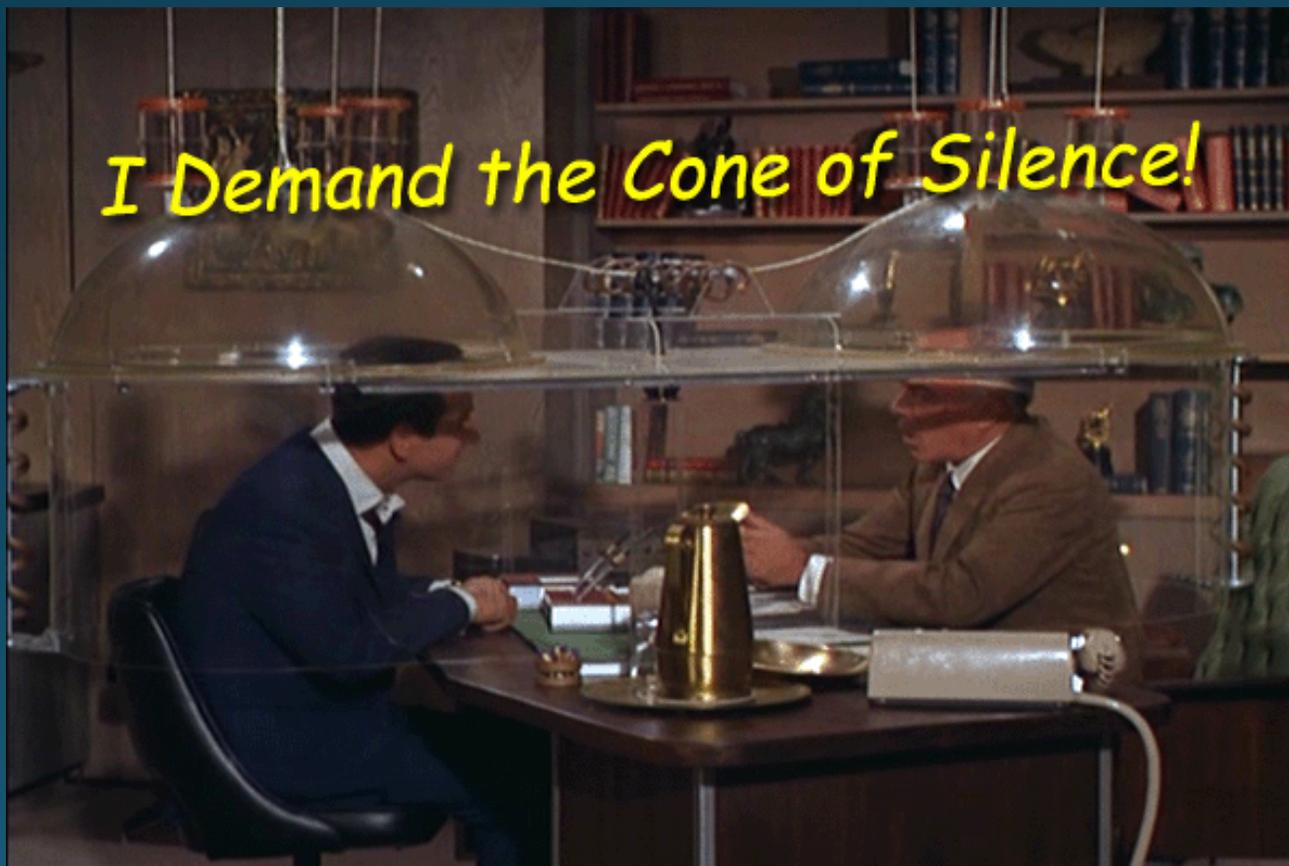
Information worker in Tom Cruise wave your arms world

Take 2: Talking to the Computer



Again let's reason this out given our theory

Ok I am wrong...we need these then



Common Characteristics

- While there may not be a totally typical person people do have similar capabilities. Remember the main way that a user interacts with a Web site (keyboard, mouse, monitor).
 - Sight
 - Movement and Reaction time
 - Memory
- Other possibilities may include hearing
- All capabilities have ranges - example: vision

Vision

- Visual design of our app must account for vision limits
- Contrast
 - Light on light, dark on dark = BAD
 - Consider - road signs, typical print media



Vision Contd.

- Size and Spacing
 - Avoid too small and too big
 - Too close together
- Noticeably different
 - Thresholds and the “fuzzy eye” test
- Gestalt Principles - similarity, continuation, closure, proximity, figure & grouping
 - <http://graphicdesign.spokanefalls.edu/tutorials/process/gestaltprinciples/gestaltprincip.htm>
 - http://en.wikipedia.org/wiki/Gestalt_psychology#Gestalt_laws_of_grouping

Vision Contd.

- Don't forget color
 - Is it all we rely on in an interface?
 - What about color blind?
 - How is it perceived?
 - Social, Culture, Taste
 - Not absolute even w/o human - monitor age, settings, gamma, etc.

Memory

- Rule: Users try to maximize gain and minimize work. (The lazy person rule)
- Memory requires work!
- Rule: Recognition is easier than recall, so don't force users to memorize information.

Memory Contd.

- Miller's Law - Short term memory wise, users are able to remember around 7 ± 2 items.
- This is important to consider in app dev
- Add to it though the depth and breadth problem
 - Sequential memory constraints - 3?

http://en.wikipedia.org/wiki/The_Magical_Number_Seven,_Plus_or_Minus_Two

Time and Speed

- Not absolute
- Reaction times have a floor
 - Sub 1-second
- Attention times checkpoints
 - 0-1s, 1s, 5-10s, >10s
- Never have enough speed, though wait t correlate
- Speed should be a feature



Movement

- Consider finger, mouse, keyboard
- Zen UI - Minimize movement
- Fitt's Law - the math basically shows time to acquire/activate target is function of distance to and size of target
 - Big and close vs. small and far

$$T = a + b \log_2 \left(2 \frac{D}{W} \right)$$

Diagram illustrating Fitt's Law equation with annotations:

- Time* (with a downward arrow) points to T .
- Distance* (with a downward arrow) points to D .
- Coefficients* (with an upward arrow) points to a and b .
- Width* (with an upward arrow) points to W .

Engineering UI Thinking

- Acknowledge the fallibility of our humans
- As a good engineer have a fail safe
- If you have a system you have many ways to succeed in case a system fails
 - Many ways to accomplish task - keyboard, mouse, wizard, etc.
 - Many ways to notice button/link - color, style, location

General User Types

- In general there are three types of users
 1. Novices
 2. Infrequent intermediates
 3. Power users
- These are gross groupings
- Q: What should we focus on?

Real Users

- Users are not so simple as we have heard already.
- Each is an individual
 - We may find personas useful though
- Talking to real users will be confusing
- Do not be an absolutist here and really believe next slide

Humans - Irrational, Mostly-Constant, Social Animals

- Humans are not computers
- Humans have emotions
- Humans are fairly constant, technology is variable
 - Slow change over time (aka evolution)
 - Do not confuse societal changes
- Lots of Real Challenges Out There - Movie time:
<https://www.youtube.com/watch?v=PrISFBU5CLs>

Back to Vegas

- If we understand them better than they understand themselves is that an advantage?
- Visiting Cialdini's Influence ideas
 - http://en.wikipedia.org/wiki/Robert_Cialdini
- Google does this? Amazon does this?

Starting to Get Practical

- Before writing code need to have a pretty good idea what the user interface will look like
- “The interface is the application”
- This suggests interface first, wireframing, etc.
- Of course we need to consider back-end too but this fits iceberg model
- Up next ...
 - Medium Constraints
 - Requirements