CSC318 Tutorial **January 27, 2004**
Assignment 2 Prep

Why do field work?

- The purpose is **NOT** to confirm an *existing* plan or design
- You don't know all the answers! Introspection alone is dangerous.
- Stakeholder perspectives are extremely valuable
- Ideas are generated to understand stakeholders' needs, interests, abilities, limitations, etc.

Tips For Formulating a 'Concept":

- What problem are you solving?
- How is this problem solved now, if at all, and why?
- Who are the user's? Who are the stakeholders?
- What a priori knowledge/access/technology do they need to use your system?
- What artifacts are created/used in the process?
- What is the environment of those artifacts?

Tips For Conducting Field Work

Concepts	Tips
Tacit Knowledge E.g. Tell me how to tie your shoelaces	• Don't be afraid to ask for stories • Consider spending time with the participants in action (i.e. ethnography)
Social Pressures E.g. What do you think of Prof. Posner?	• Consider the difference between the 'right' answer and the 'real' answer • Word problematic questions carefully
Assumptions and Naiveties E.g. Postal codes of online webforms E.g. Asking if they want a functionality begs a positive response	• Avoid questions that 'lead' the participants or make assumptions about them
Settings People perform better in the natural setting	• Consider the location of the interview – try making it in the same location as the topic you are discussing
Biases E.g. Asking your mom if it's a good design	• Consider interviewing people who don't know you
Invasiveness E.g. People act differently when being watched	• Keep recorders in less obvious places, e.g. videotape from behind
Effort & Organization Make it easy for the participant KISS: Keep It Simple Stupid!	• Transcribe your speeches ahead of time • Have an ice breaker • Breaking down complex questions into simpler questions. • Watch for questions that will likely have the same answer from everyone. Consider rewording in order to get more info from each participant • Anticipate potential answers. What are you trying to learn? Are you learning anything new? For closed ended questionnaires, make sure to include the most popular answer as an option. • When you may get varied details from each participant consider open ended questions, to learn more about each individual's needs. • Watch spelling and grammar • Test out your questionnaire with a practice participant • Make sure not to re-ask questions from the survey in the interview. You don't want to waste the participant's time (nor yours).

CSC318 Tutorial **January 27, 2004**
Assignment 2 Prep

Task Analysis (PRS pgs 231-234):

- Deconstructing how something gets done into hierarchical unit tasks and subtasks.
- This is often done at either the "action" level, also done at the "keystroke" level when trying to optimize performance
- Can be difficult for complex, parallel, or interruptable systems
- Task analysis can be used at a broader level to deconstruct user behaviour as observed (and as planned)

Essential Use Cases (PRS pgs 229-231)

- More general than scenarios, avoids assumptions
- Three components: Name of user intention, steps of users actions, steps of systems responsibilities.

Scenarios (PRS pgs 223-226)

- Narratives that describe the user & stakeholders in their (current) environment and how your proposed project relates to that picture
- Identifies both problems and opportunities with your idea
- Note that you cannot always take what a user says at face value.

Requirements (PRS Ch 7)

- Requirements definition is the first step in translating needs into a solution.
- It may be tempting to pay little attention to detailed requirements, because you will not actually be BUILDING the real product. BUT, carefully considered and justified requirements are an excellent way to demonstrate a thorough needs analysis.
- User needs analysis is not meant to confirm an existing plan, and that final requirements come from user needs, NOT the reverse!

Step 1: Needs & Alternatives

- Given a set of needs that you have uncovered, what are all of the various alternatives that could be put in place?
- What are the advantages and drawbacks of each approach? Why is your chosen approach the best? Does your chosen approach really have a net advantage over alternatives?
- Consider
  - speed, efficiency, productivity
  - entirely new capabilities
  - same capabilities, but higher quality
  - learning curve/resistance to adoption
  - implementation costs
  - ongoing maintenance/upgrade issues
  - risks of unintended effects
  - opportunity costs

Step 2: Define Functional Requirements ("what")

- **Functional**: *what* the system does ("features")
- List of what the system must do
- Functional requirements often imply corresponding non-functional requirements
- Functional requirements may be added or dropped after prototyping phase – why?
- 75% of all software projects fail; the vast majority fail over non-functional requirements
- Non-functional requirements often represent tacit/implicit ("in the head") knowledge that is not obvious during needs exploration

Step 3: Define Non-Functional Requirements ("how")

- **Non-functional**: *how & where* the system operates, constraints on the system
- *Usability requirements*
- Data requirements (can include interoperability and disaster recovery)
- Mobility requirements
- Performance requirements
- Security/safety requirements (also functional)
- Operating requirements (environmental considerations)
- Availability requirements
- Lifecycle requirements (can include business goals)