Welcome to SENG 321

Requirements Engineering

Let's make this an engaging course

SENG 321
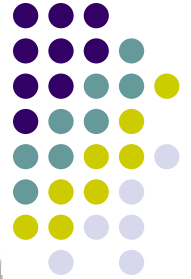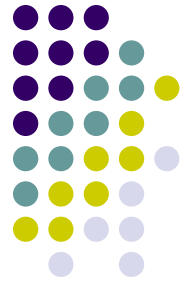
Professor Hausi A. Müller PhD PEng FCAE
Department of Computer Science
Faculty of Engineering
University of Victoria

http://www.engr.uvic.ca/~seng321/
https://courses1.csc.uvic.ca/courses/201/spring/seng/321

# BSENG Accreditation

Thank you

# Announcements

- S2 & C2
  - Posted

- Quiz 1 (tomorrow)
  - ***Wed, Feb 24 in class***
  - Class attendance must increase

- Midterm
  - Wed, March 2

- Final Exam
  - Sat, April 16
  - 19:00-22:00
  - ECS 125

# SENG 321 Calendar

| | | | |
|---|---|---|---|
| Quiz 1 | Wed, Feb 24 | In class | 2% of course |
| Midterm (revised) | Wed, Mar 2 | In class | 14% of project |
| Deliverable S2a (revised) | Fri, Mar 4 | S2a Detailed req spec; conceptual design | 10% of project |
| Deliverable S2b (revised) | Tue, Mar 8 | S2b Class presentation of S2a to customer | 5% of project |
| Deliverable C2 (revised) | Thu, Mar 10 | C2 feedback on S2a&S2b | 5% of project |
| Deliverable S3a | Tue, Mar 15 | S3a Technical Design Spec | 15% of project |
| Deliverable S3b | Tue, Mar 22 | S3b Manual | 10% of project |
| Deliverable C3 | Thu, Mar 24 | C3 feedback on S3a&S3b | 10% of project |
| Easter break | Mar 25-28 | Fri, no class | |
| Deliverable S4 | Mar 29-31 | S4 project demo | 10% of project |
| Deliverable C4 | Mar 29-31 | C4 feedback on S4 | 5% of project |
| Last Day of Classes | Fri, Mar 31 | | |
| Final Exam | Sat, Apr 16 | 19:00-22:00 ECS 125 | 35% |

# Questionnaires

- Good for large groups when you have specific questions
- **NOT** appropriate as the only way, they suffer from being a *one way* communication technique and suffer *time lag*:
  - You cannot do follow up questions
  - You cannot follow users down interesting paths/points that they might raise during their answer

# Good Usages of Questionnaires
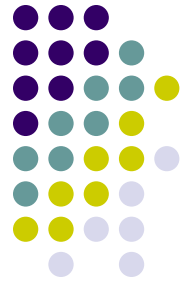
- Good uses of questioners include:
  - Survey of currently used features
  - Reasons for not using specific features
- Questionnaire can be used to establish usability requirements and priorities of features
- You should use interviews to create useful questionnaire for example:
  - "What new features would you like to see?" Open ended and needed... but then you can create a questionnaire and send to users for voting

# Common Questionnaires Mistakes
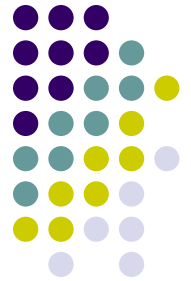
- Bias in sample selection
- Bias in responding users
- Small sample size
- Untested questions that are ambiguous

# Elicitation Techniques

1. Reuse old requirements or existing system
2. Questionnaire
3. Interviews
4. Brainstorming
5. Observation and apprenticeship
6. Ethnographic studies
7. JAD: Joint Application Design
8. Nominal group technique
9. Delphi technique
10. PIECES Approach
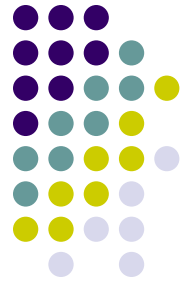
# Interviews

- Key point is to pick the right people to interview!

- Users usually do not know what they want:
  - You built EXACTLY what I asked for!
  - But not what I wanted
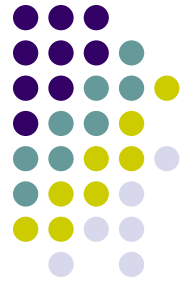
# Interviews

- The people you interview should fill different user/stakeholder roles
- Select users that are:
  - Have authority and motivated to see the project succeed
  - Accountable and knowledgeable
- Interviews should be done in person and not through email/surveys:
  - Observe body language
  - helps build rapport

# Interviews

- Interviews permit us to listen and understand better the problems faced by customers
- Before the interview, research the background of the stakeholder to avoid boring them with questions you easily answer
- You can quickly review some answers if needed
- During the interview, jot down the answers
- Make sure the interview schedule is not overly constraining
- Once rapport is established the interview likely will take a life of its own (horror stories and true/root causes)

# Interviews: Types of Questions

- You should start with open-ended and context free questions then move to more specific questions  as you learn more and gain a better understanding of users needs and requirements:
    - Open Ended: "How will you search for a job?" vs.
    - Context Free: "Will you search using title?"
- Make sure you remove context from question:
    - You would not really give up features and performance just to have it run on a browser - would you?!! (What would the answer be…)
    - Instead ask questions like:
        - How fast should the search be? What kind of performance is required?
        - *Is performance more important than other features like usability?*
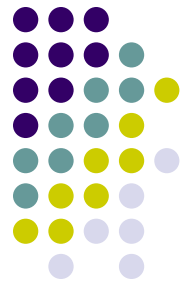
# Examples of Questions

- Example of closed-ended question
  - Would you like the new system to work in your web browser? Y/N (This question does not mention the alternatives and associated costs! What would one give up to get such a feature?)

- Example of open-ended question
  - Would you like the new application be in your web browser even if it means fewer features, and less interactivity?
  - What would you be willing to give up to use your application in your web browser?
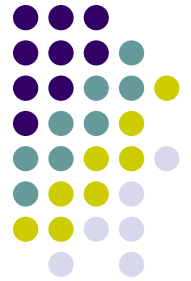
# Template for an Interview

- Establish the Customer/User Profile
  - Name, responsibility
  - *How is success measured?*
  - What problems interfere with success?
- Assessing the Problem
  - *What problems do you face for which you lack a good solution?*
  - Why does the problem exist? How do you solve it? How would like it solved? (Any other)
- Understanding the User Environment
  - *User background*, education, computer systems, expectation for product usability
- Recap for Understanding
  - Repeat to customer in your own words their problems and ask for feedback, clarifications, or additions
  - *Repeat to the customers the answer to the red questions*
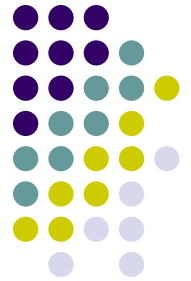
# Template for an Interview

- **The Analyst's Input on the Customers Problems**
  - Validate or invalidate assumptions, suggested problems
  - For each suggested problem, ask the customer to prioritize it in relation to their mentioned problems, and ask for how it is addressed now, how would the customer solve it
- **Assessing Your Solution**
  - Give a rough description of a solution and get users' feedback
- **Assessing the Opportunity**
  - What are the success factors in the user's opinion? Who else uses/would use the application?

# Template for an Interview

- Assessing the Reliability, Performance and Support Needs (Non Functional Requirements)
  - *How robust should the system be?* How long can it go down for? How easy should it be to use? How long should an operation take?
- Other Requirements
- Wrap Up
  - Can I contact you later to check on things?
- The Analyst's Summary
  - Summarize the higher priority needs or problems identified by the user

# After Interviews

- Using the analyst's summary we are likely to get 10-20 top requirements or needs that should be explored in more detail

# Questions to be asked indirectly

- Are you opposed to the system?
- Would you benefit from delaying or obstructing the system?
- Do you feel threatened by the proposed system?
- Is your job threatened by the new system?
- Is anyone else's?

# Common Interviewing Mistakes

- Missing to interview people

- Assuming stated needs are correct (Why?)

- Letting one person in a group interview dominate the interview

- Not conducting group interviews