Lecture 2: User Requirements

CS190: Software System Design

January 25, 2002
Steven P. Reiss

I. Why User Requirements

A. You must know what to build

1. Most common of project failure

a) 13% of projects fail due to incomplete requirements
b) 12% fail due to lack of user involvement
c) 10% fail due to unrealistic expectations
d) 9% fail due to changing requirements
e) 7% fail because the system is no longer needed

2. This has been the difference in past 190 projects

B. This is an ongoing thing

1. You need to get a good first approximation
2. But you can't anticipate all the questions
3. And building the system will create opportunities

a) Different ways of doing things
b) Things that are exceptionally difficult
c) Things that can easily be added

4. Users change their mind a lot

a) The system is a moving target

C. Goals for user requirements

1. Define the problem from the user's point of view
2. Determine outlines of the "best" solution
3. Determine what is required and optional -- priorities
4. Determine limitations on resources
5. Determine acceptance criteria

a) Requirements should be testable
b) Requirements should be as precise as possible

II. Types of Requirements

A. Non-functional

1. **Physical environment**
2. **Interfaces with other systems**
3. **Who will use the system**
4. **What documentation will be required**
5. **What resources will be needed to build, use, maintain**
6. **Security, privacy, etc.**

B. Functional

1. **What will the system do**
2. **When will the system do it**
3. **Resource constraints on system actions**
4. **What does the user interface look like**

III.Forms of Requirements

A. English requirements

1. **Description of the system**
2. **Enumeration of all the different requirements**
   a) Typically ordered hierarchically by type
   b) Typically clustered by functional unit

B. Formal requirements

1. **Model the system in a high-level language**
   a) Show user interaction via state machines
   b) Show process interaction via petri nets
2. **Mathematical modeling**
   a) Z, Larch use logical for describing what the system does
   b) They provide a formal, checkable basis

C. Modeled requirements

1. **Build a model of the system from the user's view**
2. **Data flow models**
   a) What data comes in, what data goes out
   b) How is that data manipulated
3. **Object-based models**

a) Describe the system in terms of objects

b) Describe how the objects interact with each other

c) Describe how the basic operations of the system are done in terms of the objects

D. Use cases

1. A use case describes particular functionality of a system by modeling the dialog that a user, external system, or other entity would have with the system when it is developed

a) Also called scenarios Advantages

b) Easier for customer to determine if the system does what is expected

c) Each use case can be considered separately without a complete understanding of the overall system

d) Can be used to derive features

(1) These can drive system development, especially with multiple releases

(2) Can be used for prioritizing parts of the system

(3) Can be used for tracking development

2. Use case analysis

a) UML provides for use-case diagrams

b) Visual objects represent entities (user, system pieces)

c) Links describe relationships among these

d) Descriptions with the links and diagram provide details

E. Stories

1. This is what is emphasized in XP

2. What they are

a) Short descriptions of the behavior of the system

b) From the user's point of view

3. What they serve as

a) A start for conversation

b) A start for other types of requirements analysis

4. Similar to use cases

IV. Obtaining Requirements

A. Interviewing potential users

1. **Interviews can be structured or unstructured**
2. **Interviews require considerable preparation**
   a) Determine what types of information are needed
   b) Find out about the interviewee
   c) Decide on the questions and organization
3. **Process**
   a) Move from general to specific questions
   b) End with general questions
   c) Record the interview in terms of stories/use cases
4. **Follow up**

B. Questionnaires

1. **Where there is a large set of users**
2. **These are hard to develop as well**
3. **Leave some open ended questions**

C. Conversations

1. **Ideally this should be an ongoing process**
2. **Initial interview yields set of stories/use cases**
   a) These will not fully cover the requirements
   b) These are not even internally complete
3. **You should get back and use them as a starting point for understanding what the user really wants**
4. **You should accept that the requirements will change**
   a) New use cases/stories will be developed
   b) As the user gets a better understanding of what the system can/will do, their ideas will change

D. There is no magic bullet

V. CS190 Approach

A. Try stories as they fit in with XP

B. Combine with other techniques as appropriate

1. Formal methods to describe algorithmic areas
2. Data flow to provide a system overview
3. English description to give an overview to new customers

VI. HOMEWORK

A. Read Chapters 1 & 2 of SE book (Shari)

B. Do exercise 1.16 # 1 (at least think about it).