Requirements Engineering

(Summer 2021)

Prof. Nan Niu (nan.niu@uc.edu)

https://github.com/nanniu/RE-Summer2021

© 2021, Nan Niu

Today's Menu

Thursday (July 15)
Meaning of Req.s
ASN1 Release

Friday (July 16):
Req.s Elicitation
Functional Req.s & NFRs

Monday (July 19):
ASN1 Due (before 9am)
Req.s Modeling
ASN2 Release

© 2021, Nan Niu

Yesterday's take-aways

→ What's the meaning of requirements?

→ Why're the implications of the "meaning of requirements"?

→ What's ASN1 about?

© 2021, Nan Niu

Let's do a quiz on Webex

(participation only)

→ Meaning of requirements

© 2021, Nan Niu

Today's take-aways

→ Why isn't requirements elicitation about "asking the right questions"?

→ What're RE's starting points & how to elicit requirements then?

→ What're functional and nonfunctional requirements?

© 2021, Nan Niu

… because there's no right QUESTION to ask

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

How the customer explained it

How the Project Leader understood it

How the Analyst designed it

How the Programmer wrote it

How the Business Consultant described it

How the project was documented

What operations installed

How the customer was billed

How it was supported

What the customer really needed

… because there's no right ANSWER

How the customer explained it

What the customer really needed

© 2021, Nan Niu

… because there's no right person to ASK

Nan's cell phone in 2009 versus today

"People don't know what they want until you show it to them."

- Steve Jobs
en.wikiquote.org/Steve_Jobs

© 2021, Nan Niu

… because there's no right PERSON to ask

Source: *Google Images*

Scoping decision I

- Decide the scope of the problem:
  - e.g. Bookstore example:
    "Textbooks are often not ordered in time for the start of classes"
  - But that's just a symptom. (So you ask the manager "why?")
    "Because we don't receive the booklists from instructors early enough"
  - Is that just a symptom of some other problem? (…so ask the instructors "why?")
    "Because the instructors aren't allocated to courses early enough"
  - Is that just a symptom of some other problem? (…so ask the UG office "why?")
    "Because we never know who's available to teach until the last minute"
  - Is that just a symptom of some other problem? (…so ask the dept chair "why?")
    "Because there's always uncertainty about who gets hired, sabbaticals, etc."
  - Is that just a symptom of some other problem? (…so ask the dept chair "why?")
    "Because instructors we want to hire don't accept our offers early enough"
  - Is that just a symptom of some other problem? (…so ask the new recruits "why?")
    "Because some other universities seem to wait for ages before making offers"
  - Is that just a symptom of some other problem? (…so ask Ohio State, etc, "why?")
    "Because it takes our department a long time to reach consensus on hiring"
  - Is that just a… …oh wait… …maybe we can develop a decision support system for faculty hiring at Ohio State, and that will help us get our textbooks for the start of class…

© 2021, Nan Niu

How to scope the problem

→ **Difficulty:**

- Every problem can be seen as as symptom of some other (larger) problem
- You can keep on tracing root causes forever if you're not careful

→ **Approach:** (…ask yourself these questions…)

- Is there a reasonable expectation that this problem can be solved?
  (…independently of the larger problem?)
- Is there a reasonable expectation that solving this problem will help?
  (…without also solving the larger problem?)
- Is this a problem that the stakeholders want solved?
  (do the "local experts" think this problem is the one that matters?)
- Is this a problem that someone will pay you to solve?
  (Hint: a feasibility study should quantify the return on investment)

© 2021, Nan Niu

Scoping Decision II

→ Decide the scope of the solution

- Say you decided that *delay in processing booklists from instructors* is the right level of problem to tackle.
  - "So, let's computerize the submission of textbook forms from instructors"
- But while we're at it:
  - "it would help if we also computerized the submission of orders to the publishers"
- …and of course:
  - "we ought to computerize the management of book inventories too, so we can quickly check stock levels before ordering new books"
- …and in that case:
  - "we might as well computerize the archives of past years booklists so that we can predict demand better"
- …and therefore:
  - "it would also make sense to provide a computerized used book exchange, because that has a big effect on demand for new books"
- …and then of course there's … oh, wait, this is going to cost millions!
  - Bookstore manager: "tell me again how this automated used book exchange will help me order books faster?"

© 2021, Nan Niu

How to scope the solution

→ Difficulty:

- We could keep on throwing more technology at the problem forever
- It's hard to decide when to stop adding extra "bells and whistles"

→ Approach (…select among alternatives carefully…)

- Is there a reasonable expectation that this alternative can be implemented?
  (…independently of all the other options?)
- Is there a reasonable expectation that implementing this alternative will (help to) solve the original problem?
  (…without also having to address other aspects of the problem?)
- Is this a solution that the stakeholders can live with?
  (do the "local experts" think they would use all these functions?)
- Is this a solution that someone will pay you to build?
  (Hint: a feasibility study should quantify the return on investment for each alternative)

© 2021, Nan Niu

Starting Points of RE

- Stakeholders
  - If the software (read: RE) fails, who will suffer?
- Boundaries
  - How do you scope the problem?
- Goals and Scenarios
  - A useful way to organize initial collection of information
- Feasibility
  - How to conduct a feasibility study?
  - How to choose which project to pursue?
- Risk
  - Continuous risk management
  - Identifying risks through hazard and fault analysis

© 2021, Nan Niu

Functional Safety Requirements

© 2021, Nan Niu

Source: *Maeder et al. "Strategic Traceability for Safety-Critical Projects", IEEE Software, 2013*

Requirements ≠ Butterflies

requirements elicitation ≠ asking the right questions

→RE's 5 starting points

- Stakeholders
- Boundaries
- Goals and scenarios
- Feasibility
- Risk

Difficulties of Elicitation

- Thin spread of domain knowledge
  - The knowledge might be distributed across many sources
    - It is rarely available in an explicit form (i.e. not written down)
  - There will be conflicts between knowledge from different sources
    - People have conflicting goals
    - People have different understandings of the problem
- Tacit knowledge (The "say-do" problem)
  - People find it hard to describe knowledge they regularly use
    - Descriptions may be inaccurate rationalizations of expert behavior
- Limited observability
  - The problem owners might be too busy solving it using the existing system
  - Presence of an observer may change the problem
    - E.g. the Probe Effect and the Hawthorne Effect

© 2021, Nan Niu

Hawthrone Effect

Source: *Google Images*

Example

- The problem area:
  - Loan approval department in a large bank
  - The analyst is trying to elicit the rules and procedures for approving a loan

- Why this might be difficult:
  - Implicit knowledge:
    - There is no document in which the rules for approving loans are written down
  - Conflicting information:
    - Different members of the department have different ideas about what the rules are
  - Say-do problem:
    - The loan approval process described to you by the loan approval officers is quite different from your observations of what they actually do
  - Probe effect:
    - The loan approval process used by the officers while you are observing is different from the one they normally use

© 2021, Nan Niu

Bias in Requirements Elicitation

→ Bias

- People may not be free to tell you what you need to know
  - Political climate & organizational factors matter
- People may not want to tell you what you need to know
  - The outcome will affect them, so they may try to influence you (hidden agendas)

→ referring to the previous example

- Bias:
  - The loan approval officers fear that your job is to computerize their jobs out of existence, so they are deliberately emphasizing the need for case-by-case discretion (to convince you it has to be done by a human!)

© 2021, Nan Niu

Elicitation Techniques

- Traditional techniques
  - Introspection
  - Reading existing documents
  - Analyzing hard data
  - Interviews
    - Open-ended
    - Structured
  - Surveys / Questionnaires
  - Meetings
- Collaborative techniques
  - Group techniques
    - Focus Groups
    - Brainstorming
  - JAD/RAD workshops
  - Prototyping
  - Participatory Design
- Cognitive techniques
  - Task Analysis
  - Protocol Analysis
  - Knowledge Acquisition Techniques
    - Card Sorting
    - Laddering
    - Repertory Grids
    - Proximity Scaling Techniques
- Contextual approaches
  - Ethnographic Techniques
    - Participant Observation
    - Ethnomethodology
  - Discourse Analysis
    - Conversation Analysis
    - Speech Act Analysis
  - Socio-technical Methods
    - Soft Systems Analysis

© 2021, Nan Niu

Interview as an elicitation technique

Question Tactic	Example
Task-Related	*Could you discuss the steps in…*
Guided	*Could you show me…*
Typical	*Could you tell me about a typical…*
Example	*Can you show me an example of…*
Personal Experience	*Could you tell me about some of your experiences?*
Direct Language	*What do you call…*
Hypothetical-Interaction	*How would you describe…*
Use	*What purpose does this feature serve?*

© 2021, Nan Niu

Source: *Wood, L.E. "Semi-Structured Interviewing for User-Centered Design", Proc. Interactions, 4,2, 1997, pp. 48-61.*

Interview design

Interview 1: Personal Experience Question

What did you like least about the website?

Sample Responses:

"Well, we had to make appointments to be able to connect with each other and stick to it."
"Connection problems with the meeting."
"It was all helpful."
"I can't remember…"

Source: *Cohene, T. and Easterbrook, S.. "Contextual Risk Analysis for Interview Design", RE'05, pp 95-104.*

Better interview design

Interview 1: Personal Experience Question	Interview 2: Hypothetical Experience Question
What did you like least about the website?	Scenario: *John currently participates in face-to-face group support for caretakers. He can either continue to go to his current support group, or try out the videoconference group.* Question: *What would be the benefit of staying in his current support group?*
Sample Responses:	
"Well, we had to make appointments to be able to connect with each other and stick to it." "Connection problems with the meeting." "It was all helpful." "I can't remember…"	"In the early stages of caregiving, face-to-face contact is more advantageous - body language, facial expressions, eyes…" "Sometimes it's hard to put caregiving on pause to use the computer." "Getting out of the house is good." "The technical issues might get in the way of getting support."

Source: *Cohene, T. and Easterbrook, S.. "Contextual Risk Analysis for Interview Design", RE'05, pp 95-104.*

Meetings (Cont'd)

A meeting can result in consensus, but if the consensus is to implement requirements that are inconsistent or have unexpected consequences, little has been gained.

Daniel Jackson

Michael Jackson

Source: *D. Jackson, IEEE SW, 13(2): 20, 22, 1996*

Background Reading

Elicitation Technique Selection: How Do Experts Do It?

Ann M. Hickey
ahickey@uccs.edu

Alan M. Davis
adavis@uccs.edu

Department of Information Systems, The University of Colorado at Colorado Springs

© 2021, Nan Niu

Functional vs. Nonfunctional

- Functional requirements describe WHAT the software does
- Nonfunctional requirements (NFRs) describe HOW WELL the software does it
- Implications: Elicitation, modeling, analysis, realization, validation, evolution … of NFRs are different from those of functional requirements

© 2021, Nan Niu

The *req.s* concerned in Jackson's paper

- The computer must not weigh more than 0.25 Kg.
- The system must be completed by 1st January 1998.
- The programs must be written in Ada.
- The system specification must be formally accepted by the steering committee.
- The operator interface must be easy to learn.
- The system must produce a monthly report of outstanding debts.
- If passenger in the lift presses the *open-doors* button while the lift is stationary at a floor, the doors should begin to open within 0.5 secs.

→ *Functional* requirements

- Real-time response
- Those properties (of operational safety that) can be precisely stated in terms of system behavior

© 2021, Nan Niu

Elevator is the machine

→ Meaning of requirements

- R: "A student carrying a heavy bag to attend a class at a different floor"
- E: "different floor of the SAME building", "the intention to move UP/DOWN", …
- S: "button pressed ➔ request sent to the elevator controller ➔ triggering the motor or opening the door"

© 2021, Nan Niu

It's not just about moving…

Comfort

Safety

Usability

Accessibility

Source: *Google Images*

On NFRs

→ Do we need a (formal) definition (or a complete list) of NFRs?

- System attributes serving as constraints or restrictions on the design of the system [Google'21].
- How the system behaves w.r.t. some observable attributes like performance [Franch'98].
- System qualities: All the 'ilities' [Easterbrook'05].
- Quality is a collection of 7 attributes: reliability, efficiency, usability, portability, testability, understandability, and modifiability [Glass'03].

> "If you want to trigger a hot debate among a group of RE people, just let them talk about NFRs. Although this term has been in use for more than two decades, there is still no consensus about the nature of NFRs and how to document them in requirements specifications."
>
> *Martin Glinz, RE'07*

© 2021, Nan Niu

Class participation time

→ I'll randomly assign you to a breakout room

→ As a group, identify *as many NFRs as you can* (starting with #8):

1. reliability
2. efficiency
3. usability
4. portability
5. testability
6. understandability
7. modifiability

© 2021, Nan Niu

Example NFRs

- Interface requirements
  - how will the new system interface with its environment?
    - User interfaces and "user-friendliness"
    - Interfaces with other systems
- Performance requirements
  - time/space bounds
    - workloads, response time, throughput and available storage space
    - e.g., "the system must handle 1,000 transactions per second"
  - reliability
    - the availability of components
    - integrity of information maintained and supplied to the system
    - e.g., "system must have less than 1hr downtime per three months"
  - security
    - e.g., permissible information flows, or who can do what
  - survivability
    - e.g., system will need to survive fire, natural catastrophes, etc
- Operating requirements
  - physical constraints (size, weight),
  - personnel availability & skill level
  - accessibility for maintenance
  - environmental conditions
  - etc
- Lifecycle requirements
  - "Future-proofing"
    - Maintainability
    - Enhanceability
    - Portability
    - expected market or product lifespan
  - limits on development
    - E.g development time limitations,
    - resource availability
    - methodological standards
    - etc.
- Economic requirements
  - e.g. restrictions on immediate and/or long-term costs

© 2021, Nan Niu

Boehm's NFR list

Source: *See Blum, 1992, p176*

© 2021, Nan Niu

McCall's NFR list

Source: *See van Vliet 2000, pp111-3*

© 2021, Nan Niu

Challenges of NFRs

- Hard to externalize (model)
  - Not directly supported in use cases, class diagrams, ERDs, sequence diagrams, statecharts, and other types of UML models
  - Often contradictory
- Hard to make them measurable
  - You can't control what you can't measure
  - Difficult to evaluate for the customer prior to delivery

© 2021, Nan Niu