Requirements: Functional & User & the Elicitation Process

TOPIC # 5
Chapter 6,7,8 – Karl Wiegers
Chapter 5 & 6 - Reference

Finding Voice of Customer

- Customer involvement is the best way to avoid the expectation gap.
- To find the voice of the customer:
  - Identify the different classes of users for your product.
  - Select and work with individuals who represent each user class and other stakeholder groups.
  - Agree on who the requirements decision makers are for your project.

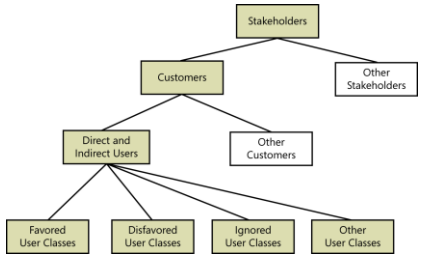
Sources of Requirements

- Various stakeholders
- Pre-existing systems
  - Not necessarily software systems
- Pre-existing documentation
- Competing systems
- Documentation about interfacing systems
- Standards, policies, collective agreements, legislation
- …

Classifying users

- A user class is a subset of product's users.
- A user can be grouped into a number of distinct classes based on these sort of differences:
  - Their access privilege or security levels (such as ordinary user, guest user, administrator)
  - The tasks they perform during their business operations
  - The features they use
  - The frequency with which they use the product
  - Their application domain experience and computer systems expertise
  - The platforms they will be using (desktop PCs, laptop PCs, tablets, smartphones, specialized devices.
  - Their native language
  - Whether they will interact with the system directly or indirectly.

Classifying users



- **Stakeholders:** Clients, customers, users (past and future), buyers, managers, domain experts, developers, marketing and QA people, lawyers, people involved in related systems, anyone who can bring added value!

Stakeholders

- Buyer
  - Pay for Software
  - What features is he willing to pay for?
- User
  - of the current system or future systems
  - Experts of the current system & competitive Products
  - Do not neglect interest groups
  - Select users with care
- Domain Experts
  - Expert who knows the work involved

Stakeholder

- Software Engineer
  - Expert who knows the technology and process
  - Determines if the project is technically and economically feasible
- Other
  - Inspectors, Auditors
  - Market research Specialist
  - Lawyers
  - Experts of the system that interacts with the system being built
  - Others

Favored versus Disfavored user classes

- Favored user classes: - more important than others for a particular project and their satisfaction is more aligned with achieving business objectives
  - When resolving conflicts between requirements from different user classes or making priority decisions, favored user classes receive preferential treatment.
- Disfavored use classes: Groups who aren't supposed to use the product for legal, security or safety reasons
  - Features might be build to deliberately make it hard for disfavored users to do things they aren't supposed to do. Examples include access security mechanisms, user privilege levels, antimalware features (for non-human users), and usage logging.

Ignored & other user classes

- Ignored user classes: They will be user classes that might use the product but the product is not built to specifically suit them.
- Indirect User classes: They wont use application themselves, instead accessing its data or services through other applications or through reports.
- Non human user classes: Software agents performing a service on behalf of the human user, such as bots. These agents can scan n/w for info about good and services, assemble custom news feeds, process incoming email, monitor systems and networks, perform data mining.

Identifying user classes

- Ask project sponsor who he expects to use the system.
- Brainstorm as many user classes you can think of.
- Look for groups with similar needs that can be combined to form major user class with several subclasses.
- Analysis models can help in finding user classes e.g. external agents in Context diagram.
- Organization chart
  - Departments that participate in the business process.
  - Departments that are affected by the business process.
  - Departments or role names in which either direct or indirect users might be found.
  - User classes that span multiple departments.
  - Departments that might have an interface to external stakeholders outside the company.

Documenting user classes

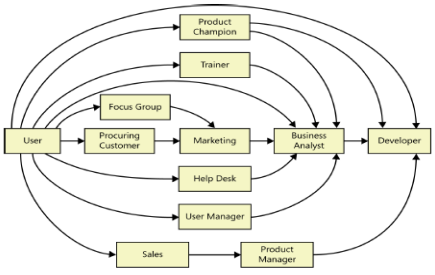
- Document all relevant information about each user class. Such as size, classes favored, volume and type of transactions one will deal with.

User class	Description
Patron (favored)	A Patron is a Process Impact employee who wants to order meals to be delivered from the company cafeteria. There are about 600 potential Patrons, of which 300 are expected to use the COS an average of 5 times per week each. Patrons will sometimes order multiple meals for group events or guests. An estimated 60 percent of orders will be placed using the corporate intranet, with 40 percent of orders being placed from home or by smartphone or tablet apps.
Cafeteria Staff	The Process Impact cafeteria employs about 20 Cafeteria Staff who will receive orders from the COS, prepare meals, package them for delivery, and request delivery. Most of the Cafeteria Staff will need training in the use of the hardware and software for the COS.
Menu Manager	The Menu Manager is a cafeteria employee who establishes and maintains daily menus of the food items available from the cafeteria. Some menu items may not be available for delivery. The Menu Manager will also define the cafeteria's daily specials. The Menu Manager will need to edit existing menus periodically.
Meal Deliverer	As the Cafeteria Staff prepare orders for delivery, they will issue delivery requests to a Meal Deliverer's smartphone. The Meal Deliverer will pick up the food and deliver it to the Patron. A Meal Deliverer's other interactions with the COS will be to confirm that a meal was (or was not) delivered.

User Personas

- User classes are brought to life by creating a *persona* for each one.
- Persona is a description of a representative member of the user class.
- It is a description of a hypothetical, generic person who serves as a stand-in for a group of users having similar characteristics and needs.

Connecting with user representatives



Some possible communication pathways between the user and the developer.

Product Champions

- Product champions are key members of user community to provide the requirements.
- Product champion serves as the primary interface between members of a single user class and the business analyst.
- Ideally, the champions will be actual users, not surrogates.
- They gather requirements from other members of user classes they represent and reconcile inconsistencies.
- External Product Champions: For commercial software companies have to rely on internal subject matter experts or outside consultants to serve as surrogates for actual users.
  - Might be difficult to engage or unknown and might require some economic incentive for participation.
  - Or hire a suitable product champion who has the right background.
  - Watch out difference between current needs and champion's perception

Product Champions

- One person can hardly describe the needs for all users of an application.
- Hence multiple product champions are required each corresponding to different user class.
- Product champion model works only when the product champions understand and sign up for their responsibilities, have the authority to make decisions at the user requirements level, and have time available to do the job.
- **Problems:**
- Managers override the decisions that a qualified and duly authorized product champion makes.
- A product champion who forgets that he is representing other customers and presents only his own requirements won't do a good job.

Product Champions traps

- A product champion who lacks a clear vision of the new system might defer decisions to the BA.
- A senior user might nominate a less experienced user as champion because she doesn't have time to do the job herself.
- Beware of users who purport to speak for a user class to which they do not belong.

User representation on Agile projects

- Some agile development methods include a single representative of stakeholders called a *product owner* in the team to serve as the voice of the customer.
- The product owner defines the product's vision and is responsible for developing and prioritizing the contents of the product backlog.
- The product owner therefore spans all three levels of requirements: business, user, and functional.
- He essentially includes the product champion and business analyst functions, representing the customer, defining product features, prioritizing them, and so forth.

Resolving conflicting requirements

Suggestions for resolving requirements disputes

Disagreement between	How to resolve
Individual users	Product champion or product owner decides
User classes	Favored user class gets preference
Market segments	Segment with greatest impact on business success gets preference
Corporate customers	Business objectives dictate direction
Users and user managers	Product owner or product champion for the user class decides
Development and customers	Customers get preference, but in alignment with business objectives
Development and marketing	Marketing gets preference

Requirements elicitation

What is Requirement elicitation?

- Requirements elicitation is "the process of discovering the requirements for a system by communicating with customers, system users and others who have a stake in the system development"
- Elicitation is a collaborative and analytical process that includes activities to collect, discover, extract, and define requirements.
- Elicitation is used to discover business, user, functional, and nonfunctional requirements, along with other types of information.

Elicitation Risks and Challenges

- You need to extract information from the brain of your customer without damaging the customer, much less his brain!
  - *Good technology and good tools can help, but cannot substitute for adequate social interaction!
- Problems of scope
- Problems of understanding
- Problems of volatility
- Other typical issues
  - Experts seldom available
  - Common vocabulary often missing
- Requirements do not fall from the sky!
- Participants often lack motivation and resist to change
- Much effort and discussion is needed to come up with a common agreement and understanding!

Requirement Elicitation Techniques

- Analysis of Existing Systems
  - Documentation, Observation, and Ethnography
- Interviews, workshops and focus groups
- Questionnaires
- Brainstorming
- Joint Application Design (JAD)
- Prototyping
- Use Cases
- Research & Site visit

Interviews

- Interview as many stakeholders as possible
- Ask problem oriented questions rather than solution oriented.
- Three main objectives:
  - Record information to be used as input to requirements analysis and modeling
  - Discover information from interviewee accurately and efficiently
  - Reassure interviewee that his/her understanding of the topic has been explored, listened to, and valued
- Process consists of four important steps:
  - Planning and preparation
  - Interview session
  - Consolidation of information
  - Follow-up

Interviews

- Establish rapport:
  - introduce yourself i, review the agenda, remind attendees of the session objectives, and address any preliminary questions or concerns attendees have.
- Stay in scope
- Prepare questions and straw man models ahead of time
- Suggest ideas
- Listen actively
- Organize the environment for conducting an effective interview
- Interview several people at once to create synergy

Interviews

- Revise and complete the elicitation notes after the interview
- Identify inconsistencies and address them in a follow-up interview or by email
- Keep all diagrams, charts, models created during the discussions
- You are learning, so be precise and learn business language

Workshops

- Encourage stakeholder collaboration in defining requirements.
- "A structured meeting in which a carefully selected group of stakeholders and content experts work together to define, create, refine, and reach closure on deliverables (*such as models and documents) that represent user requirements."
- Workshops are facilitated sessions with multiple stakeholders and formal roles, such as a facilitator and a scribe.
- Workshops often include several types of stakeholders, from users to developers to testers.
- They are used to elicit requirements from multiple stakeholders concurrently.
- Working in a group is more effective for resolving disagreements than is talking to people individually.
- Also, workshops are helpful when quick elicitation turnaround is needed because of schedule constraints.

Joint Application Design (JAD)

- A more structured and intensive brainstorming approach
- Several activities and six (human) roles to be played
- JAD session may last few days
- Effective use of group dynamics
- Use of visual aids
- Defined process
- Standardized forms for documenting results

Joint Application Design – Activities

- Preparation
  - Pre-session Planning
  - Pre-work
- Working Session
- Summary
  - Follow-up
  - Wrap-up

Joint Application Design – Pre-session Planning

- Preparation is essential – this is not an informal session
- Evaluate project
- Select JAD participants
- Create preliminary agenda
- Determine deliverables for the working session
- Enable participants to prepare for the session

The 6 "P"s

1. Purpose - Why do we do things?
2. Participants - Who is involved?
3. Principles - How do we function?
4. Products - What do we create?
5. Place - Where is it located?
6. Process - When do we do what?

Joint Application Design – Pre-work

- Gather information
- Clear schedules for the working session
- Refine session agenda
- Finalize pre-session assignments
- Prepare material for session (flip-charts, presentations, markers, pizza...)

Joint Application Design – Working Session

- Set-up stage
- Generate common understanding
- Achieve consensus on decisions
- Generate ownership of results
- Create the deliverables (using standard JAD forms)
- Identify open issues and questions

Joint Application Design – Follow-up and Wrap-up

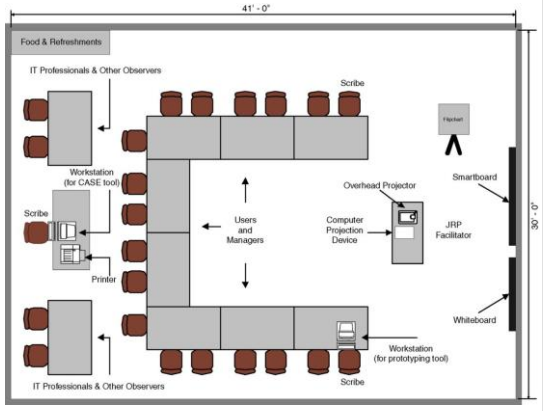
- Follow-up
  - Resolve open issues and questions
  - Follow-up on action items
  - Re-evaluate project
- Wrap-up
  - Review results of follow-up items
  - Evaluate the JAD process
  - Discuss "lessons learned"
  - Finalize deliverables

Joint Application Design – Roles

- Session leader
  - Organizer, facilitator, JAD expert
- Analyst
  - Scribe++
  - Produces official JAD documents, experienced developer who understands the big picture, good philosopher/writer/organizer
- Executive sponsor
  - Manager who has ultimate responsibility for product being built

Joint Application Design – Roles

- User representatives
  - Selection of knowledgeable end-users and managers
- Information system representatives
  - Technical expert on ISs
- Specialists
  - Technical expert on particular narrow topics,



Guidelines for Conducting a JRP Session

- Do not unreasonably deviate from the agenda
- Stay on schedule/Time box discussions
- Ensure that the scribe is able to take notes
- Use parking lots to capture items for later considerations
- Apply conflict resolution skills
- Allow for ample breaks
- Encourage group consensus/ keep everyone engaged
- Encourage user and management participation without allowing individuals to dominate the session
- Make sure that attendees abide by the established ground rules for the session
- Keep team small but include right stakeholders

FOCUS GROUP

- A focus group is a representative group of users who assemble in a facilitated elicitation activity to generate input and ideas on a product's functional and quality requirements.
- Focus group sessions must be interactive, allowing all users a chance to voice their thoughts.
- Focus groups are useful for exploring users' attitudes, impressions, preferences, and needs.
- They are particularly valuable if you are developing commercial products and don't have ready access to end users within your company.
- Must be facilitated.

Questionnaires

- Questionnaires are a way to survey large groups of users to understand their needs.
- They are inexpensive, making them a logical choice for eliciting information from large user populations, and they can be administered easily across geographical boundaries.
- The analyzed results of questionnaires can be used as an input to other elicitation techniques.
- **Tips for developing questionnaire**
  - Provide answer options that cover the full set of possible responses.
  - Make answer choices both mutually exclusive and exhaustive
  - Don't phrase a question in a way that implies a "correct" answer.

Questionnaires tips

- Consistent throughout. If one uses scale they must be consistent.
- Use closed questions with two or more specific choices if you want to use the questionnaire results for statistical analysis. Open-ended questions allows users to respond any way they want, so it's hard to look for commonalities in the results.
- Always test a questionnaire before distributing it.
- Don't ask too many questions or people won't respond.

Observations

- Learning by observing how users perform their tasks.
- Time consuming.
- Important to limit time, select multiple user classes for observations, select important tasks.
- Limit the user to demonstrate the tasks related to forthcoming iterations if using observation on agile projects.
- Observations can be silent or interactive.
- Can be Living the system
- Can Ask user to explain everything he or she is doing

Ethnography

- Ethnography attempts to discover social, human, and political factors, which may also impact requirements.
- Essentially seeks to explore the human factors and social organization of activities → understand work
  - Studies have shown that work is often richer and more complex than is suggested by simple models derived from interviews
- Social scientists are trained in observation and work analysis
- Discoveries are made by observation and analysis, workers are not asked to explain what they do
  - Collect what is ordinary/what is it that people do (aim at making the implicit explicit)
  - Study the context of work and watch work being done

Existing system & documentation analysis

- Start with reading available documentation
  - User documents (manual, guides…)
  - Business processes
  - *Development documents
  - *Requirements documents
  - Internal memos etc.
  - *Change histories
- Of course, often these are out of date, poorly written, wrong, etc., but it's a good starting point
- Discourse analysis
  - Use of words and phrases is examined in written or spoken language
- User Interface analysis) & Systems interface analysis

Research & Site Visits

- Research
- Benchmark other systems
- Websites
- Papers & Journals

Prototyping

- A software requirements prototype is a mock-up or partial implementation of a software system
  - Helps developers, users, and customers better understand system requirements
  - Helps clarify and complete requirements
  - Helps find new functionalities, discuss usability, and establish priorities
- Prototyping is effective in resolving uncertainties early in the development process
  - Focus prototype development on these uncertain parts
  - Encourages user participation and mutual understanding

Prototyping – Realizations

- Prototypes can take many forms:
  - Paper prototypes
    - Prototype on index card
    - Storyboard
  - Screen mock-ups
  - Interactive prototypes
    - Using high-level languages
    - Using scripting languages
    - Using animation tools
  - Models (executables)
  - Pilot systems

Prototyping – Types

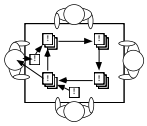
- Horizontal: focus on one layer – e.g., user interface
- Vertical: a slice of the real system
- Evolutive: turned into a product incrementally, gives users a working system more quickly (begins with requirements that are more understood)
- Throw-away: less precise, thrown away, focusing on the less well-understood aspects of the system to design, designed to elicit or validate requirements
  - Fidelity is the extent to which the prototype is real and (especially) reactive (High to low)

Prototyping – Risks

- Prototypes that focus on user-interface tends to lose the focus of demonstrating/exploring functionality
- Prototypes can bring customers' expectations about the degree of completion unrealistically up
- Premature commitment
- Do not end-up considering a throwaway prototype as part of the production system

Brainstorming

- To invent new way of doing things or when much is unknown
  - Early on in a project particularly when:
    - Ground is uncertain
    - There is little expertise for the type of applications
    - Innovation is important
  - Two main activities:
  - The Storm: Generating as many ideas as possible (quantity, not quality) – wild is good!
  - The Calm: Filtering out of ideas (combine, clarify, prioritize, improve…) to keep the best one(s) – may require some voting strategy
- Roles: scribe, moderator (may also provoke), participants



Brainstorming – Objectives

- Hear ideas from everyone, especially unconventional ideas
  - Keep the number of participants "reasonable
- Encourage creativity

Brainstorming – Roles

- Scribe
  - Write down all ideas (may also contribute)
  - May ask clarifying questions during first phase but without criticizing
- Moderator/Leader
  - Cannot be the scribe
  - Two schools of thought: traffic cop or agent provocateur
  - Traffic cop – enforces "rules of order", but does not throw his/her weight around otherwise
  - Agent provocateur – traffic cop plus more of a leadership role, comes prepared with wild ideas and throws them out as discussion wanes

Brainstorming – Participants

- Virtually any stakeholder, e.g.
  - Developers
  - Domain experts
  - End-users
  - Clients
  - …
- "Ideas-people" – a company may have a special team of people
  - Chair or participate in brainstorming sessions
  - Not necessarily further involved with the project

Brainstorming – Eliminating Ideas

- There are some common ways to eliminate some ideas
- Blending ideas
  - Unify similar ideas but be aware not to force fit everything into one idea
- Enforce, Mutual Consensus etc.
- Apply acceptance criteria prepared prior to meeting
  - Eliminate the ideas that do not meet the criteria
- Various ranking or scoring methods
  - Assign points for criteria met, possibly use a weighted formula
- Vote with threshold or campaign speeches
  - Possibly select top k for voting treatment

Planning for Elicitation

Why? Who? When? How? Risks?

Elicitation Plan should include:

- Objectives
- Strategies and processes
- Products of elicitation efforts
- Schedule and resource estimates
- Risks

Elicitation Plan – Objectives / Strategies & Processes

- Objectives: Why this elicitation?
  - Validate market data
- Set elicitation strategies and processes
  - Approaches used
  - Often a combination of approaches depending on the types and number of stakeholders

Elicitation Plan – Products

- Usually set of rough requirements
  - Written, audio, video notes
  - Documentation
- Deliverables depend on objective and technique, e.g.
  - Notes
  - Goals
  - List of use cases, scenarios
  - A set of high-level requirements
  - Detailed Software Requirements Specification (SRS)
  - Analysis of survey results
  - Performance attribute specification
- Generally: un-organized, redundant, incomplete

Elicitation Plan – Estimates

- Identify development and customer participants in various elicitation activities
- Estimate of effort for elicitation
- Scheduling of resources

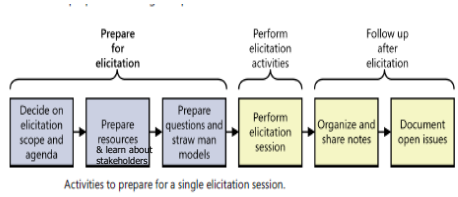
Elicitation Plan – Risks

- Factors that could impede completion of elicitation activities
  - e.g., hostile stakeholders
- Severity of each risk
- Likelihood of occurrence for each risk
- Mitigation strategy for each risk

Examine Project Viability

- Does-it make good business sense ?
  - It's very difficult to cancel a project once started
- Based on:
  - Product's purpose
  - Business advantage
  - Costs vs. benefits
  - Feasibility
  - Scope
  - Required resources
  - Requirements constraints
  - Risks

Elicitation Process



Activities to prepare for a single elicitation session.

Classifying customer inputs

- **Business requirement:** Anything that describes the financial, marketplace, or other business benefit that either customers or the developing organization wish to gain from the product is a business requirement
  - Increase market share in region X by Y percent within Z months."
- **User requirement:** General statements of user goals or business tasks that user need to perform.
  - I need to print a mailing label for a package.
- **Business rules:** Conditions or enforcing something based on rules. Certain users are allowed to do this, must comply with, must be calculated according to etc.
  - A new client must pay 30 percent of the estimated consulting fee and traveling expenses in advance.

Classifying customers input

- **Functional requirement:** Describes the observable behaviors of the system will exhibit under certain conditions and the actions the system will let user take.
  - The user shall be able to sort the project list in forward and reverse alphabetical order
- **Quality attributes:** Statements that describe how well the system does something. Look for words user friendly, secure, reliable etc.
  - "The mobile software must respond quickly to touch commands."
  - "The shopping cart mechanism has to be simple to use so my new customers don't abandon the purchase."
- **External interface requirements:** Describe the connections between the system and the rest of the universe.
  - "The mobile app should send the check image to the bank after I photograph the check I'm depositing."

Classifying customer input

- **Constraints:** Design and implementation constraints legitimately restrict the options available to the developer.
  - "Files submitted electronically must not be exceeded 10MB"
  - The browser must use 256-bit encryption for all secure transactions.
- **Data Requirements:** Customers are presenting a data requirement whenever they describe the format, data type, allowed values, or default value for a data element; the composition of a complex business data structure; or a report to be generated
  - "The ZIP code has five digits, followed by an optional hyphen and four digits that default to 0000."
  - "An order consists of the customer's identity, shipping information, and one or more products, each of which includes the product number, number of units, unit price, and total price."

Classifying customer input

- **Solution ideas:** Someone who describes a specific way to interact with the system to perform some action is suggesting a solution.
  - "Then I select the state where I want to send the package from a drop-down list."
  - "The phone has to allow the user to swipe with a finger to navigate between screens."

Classifying customer input

- Anything that doesn't fit into one of those categories discussed so far might be:
  - A project requirement not related to the software development,
  - A project constraint, such as a cost or schedule restriction
  - An assumption or a dependency.
  - Additional information of a historical, context-setting, or descriptive nature.
  - Extraneous information that does not add value.

Signals for Elicitation completion

- Some clues might be:
  - The users can't think of any more use cases or user stories. Users tend to identify user requirements in sequence of decreasing importance.
  - Users propose new scenarios, but they don't lead to any new functional requirements.
  - Users repeat issues they already covered in previous discussions.
  - Suggested new features, user requirements, or functional requirements are all deemed to be out of scope.
  - Proposed new requirements are all low priority.
  - The users are proposing capabilities that might be included "sometime in the lifetime of the product" rather than "in the specific product we're talking about right now."

Cautions about elicitation

- Balance stakeholder representation *don't only include the louder ones or just few
- Define scope appropriately
- Avoid the requirement versus design argument
- Research within reason

Assumed and implied requirements

- Assumed requirements: What one assumes must be documented. As people might not have same interpretations
- Implied requirements: Developers cant implement functionality they don't know about.

Missing Requirements

- Ways to detect previously undiscovered requirements:
  - Decompose high level requirements into enough detail to reveal what exactly is required.
  - Ensure that all user classes have provided input.
  - Trace system, user, event-response list, business rules to functional requirements
  - Check boundary values and ensure they have been mentioned.
  - Represent requirement info in more than one ways.
  - CRUD and data models also help to find functionalities.

END OF TOPIC # 5

-COMING UP!!!!!!
-Stakeholder Analysis (contd)
-Use cases & User Stories
-Business Rules
-Requirement Analysis & Specification

70