Role of a Business Analyst // What is Business Requirements Modelling?

- References for this lecture
  - Mastering the requirements process – chapter 1
  - BABOK Guide Version 2.0 – chapter 1

Who are stakeholders?
- An individual, team or organisation who have interest in, or participate in the development of requirements and relative software system
- Stakeholders have different roles based on their interest and responsibility in an organisation
- E.g. Project manager, business analyst, sponsor, end user, owner, subject matter expert
- Failure to discover all stakeholders → failure to discover all their needs

What are requirements?
- A condition of capability needed by a stakeholder to solve a business problem, or achieve an objective
- A statement of need that must be met by a particular product or service to solve a business problem or achieve an objective

Requirements Examples
- A customer must be able to place an order for a book on the phone in less than 5 minutes between 9:00 AM and 5:00 PM (Monday to Friday).
- A customer must be able to place an order for a book via a 24/7 online system in less than 3 minutes

Requirement Statement
Requirement = People + Process + Data + Software System + Quality + Assumption

Why Requirements?
- Developing a new or altering an existing business process service or product. E.g. Order management process
- Developing a new or altering an existing software system. E.g. Online order processing system

Requirements Quality (SMART)
- Specific, Simple
- Measureable, Manageable
- Attainable (Achievable, Actionable, Appropriate)
- Realistic (Result orientated, Rationale)
- Time-bound (Timely, Testable, Traceable)

Requirements Activities
- Identified
- Communicated
- Scoped
- Captured
- Prioritised
- De-scoped
- Managed
- Estimated
- Signed Off

What is modelling?
31269_BUSINESS REQUIREMENTS MODELLING– LECTURE 1 – 27TH JULY

- Model
  - A representation of a real world entity or object or subject of interest (TOGAF 9.1) e.g. business process model, data model, software system model
  - AS IS (Current State) and TO BE (Future State) Models

- Modelling
  - is a technique for developing different models e.g., process modelling, data modelling

What is requirements modelling?

- Requirements (modelling)
  - People or stakeholders (week 2)
  - Process (week 4)
  - Data (week 5)

- Object Oriented Modelling (Week 7)
  - Using UML
  - Examples of UML models are Use Case Diagram, Class Diagram, Sequence Diagram, State and Event Diagrams

Why business requirements modelling is important?
The requirements goals
- To create the blueprint of what has to be built using
  - Information technology
  - People
  - Processes
- That is clear, coherent (makes sense) and communicates to all stakeholders

Requirements for a software system
- What does the business need?
  - Improved systems
  - New systems
  - Improved integration between systems
  - Access to external systems

Requirements for a software system – Types of requirements
- Functional requirements (things products must do)
  - The software system should be able to produce a monthly sales report for a given month
- Non-functional requirements (qualities product must have)
  - These include requirements related to capacity, speed, security, availability and the information architecture and presentation of the user interface.
  - Software system should be able to produce a monthly sales report for a given month in less than 5 seconds

Business analyst role
- Tasks
  - Understand what the business does, the business problems, and needs (requirements)
  - Manage stakeholders and communication
  - Develop and communicate business requirements models
  - Review business requirements with stakeholders
  - Recommend software solutions, alternatives and cost estimates
  - Engage in software testing
- Techniques
  - Interviews
  - Requirements workshops
  - Implement requirements modeling as mentioned above
- Skills
  - Analytical thinking and problem solving
  - Business and software knowledge
  - Communication, interaction, and software skills
  - Self management
Requirements Process

- References for this lecture
  - Mastering the requirements process – chapter 2
  - BABOK Guide Version 2.0 – chapters 1 and 2

Topics
- System Development Process
  - Requirements Engineering Process
    - Stakeholders Analysis

Waterfall System Development Process
- The waterfall model, is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards through the phases of
  - Conception
  - Initiation
  - Analysis
  - Design
  - Construction
  - Testing
  - Production
  - Implementation and Maintenance

Agile System Development Process
- Agile software development is a group of software development methods in which requirements and solutions evolve through collaboration between self-organising, cross-functional teams.
- It promotes adaptive planning, evolutionary development, early delivery, continuous improvement and encourages rapid and flexible response to change

Requirements Engineering / Development / Gathering Process
- To find the correct and complete requirements, we need some kind of orderly process. And hence we need “requirements process”
  - Process: A set of steps that software programs goes through when developed
- It is a structure imposed on the development of a software product
- Each phase produces deliverables required by the next phase in the life cycle
Requirements Process

- Requirements process includes
  - Stages/phases
  - Activities and tasks
  - Techniques
  - Tools
- It is used to discover the purpose of the “software” and the needs of the users from the “software” to support their activities
- Requirements process includes
  - Identifying stakeholders
  - Their needs
  - Documenting these requirements in a form of model that is amenable to
    - Analysis
    - Communication
    - Subsequent implementation
- Software system development is driven by “requirements engineering”

Requirements Engineering

- Is the branch of software (systems) engineering concerned with the real-world goals for, functions of and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families
- Def: Refers to the process of defining, documenting and maintaining requirements and to the subfields of systems engineering and software engineering concerned with this process
Requirements process stages:

1. **Requirements Elicitation**
   - Stakeholder analysis
   - Process of seeking, uncovering, acquiring, and elaborating requirements for computer based systems
   - Requirements are elicited rather than just captured or collected. This implies there are discovery, emergence, and development elements to the elicitation process
   - A complex process involving many activities with a variety of available techniques, approaches and tools
   - The relative strengths and weaknesses of these techniques and methods determine when each is appropriate depending on the context and situation
   - Techniques
     - Brain storming
     - Interviews, questionnaires
     - Requirements Workshops
     - Document Analysis
     - Observation

2. **Requirements Analysis and Modeling**
   - Analysing requirements: Determining whether the stated requirements are clear, complete, consistent and unambiguous, and resolving any apparent conflicts
   - Requirements modeling: Developing a set of diagrams known as requirements models, each of which focuses on a different aspect of the users’ needs. E.g.:
     - Business Process Modeling with BPMN
     - Data modeling with ERD
     - Object Oriented Modeling with UML

3. **Requirements Specification**
   - After analysing and modeling the requirements, requirements are specified and documented in a requirements specification document
   - Assumptions and contraints

4. **Requirements Validation**
   - Checking that the documented requirements and models are consistent and meet stakeholder needs

5. **Requirements Management (Continuous Process)**
   - The process of documenting, analysing, tracing, prioritizing and agreeing on requirements and then controlling change and communicating to relevant stakeholders
   - The purpose of requirements management is to ensure that and organisation documents, verifies, and meets the needs and expectations of its customers and internal or external stakeholders
   - Requirements Management Tools:
     - Requirements Backlog
     - Requirements Register / Excel Spreadsheet
     - IBM JazzHub
     - JIRA
   - Requirements Traceability
   - Requirements Communication
   - Change Management

**Who are Stakeholders?**
- An individual, team or organisation who have interest or can influence or be affected by or participate in the development of requirements and relative software system projects
- Stakeholder have different roles based on their interest and responsibility in an organisation
- Failure to discover all stakeholders can mean failure to discover all their needs

**Stakeholder Classes**
- Users
- Sponsor
- Subject matter experts
- Customer
- Developers / Software Engineers
- Project manager

**Stakeholders analysis**
- Who has an interest or can influence or be affected by the changes or requirements?
  - Who is involved
  - Who do we get and communicate the requirements?
  - Who can influence
  - Who will be impacted
  - Who is paying for the project
  - Who will be managing
  - Who will be reviewing
- Stakeholder analysis is the review and consideration of the impact stakeholders have on your business. Companies need to understand the interests of each stakeholder.
- Gaining support from powerful stakeholders can help you to win more resources – this makes it more likely that your projects will have less obstacles and conflicts and hence be successful

**Stakeholders Analysis: Stakeholder Register/Map**

<table>
<thead>
<tr>
<th>Name</th>
<th>Position</th>
<th>Project Role</th>
<th>Contact Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike Sundry</td>
<td>VP of HR</td>
<td>Project Champion</td>
<td><a href="mailto:mundry@globuction.com">mundry@globuction.com</a></td>
</tr>
<tr>
<td>Lucy Cameron</td>
<td>Training Director</td>
<td>Project Sponsor</td>
<td><a href="mailto:camerena@instruction.com">camerena@instruction.com</a></td>
</tr>
<tr>
<td>Ron Ryan</td>
<td>Senior HR staff member</td>
<td>Led the Phase I project</td>
<td><a href="mailto:ryan@globuction.com">ryan@globuction.com</a></td>
</tr>
</tbody>
</table>

**Stakeholders Analysis: Empathy Map**
An empathy map is a collaborative tool to gain a deeper insight into their customers. It is used to understand what it is that motivates the client.
- Helps us consider how other people think and feel – puts us in the shoes of our clients.
- Consists of a simple face, surrounded by 6 sections:
  - Think and feel
  - Hear
  - See
  - Say and do
  - Pain
  - Gain

Some characteristics of user classes
- Frequency of system use
- Subset of system functions used
- Technical expertise
- Security or privilege levels
- Educational level
- Experience of members
- Certain requirements may pertain only to certain user classes
- Requirements of some user classes may be less important to satisfy

What do we want from them?
- What?
  - Understand their problems – what is limiting them?
  - Understand the requirements of the current and future system
  - Understand the scope
  - Understand the requirements priority
- Why
  - Why is this important?
- Important: If it’s not in the requirements, it doesn’t get built.

Constraints and challenges
- CONSTRAINTS: Restrictions or limitations on possible solutions
- The business analyst is responsible for documenting any restrictions or limitations to the solution design, construction, testing, validation and deployment
- Solution constraints describe aspects of the current state, or planned future state that may not be changed. They are no requirements, since they are not implemented in any form by the project team.
- Constraints are provided to the project team to inform them that options they would normally be allowed to consider are not available.

- Project constraints
  - Manpower resources
  - Equipment & facilities
  - Costs
  - Scheduling
- Technical constraints
  - Technology
  - Technical opinions & tradeoffs
- System constraints
  - Priorities of requirements
- Organizational constraints
  - Administrative procedures
  - Responsibilities
- People constraints
Conflicts and challenges
- A conflict may result from stakeholders in different areas, viewing requirements from different perspectives. It may also result from conflicting priorities.
- May be resolved through meetings among affected stakeholders, through research, resolution by a third party, etc.
- Conflicts that affect the requirements must be resolved before formal approval is given to those requirements.

- Why are there conflicts?
  o Empires
  o Power and fear
- Why are there differences between stakeholders?
  o Different views
  o Time-frames
  o Granularity
  o Experiences
- What influences these differences?
  o Communication, openness

- Are there any patterns to these differences?
  o Grouping by function
  o Location
  o Experience
  o Education
- Are there any common goals?
  o Highest common denominator
  o Can common goals be stated and communicated as project objectives?

Summary of Stakeholders
- Stakeholders may have differing or conflicting requirements.
- Not understanding stakeholder differences can lead to poor communications, miscommunication and conflicts and failed software projects.
- Unless there is understanding of what causes the conflicts, it is very difficult to determine appropriate trade-offs.
- It is only then that one can prioritise, re-schedule and define project goals appropriately.
Requirements Process

- References for this lecture
  - Mastering the requirements process – chapter 5
  - BABOK Guide Version 2.0 – chapters 3 and 9

Topics
- Requirements elicitation process
- Types of requirements
  - Functional requirements
  - Non-functional requirements
- Techniques for eliciting requirements
  - Interview, questionnaires, observation, prototyping, requirements workshop

**Negotiation, Prioritising & Management**

- Requirements Elicitation Process
  - Following activities should be included in any requirements elicitation process:
    - Understanding the application domain and the properties of the existing system
    - Identifying the sources of requirements
    - Identifying and analysing all the relevant stakeholders
    - Selecting the approaches, techniques and tools for elicitation
    - Eliciting the requirements from the stakeholders and other sources using the selected techniques, approaches and tools

Present VS Future system
- Get a clear understanding of
  - The overall objectives of the enterprise
  - What do individual users of the system want to achieve in their job
- Understand how
  - The business is operating at present
  - How people are working right now and what they cannot do with the existing system
  - Problems with and inadequacies of the current system
- Hence, discover the “new requirements”
Investigate the current system
- Answering these questions leads to defining the boundary of the new system and its interface
  o Is the existing system a manual one, based on paper documents, forms and files?
  o Is it already computerized?
  o Is it a combination of manual and computerized?
  o If it has evolved over the years, what sections are still usable and what sections do not meet the needs of users?

Requirements types
- Functional requirements (things products must do)
  o They describe the capabilities the system will be able to perform in terms of behaviors or operations – specific information technology application actions or responses
    ▪ Example: the software system should be able to produce monthly sales report for a given month
    ▪ The system shall enable hotel guests to book a room online
- Non-functional requirements (qualities product must have)
  o Describe the environmental conditions under which the solution must remain effective or qualities that the systems must have.
  o These include requirements related to capacity, speed, safety, security, availability and the information architecture and presentation of the user interface.
    ▪ Example: The software system should be able to produce a monthly sales report for a given month in less than 5 seconds and display in on iPad
    ▪ The system shall be able to process 100 payment transactions per second

Requirements Elicitation Techniques

**Interview**
- An interview is a systematic approach designed to elicit information from a person or group of people in an informal or formal setting by talking to an interviewee, asking relevant questions and documenting the responses
- The interviewer formally or informally directs questions to a stakeholder in order to obtain answers that will be used to create formal requirements
- Requires good planning, good interpersonal skills and an alert and responsive frame of mind
- Must ensure that the biases and predispositions of the interviewer do not interfere with a free exchange of information
- Steps:
  o Determine the people to interview ➔ Establish the objectives ➔ Develop questions ➔ Prepare ➔ Conduct ➔ Document ➔ Evaluate the interview
- During the interview:
  o Probe for details by using different types of questions
  o Take thorough notes
  o Identify and document unanswered items or open questions
- After the interview:
  o Thank interviewees ➔ Review and transcribe tapes or notes ASAP (content is still fresh) ➔ Review notes for accuracy, completeness, consistency and understanding ➔ Transfer information to appropriate models and documents ➔ Identify areas needing further clarification ➔ Send interviewees your notes to reflect their comments
- Interview guidelines
- Start with high-level general questions
- Seek lead for more information from stakeholders and share meeting notes and minutes
- Record following up and action items
- Focus on “project” and not the people
- Ask, listen, probe, understand and record (ALPUR)
- Avoid blaming, jargons and forcing your opinion

- Examples
  - Open questions encourage spontaneous and unstructured responses. General questions establishing a viewpoint
    - What do you think of...
    - How relevant is...
    - Is there anything else I should be asking you
  - Closed questions limit or restrict the response requiring specific answer such as a number, explanation of a report, reason for an action
    - Where do you get...
    - How often do you...
    - What do you do with...
  - Probes follow up from a previous answer
    - Why do you...
    - Where do you...
    - How often do you...

Surveys/Questionnaires
- A questionnaire or “survey” is a document containing a number of standard questions that can be sent to individuals to obtain information from a large number of people or when the people are geographically dispersed
- Appropriate for systems that will be used by the general public, where the analyst has to investigate all the types of used of the system.
- Types of questions:
  - Yes/No
  - Multiple Choice
  - Scaled questions – (E.g. Scale from 1 – 5 OR Scale from very satisfied to dissatisfied)
  - Open-ended

Observation
- In this approach, the analyst observes the actual execution of existing processes by the users, usually without direct interference
- Seeing the environment and domain where the system will be situated in action gives additional perspectives and a better understanding of system functionalities
- Through observation, you might discover that neither the system documentation nor the interview statements are accurate
- Observe from within rather than outside
- Observation is essential for gathering some quantitative BUT mostly qualitative data about people’s jobs
- Useful in situations where different interviewees have provided conflicting information about the way the existing system works.

Prototype
A prototype is an initial working model of a larger, more complex entity, usually a program with limited functionality that is built to test out some aspect of how the final system will work.

For requirements elicitation, we often build “throwaway” prototypes, mostly for user interface and in conjunction with developing use cases.

Guidelines:
- Operative: A prototype should be a working model. A simple start will be a “mock-up”
- Focused: To test specific concept or verify an approach, it should be focused on a single objective
- Quick: Tools are needed to that the prototype can be built and modified quickly

Prototyped may be constructed with various objectives in mind
- To investigate user requirements
- To focus on human-computer interface
  - Investigate input and output and its form
  - Investigate most suitable interface
- To determine whether a particular implementation platform can support certain processing requirements
- To determine the efficacy of a particular language, a DBMS, or a communication infrastructure

Paper prototype
- UI flows or storyboarding:
  - Series of drawings used mostly for identifying user interfaces; screens that the software will display are drawn
  - User interface-flow diagrams offer a high-level view of the interface of a system, you can quickly gain an understanding of how the system is expected to work. It puts you in a position where you can validate the overall flow of your application’s user interface
  - Typically used for 2 purposes
    - Model the interactions that users have with your software, as defined in a single use case
    - Enable you to gain a high-level overview of the user interface for your application

Workshop
- Technique used to expedite requirements elicitation, where the objective is to compress all of the activities involved in other fact finding techniques into a shorter series of workshop sessions with users and project team members
- Usually conducted in special rooms with supporting facilities – projector, white board, flip charts etc.
- May be used to scope, discover, define, prioritise and reach closure on requirements for the target system; Generate ideas for new features or products, to reach consensus on a topic of conflicting views
- Organised process: Techniques such as brainstorming, top down analysis
- Documented approach: Output of each session is documented in such a way to make it easy to read, understand and agree on

Guidelines:
- Participants must be selected carefully representing different classes of stakeholders
- Ensure that all stakeholders participate and have their input heard
- Must have a skilled facilitator
- Necessary facilities (visual aids too – flip chats, large screens, etc.) and suitable environment

Advantages and Disadvantages
<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| Interview   | * Allows the interviewer and participant to have full discussions and explanations of the questions and answers.  
* Personal contact allows responsiveness and adapting to what the user says.  
* Analyst can probe in greater depth than any other methods of elicitation.  
* Allows interviewees to express opinions in private that they may be reluctant to express in public. | * Can be time consuming and costly.  
* Requires considerable commitment and involvement of the participants.  
* Interview results have to be transcribed and written, and analysis of interview data can be complex and expensive.  
* Can be subject to bias.  
* If conflicting information is given, it can be difficult to resolve and interviews are not an ideal means of reaching a consensus across a group of stakeholders.  
* Risk of unintentionally leading the interviewee. |
| Questionnaires | * An economical and quick method of gathering data from a large sample.  
* Can reach many people with low resource.  
* Used for answering specific questions.  
* Can be administered remotely.  
* Depending on how well it is designed, the results can be analysed easily by software applications.  
* Effective and efficient when stakeholders are not located in one location. | * Effective questionnaires are hard to design (e.g. leading questions, misinterpretation of questions).  
* Questions are usually not answered completely.  
* The response rates for surveys are often too low for statistical significance.  
* There is no automatic way of follow up unless you do interviews later. |
| Observation | * Provide first hand experience of the way the current system works.  
* Data is collected in real time and can have a high level of validity.  
* Can be used to verify information from other sources or to look for exceptions.  
* Baseline data about the performance of the existing system and of users can be collected. | * Could be very time consuming.  
* Need to analyse huge amounts of data.  
* Most people do not like to be observed and may be disruptive to the person being observed.  
* Requires trained and skilled observer to be most effective.  
* There may be ethical problems if the person being observed deals with sensitive private or personal data or directly with members of public.  
* There may be logistical shifts if the staff being observed work shifts.  
* Unusual exceptions and critical situations that happen infrequently may not occur during the observation. |
| Prototype   | * Allows for early user interaction and feedback.  
* Can be an inexpensive means to quickly uncover and confirm a variety of requirements.  
* Supports users who are more comfortable and effective at articulating their needs by using pictures, as prototyping lets them “see” the future system’s interface. | * Depending on the complexity of the target system, using prototyping to elicit requirements can take considerable time.  
* May lead users to develop unrealistic expectations regarding the delivered systems performance, completion data, reliability and usability characteristics. |
<table>
<thead>
<tr>
<th>Workshop</th>
<th>*Provides a vehicle for designers and developers to learn about the users’ interface needs and to evolve system requirements</th>
<th>This is because an elaborated, detailed prototype can look a lot like a functional system</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*Reduces project development efforts and shortening the schedule</td>
<td>*Risk involved in speeding up the decisions</td>
</tr>
<tr>
<td></td>
<td>*Able to gauge reaction to stimulus material (e.g. storyboards, screenshots)</td>
<td>*Details are inappropriately defined or missed altogether</td>
</tr>
<tr>
<td></td>
<td>*Provides a means for stakeholders to collaborate, make decisions and gain a mutual understanding of requirements</td>
<td>*May suffer from dominance and submission</td>
</tr>
</tbody>
</table>