Requirements Inception

This is Ed shotgun's voice mail. I'm not available... ever.

Finish this project in two weeks.

And make sure you get input from all the executive stakeholders.

That's impossible.

I am either on vacation, or sick, or traveling, or in a meeting.

I do not check e-mail, or return phone calls like the horizon. I am more of a concept than a corporeal being.

Despair is your only option.

Try facing him.
Problem Analysis

- **Goal:** gain a better understanding of the problem being solved before development begins
  - Identify root cause
  - Identify stakeholders and their needs (or problems)
  - Identify solution boundary
- Uses business requirements obtained from stakeholders
- Results in **Product Vision** and **Project Scope**

Business Requirements (1)

- **Business Opportunity**
  - *Description of market opportunity, competing market, business problem being solved or improved, advantage of proposed solution, problems that will be solved...*
- **Business Objective and Success Criteria**
  - *Important business benefits the product will provide in a quantitative and measurable way, how success will be measured, factors that have great impact on success...*
- **Customer or Market Needs**
  - *Problems that customers currently encounter that will be addressed*
- **Business Risks**
  - *Major risks associated with developing or not developing the product (marketplace competition, timing, user acceptance, implementation issues...)*
Business Requirements – Example

■ Business Opportunity
  ● *Exploit the poor security record of a competing product*

■ Business Objective and Success Criteria
  ● *Capture a market share of 80 percent by being recognized as the most secure product in the market through trade journal reviews and consumer surveys*
  ● *Achieve positive cash flow on the product within 6 months*

Business Requirements (2)

■ Important for:
  ● *Ensuring that all project participants work for the same reasons*
  ● *Getting stakeholders agreement on requirements*

■ User and software requirements must align with the context and objective defined by business requirements

■ Requirements that do not help achieve business objective should not be included
Problem Analysis – Five Steps

Five steps for problem analysis (Leffingwell and Widrig)

1. Gain agreement on the problem definition
2. Understand the root causes – the problem behind the problem
3. Identify the stakeholders
4. Define the solution system vision and boundary
5. Identify the constraints to be imposed on the solution

Problem Analysis – Gain Agreement

Document the problem and seek agreement

- Ask stakeholders to write a problem statement in an agreed format
- Statement should include
  - What the problem is
  - Who is affected by it?
  - What is the impact?
  - Is there a proposed solution?
  - What are key benefits?
Problem Analysis – Understand Root Causes (1)

- There is often a problem behind the problem
- Root cause analysis consists of finding underlying causes that may not be immediately apparent
- Example: Our e-commerce site is not profitable
  - Why is it not profitable?
  - Poor site design?
  - Bad pricing?
  - Poor customer management after the sale?
  - Some or all of the above?

Problem Analysis – Understand Root Causes (2)

- Root cause analysis can be used to understand root causes
  - Determine what factors contribute to the problem (sub-problems)
  - Recursively determine what factors contribute to these problems
- Decompose until causes are understood (possible solution clear)
- Decomposition can be represented using a Fishbone diagram, an itemized list...

![Fishbone Diagram Image]
Problem Analysis – Understand Root Causes (3)

- Address Root Causes
  - *Root causes do not all have same impact*
  - *Some may not be worth fixing, at least not now*

- Estimate relative impact of root causes (e.g., with the help of a Pareto (bar) chart)

- Create **problem statement** for root cause problem identified as worth solving (and with computer solution)

![Pareto chart of root causes](image)

Source: Leffingwell & Widrig

Table 5-2  Sales Order Problem Statement

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The problem of…</td>
<td>Inaccuracies in sales orders.</td>
</tr>
<tr>
<td>Affects…</td>
<td>Sales order personnel, customers, manufacturing, shipping, and customer service.</td>
</tr>
<tr>
<td>And results in…</td>
<td>Increased scrap, excessive handling costs, customer dissatisfaction, and decreased profitability.</td>
</tr>
<tr>
<td>Benefits of a solution…</td>
<td>That creates a new system to address the problem include</td>
</tr>
<tr>
<td></td>
<td>- Increased accuracy of sales orders at point of entry</td>
</tr>
<tr>
<td></td>
<td>- Improved reporting of sales data to management</td>
</tr>
<tr>
<td></td>
<td>- Ultimately, higher profitability</td>
</tr>
</tbody>
</table>

Source: Leffingwell & Widrig
Problem Analysis – Stakeholder Profiles (1)

Stakeholder Profiles

- Stakeholders are individuals, groups, organizations who are actively involved in the project, are affected by its outcome or are able to influence its outcome
- Profile should include:
  - Major value or benefit that stakeholder will receive from product (e.g., improved productivity, reduced rework, cost saving, ability to perform new tasks...)
  - Likely attitude toward the product
  - Major features and characteristics of interest
  - Any known constraints that must be accommodated

Problem Analysis – Stakeholder Profiles (2)

- How to identify Stakeholders?
- Elicitation would ask questions such as
  - Who uses the system?
  - Who is the customer?
  - Who is affected by outputs?
  - Who evaluates/approves system?
  - Other external/internal users?
  - Who maintains the system?
  - Anyone who cares? (e.g., legal/regulatory, etc.)
Problem Analysis – Product Vision and Project Scope

- **Product Vision**: describes what the product is about and what it could eventually become
  - Aligns all stakeholders in a common direction
- **Project Scope**: identifies what portion of the ultimate long-term product vision the current project will address
  - Draws boundary between what is in and what is out

---

**Vision Statement (1)**

Vision Statement template (according to Moore)

- **For** [target customer]
- **Who** [statement of the need or opportunity]
- **The** [product name]
- **Is** [a product category]
- **That** [key benefit, compelling reason to buy or use]
- **Unlike** [primary competitive alternative, current system, or current business process],
- **Our product** [statement of primary differentiation and advantages of new product]
Vision Statement (2)

Example

- For scientists who need to request containers of chemicals, the Chemical Tracking System is an information system that will provide a single point of access to the chemical stockroom and vendors. The system will store the location of every chemical container within the company, the quantity of material remaining in it and the complete history of each container's location and usage. This system will save the company 25% on chemical costs in the first year of use by allowing the company to fully exploit chemicals that are already available within the company, dispose of fewer partially used or expired containers and use a single standard chemical purchasing process. Unlike the current manual ordering processes, our product will generate all reports required to comply with government regulations that require the reporting of chemical usage, storage, and disposal.

Problem Analysis – Definition of Scope (1)

- Definition of solution system boundaries
- Requirements baseline is defined according to the release scope
- New requirements during development are evaluated according to the scope
  - New in-scope requirements can be incorporated if they are of high priority relative to the other requirements in the baseline
    - Usually implies deferring or canceling other requirements or negotiating a new schedule
  - Out-of-scope requirements should be deferred to a following release
Problem Analysis – Definition of Scope (2)

Context Diagram

- Top-level view of a system that shows the system’s boundaries and scope
  - Identifies terminators outside the system
  - Data, control, and material flow between terminators and the system

![Context Diagram](image.png)

Problem Analysis – Identify Constraints

Restrictions on the solution space

- Put limitations on the ability to deliver a solution as envisioned
- Usually non-functional requirements that impose major restrictions on the system

Sources of constraints include:

- Economics (e.g., costs, licensing issues)
- Politics (e.g., internal or external, interdepartmental issues)
- Technology (e.g., choice of technology/platform)
- Systems (e.g., existing system, compatibility issues)
- Environment (e.g., legal/environmental/security/standards)
- Schedule and resources (e.g., fixed schedule, team)
Vision and Scope Document (1)

Contents:

- Business requirements
- Vision of the solution
  - Vision statement
  - Major features (numbered list of major features or user capabilities unique to the new product)
  - Assumptions (made while developing vision and scope)
  - Major dependencies to external factors outside of the project’s control (e.g., pending industry standards, government regulations, other projects, third party suppliers, development partners)
- Scope and limitation (for initial and subsequent releases)
- Business context

Vision and Scope Document (2)

Contents – Scope and limitations:

- Concept and range of proposed solution
  - What the system is
- Limitations
  - Capabilities that the product won’t include (what the system is not)
  - Record rejected requirements with the reason for rejecting them
- Scope of initial release
  - Major features planned for initial release
  - Acceptable quality characteristics of initial release
- Scope of subsequent releases
Vision and Scope Document (3)

Contents – Business context:

- Project Priorities
  - Stakeholders must agree on the project priorities
  - Help effective decision making

- Operating Environment
  - Environment in which system will be used (e.g., distributed environment)
  - Vital availability, reliability, performance and integrity requirements linked to operating environment

Requirements Elicitation
Goals, Risks, and Challenges
What is Requirements 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”\(^1\)
- More than a simple request or collection; should evoke and provoke!
- Elicitation means “to bring out, to evoke, to call forth”
- Human activity involving interaction between a diverse array of human beings

---

Elicitation Goals

- Determine sources of information & appropriate techniques
- Get information on domain, problem, constraints
  - \(\rightarrow\) requirements \(\rightarrow\) system development
- Determine the scope and feasibility early
- Produce a first document
  - Mainly user requirements and elicitation notes
  - Potentially incomplete, disorganized, inconsistent
  - But we must start somewhere!
Consider the Following Conversation

- Gerhard, a senior manager at Contoso Pharmaceuticals, was meeting with Cynthia the new manager of Contoso's information systems (IS) development group.
- “We need to build a chemical-tracking information system for Contoso. The system should let us keep track of all the chemical containers we already have in the stockroom and in individual laboratories. That way, maybe the chemists can get what they need for someone down the hall instead of buying a new container from a vendor. This should save us a lot of money. Also, the Health and Safety Department needs to generate some reports on chemical usage for the government. Can your group build this system in time for the first compliance audit five months from now?”
- Are the above requirements? Are they enough to build the system?

Elicitation Risks and Challenges (1)

- 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
  - System boundaries inadequately defined or defined too soon
  - Unnecessary technical details
- Problems of understanding
  - Stakeholder not sure of what is needed
  - Stakeholder has trouble communicating needs
  - Stakeholder does not understand capabilities and limitation of computing environment
Elicitation Risks and Challenges (2)

- Problems of understanding (cont’d)
  - *Stakeholder does not have full understanding of domain*
  - *Stakeholders state conflicting requirements*

- Problems of volatility
  - *Stakeholders will not commit to a set of written requirements*

Elicitation Risks and Challenges (3)

- Other typical issues
  - *Experts seldom available*
  - *Finding an adequate level of precision/detail*
  - *Common vocabulary often missing*

- Requirements do not fall from the sky!
  - *Sometimes hidden*
  - *Sometimes too obvious, implicit, ordinary…*
  - *Assume == “ass” of “u” and “me”*

- Participants often lack motivation and resist to change

- We need much effort and discussion to come up with a common agreement and understanding!
“Ignorance is a bliss”¹

- According to Dan Berry, ignorance of a domain is a good thing!
- Ignorance (not stupidity!) allows one to expose hypotheses and some implicit facts
- Berry even suggests that one day, requirements engineers may advertise their domains of ignorance (rather than their domains of expertise) to find a job!

¹ The Matrix, 1999

RE: More an Art than Science
Sources of Requirements

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

SE502: Software Requirements Engineering
Stakeholder – Customer/Client

- Person who pays for the software development
- Ultimately, has the final word on what will be the product
- For an internal product, the client is probably a product manager
- For the consumer market, the customer may be the marketing department

Stakeholder – Buyer

- Person who pays for the software once it is developed
- Possibly a user or a business owner buying a product for his employees
- What features is he willing to pay for?
  - Which are trivial or excessive?
- Must participate actively in the project (or have a representative)
**Stakeholder – User**

- ... of the current system or future systems
- Experts of the current system: indicate which functions to maintain or improve
- Experts of competitors’ products: suggestions on designing a superior product
- May have special needs or requirements
  - Usability, training, online help ...
- Do not neglect interest groups
  - Expert users, or with disabilities or handicaps
- Select users with care
  - Different seniority
  - Must speak with authority and be responsible and motivated

**Stakeholder – Domain Expert**

- Expert who knows the work involved
- Familiar with the problem that the software must solve. For example:
  - Financial expert for finance management software
  - Aeronautical engineers for air navigation systems
  - Meteorologist for weather forecasting system, etc...
- Also knows the environment in which the product will be used
**Stakeholder – Software Engineer**

- Expert who knows the technology and process
- Determines if the project is technically and economically feasible
- Specifically estimates the cost and time of product development
- Educates the buyer/client on the latest and innovative hardware or software, and recommends new features that will benefit from these technologies

**Stakeholder – Other**

- **Inspector**
  - *An expert in governmental rules and safety relevant to the project*
  - *Examples: safety inspectors, auditors, technical inspectors, government inspectors*
- **Market research specialist**
  - *Can play the role of the customer if the software is developed for the general public*
  - *Expert who has studied the market to determine trends and needs of potential customers*
Stakeholder – Other

- Lawyer
  - Familiar with laws and legal aspects
  - Standards relevant to the project
- Expert of systems that interact with the system to be built
  - Knows the interfaces of the interacting systems
  - May be interested in product features (if the product can help the interacting system to perform its tasks)
- Others that bring added value
  - People who will use your product as a basic building block

On Stakeholders Availability...

- Stakeholders are generally busy!
  - Have priorities other than you
  - Are rarely entirely disconnected from their daily routine and tasks
  - See their participation in the elicitation process as a supplementary task
- Hence, you must have the support and commitment of managers (especially theirs!)
- You must also avoid being perceived as a threat:
  - Loss of jobs caused by the improved system
  - Loss of autonomy, powers, or privileges
  - To the recognition and visibility of their work
Requirements Elicitation Tasks

Tasks Performed as Part of Elicitation (1)

- Planning for the elicitation
- During the elicitation
  - Examine the viability of the project (is it worth it?)
  - Understand the problem from the perspective of each stakeholder
  - Extract the essence of stakeholders’ requirements
  - Invent better ways to do the work of the user
- Following the elicitation
  - Analyse results to understand obtained information
  - Negotiate a coherent set of requirements acceptable by all stakeholders and establish priorities
  - Record results in the requirements specification
Tasks Performed as Part of Elicitation (2)

- Repeat as needed
- Elicitation is incremental
  - Driven by information obtained
  - You always do a bit of elicitation – analysis – specification – verification at the same time

Planning for Elicitation

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
  - Explore usage scenarios
  - Develop a set of requirements, etc..

- Set elicitation strategies and processes
  - Approaches used
  - Often a combination of approaches depending on the types and number of stakeholders
  - Examples: Surveys (questionnaires), workshops, interviews...

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

Project Viability – Purpose & Business Advantage

Purpose
- What is the product? What does the product do?
- Highest-level customer requirement
- Business need
- All other requirements must contribute in some way to the purpose

Business advantage
- Why build the product?
- The purpose of the product should be not only to solve the problem, but also to provide a business advantage
- How will the product help the work?
- A problem can be expressed as a difficulty that customers or users are facing or as an opportunity to produce some benefit, e.g., increased productivity or sales
- Solving the problem leads to software development (or purchase)
Project Viability – Cost vs. Benefits

Cost vs. benefits

- *How much will the product help our work?*
- *How much will it cost to develop and operate the product?*
- Are the expected benefits greater than the anticipated costs?
  - *Demonstrate! If not, stop the project!*

Project Viability – Feasibility

Feasibility

- One reason for describing measurable requirements as soon as possible is to answer questions about feasibility
- Technical feasibility
  - *Does the organization have the skills needed to build and operate the product?*
  - *If not, stop the project*
- Economic feasibility
  - *Does the organization have the resources (time, money, staff) to construct the product?*
  - *If not, stop the project*
Project Viability – Scope (1)

- Scope: product's purpose and the system's boundaries
  - How much of the work will be done by the system-to-be-developed?
  - How much of the work will be done by interacting systems?
- Information needed for cost and time estimates
- Define more precisely the problem to solve
  - List all the things the system should have to do
  - Exclude as much as possible to reduce the complexity of the problem
  - Establish broader goals if the problem is too simple
- Example: an automated system for university registration

![Diagram of initial and reduced scope]

Project Viability – Scope (2)

- The product vision for the kind of product it should be - may affect multiple projects
  - Point of view of customer, business
  - Evolves relatively slowly
- The scope for a single project, defining and communicating clear limits on what to implement
  - Important for the Project Manager
  - More dynamic vision
  - Can be found in a requirements document
- The requirements provide an understanding of what is needed to meet business objectives
  - Many changes!
Project Viability – Required Resources

- Required resources
  - What are the required resources, i.e., money, time, and personnel?
  - How do they compare with available money, time, and personnel?
- If the latter are smaller than the former, we should not even start the project

Project Viability – Constraints

- Requirements Constraints: Are there constraints that will restrict the system’s requirements or how these requirements are elicited?
  - Solution constraints:
    - Mandated designs
    - Mandated interacting systems
    - Mandated COTS (commercial off-the-shelf) components
  - Time constraints
  - Budget constraints
- Warning! Fight to not impose constraints on the process, platform, or language unnecessarily or prematurely!
Project Viability – Risks

Project Risks

- Are there any high-probability or high-impact risks that would make the project infeasible?

- For example:
  - Lack of clear purpose
  - Fragile agreement or disagreement on goals / requirements
  - Unmeasureable requirements,
  - Unstable requirements (rapidly or constantly changing)
  - ...

Understand Problem

- Understand problem from each stakeholder's point of view (only the last few skills do not require social interaction!)
  - Observe current system
  - Interviews
  - Apprenticeship
  - Brainstorming
  - Facilitation
  - Make people open up to you
  - Manage expectations
  - Review documentation
  - Questionnaires
  - Identification of context
  - Detection of ambiguities and noise
Extract Essence

Extract essence of stakeholders' requirements
- Interpret stakeholders' descriptions of requirements
- Possibly build models (may be part of your documentation!)
- Gaps in the model behavior may indicate unknown or ambiguous situations
  - Models help focus our efforts
  - Should be resolved by asking the stakeholders (especially users)

Invent Better Ways

Invent better ways to do the user's work
- Ask why documented requirements so far are desired
- The client/user’s view can be limited by past experiences...
- Consider whether the system should give the user more control over its transactions
- Brainstorm to invent requirements the stakeholders have not yet thought of
Purpose of Vision and Scope Document

- The Vision and Scope Document is an intermediate document during the elicitation process.
- The idea is to do just enough investigation to form a justifiable and rational opinion of overall feasibility and the potential of the new system, and decide whether it is worth investing in further development \(^1\)


Elicitation Problems
Elicitation Problems

According to Dean Leffingwell and Don Widrig:

- The “Yes, But” syndrome stems from human nature and the users inability to experience the software as they might a physical device
- “Undiscovered Ruins”: the more you find, the more you realize still remains
- The “User and Developer” syndrome reflects the profound differences between the two, making communication difficult
- “The sins of your predecessors” syndrome where marketing (user) and developers do not trust each other based on previous interactions, so marketing wants everything and developers commit to nothing

Elicitation Problems – “Yes, But”

- When first time users see the system, the first reaction is either, “wow this is so cool” or “Yes, but, hmmmmm, now that I see it, what about this...? Wouldn’t it be nice...?”
- Users’ reaction is simply human nature
- Need to employ techniques that get the “yes, buts” out early
- Anticipate that there will be “yes, buts” and add time and resources to plan for feedback
- Tends to be user interface centric, tends to be the touch points of the system for the users
Elicitation Problems – “Undiscovered Ruins”

- Teams struggle with determining when they are done with requirements elicitation
  - *Are we done when all the requirements are elicited or have we found at least enough?*
  - *Like asking an archaeologist “how many undiscovered ruins are there?”*
- First scope the requirements elicitation effort by defining the problem or problems that are to be solved with the system
- Employ techniques that help find some of those ruins and have the stakeholders buy-into the requirements

Elicitation Problems – “User and Developer”

- **Characteristic**
- Users do not know what they want, or they know what they want but cannot articulate it
- Users think they know what they want until developers give them what they said they wanted
- Analysts think they understand user problems better than users do
- Everybody believes everybody else is politically motivated
- **Response**
- Recognize and appreciate the user as domain experts; try different techniques
- Provide alternative elicitation techniques earlier; storyboard, role playing, prototypes...
- Put the analyst in the users place; try role playing for an hour or a day
- Yes, its part of human nature, so lets get on with the program
Elicitation Problems – “Living with the Sins...”

- Like it or not, your users (marketing) and developers remember what happened in the past
  - Quality programs that promised things would be different
  - The last project where requirements were vague and/or were delivered short of expectations
  - The team “unilaterally” cut important features out of the last release
- Need to build trust, slowly
- Do not over-commit to features, schedule, or budget
- Build success by delivering highest priority features early in the process

Elicitation Problems – Other Factors (1)

- Social and organizational factors
- "No system is an island unto itself"
  - All software systems exist and are used within a particular context (technical AND social)
  - Social and organizational factors often dominate the system requirements
  - Determining what these are can be difficult and time-consuming
    - Developers are (usually) outsiders
    - People don't always tell the truth
    - Awareness of one's own "culture" can be hard
Elicitation Problems – Other Factors (2)

These factors tend to cut across all aspects of the system:

- e.g., a system that allows senior managers to access information directly without going through middle managers
  - *Interface must be simple enough for senior managers to be able to use*
  - *Middle managers may feel threatened or encroached upon, be resistant to new system*
  - *Lower-level users may concentrate on activities that impress senior managers, which is not necessarily what they ought to be doing*
  - *Users may not like "random spot checks"; may devise ways of hiding what they're doing*