SOFT 423
Software Requirements

Section III
Requirements Elicitation

Review

- Requirement Engineering Process
  - Requirement Development
    - Elicitation, Analysis, Specification and Validation
  - Requirement Management
  - Customer-developer Partnership
  - Good practices

Outline

- Establishing Business Requirements
- Finding the voice of users
- Requirement Elicitation
- Elicitation Techniques

Business Requirements

- A need that leads to one or more projects to deliver a solution and the desired ultimate business outcomes.
- To display the business benefits, vision, and scope
- Benefits represents a true value for sponsors and customers.
  - Increasing revenue and decreasing costs:

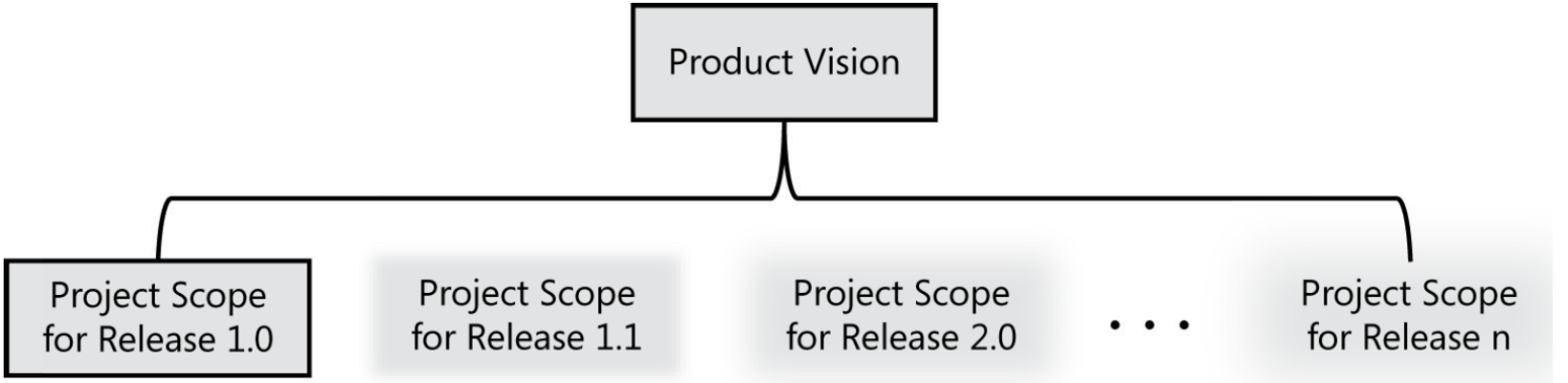
"Reduce monthly support costs from $X to $Y within Z months."

Vision and Scope

- Product Vision: succinctly describe the ultimate product that will achieve the business objectives
  - We all know where we are hoping to go eventually.
- Project Scope: identify what portion of the ultimate product vision that the current project will address
  - We are all talking about the same thing for the immediate project.

Vision and Scope

- The vision applies to the product as a whole
- The scope pertains to a specific release that will implement the next increment of the product's functionality.



Business Requirements Conflict

- Business requirements are from multiple sources
  - funding sponsors
  - corporate executives
  - marketing managers
  - product visionaries
  - …
- Business requirements might conflict !!!

Business Requirements Conflict

- Stakeholders for a kiosk don't always have congruent business interests.

The Kiosk Developers

- Generate revenue by leasing or selling the kiosk to the retailer
- Sell consumables to customers through the kiosk
- Attract retailers to the brand
- Make a wide variety of products or services available

The Retailer

- Maximize revenue from the available floor space
- Attract new customers to the store
- Increase sales to existing customers
- Increase profit margins
- Little kiosk maintenance required

The Customer

- Broad selection of products or services available
- Find desired products quickly
- Spend less time purchasing
- Easy-to-understand purchasing process

Business Requirements Conflict

- Decision makers must resolve these conflicts !!!
- The focus is on delivering the maximum business value to the primary stakeholders.
- The software team cannot resolve conflicts
- RAs can
  - help surface potential areas of conflict and differ assumptions
  - flag conflicting business objectives
  - note when requested features don't achieve those objectives
  - facilitate conflict resolution.

Vision and Scope Document

- Requirement analyst articulates the business requirements and writes the vision and scope document
- Collect the business requirements into a single deliverable

  Input sources of document:
  - customer
  - organization's senior management
  - a product visionary
  - a product manager
  - a subject matter expert
  - marketing department members

Version and Scope Document

- Template

1. Business requirements
- 1.1 Background
- 1.2 Business opportunity
- 1.3 Business objectives
- 1.4 Success metrics
- 1.5 Vision statement
- 1.6 Business risks
- 1.7 Business assumptions and dependencies

2. Scope and limitations
- 2.1 Major features
- 2.2 Scope of initial release
- 2.3 Scope of subsequent releases
- 2.4 Limitations and exclusions

3. Business context
- 3.1 Stakeholder profiles
- 3.2 Project priorities
- 3.3 Deployment considerations

Version and Scope Document

1.1 Background

Summarize the rationale and context for the new product or for changes to an existing one

1.2 Business opportunity

A comparative evaluation of existing products, indicating why the proposed product is attractive and the advantages it provides.

Version and Scope Document

1.3 Business objectives

Summarize the important business benefits the product will provide in a quantitative and measurable way.

Financial:

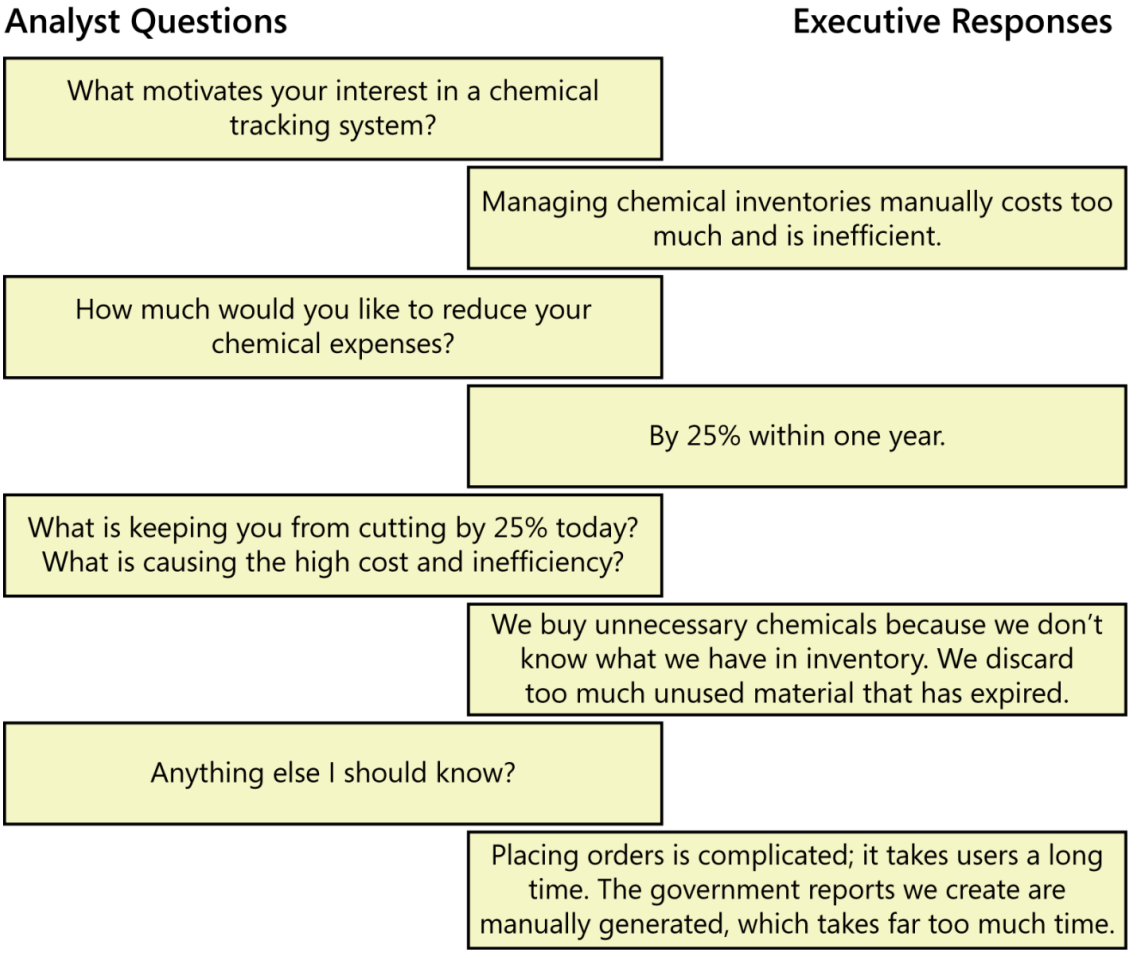
- *Increase market share in country W from X% to Y% within Z months.*
- *Save $X per year currently spent on a high-maintenance legacy system.*

Non-financial:

- *Achieve a customer satisfaction measure of at least X within Y months of release.*
- *Receive no more than X service calls per unit and Y warranty calls per unit within Z months after shipping.*

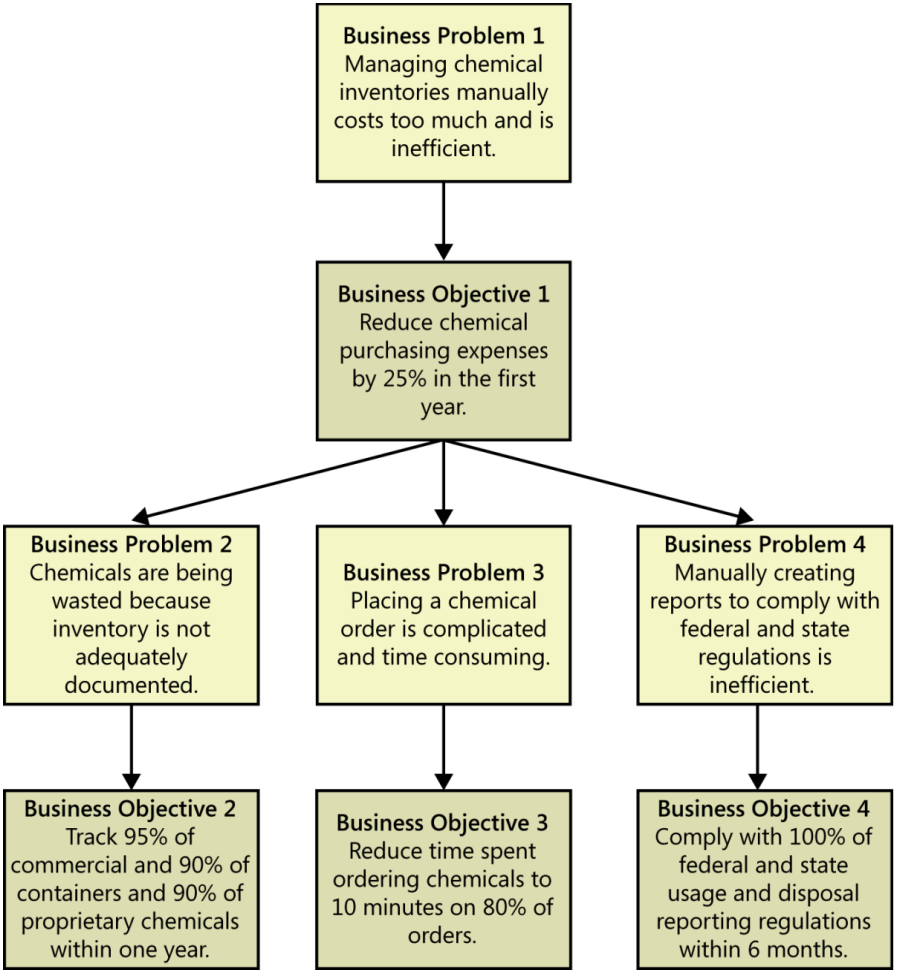
Version and Scope Document

1.3 Business objectives



Version and Scope Document

1.3 Business objectives



Version and Scope Document

1.4 Success metrics

- Specify the indicators that stakeholders will use to define and measure success
- Identify the factors that have the greatest impact on achieving that success and measure the success.

Measurable factor:

- *Increase market share in country W from X% to Y% within Z months.*
- *Receive no more than X service calls per unit and Y warranty calls per unit within Z months after shipping.*

Version and Scope Document

1.5 Vision Statement

- Summarizes the long-term purpose and intent of the product
- A balanced view that will satisfy the expectations of diverse stakeholders

—***For*** [target customer]

—***Who*** [statement of the need or opportunity]

—***The*** [product name]

—***Is*** [product category]

—***That*** [major capabilities, key benefit, compelling reason to buy or use]

—***Unlike*** [primary competitive alternative, current system, current business process]

—***Our product*** [statement of primary differentiation and advantages of new product]

Version and Scope Document

1.6 Business Risks

- Summarize the major business risks
- *Marketplace competition, timing issues, user acceptance, implementation issues.*

1.7 Business assumptions and dependencies

- A statement that is believed to be true in the absence of proof or definitive knowledge
- *Pending industry standards or government regulations, deliverables from other projects, third-party suppliers, or development partners*

Version and Scope Document

2. Scope and limitations

2.1 Major features

- List the product's major features or user capabilities, emphasizing those that distinguish it from previous or competing products.
- Features: logically related system capabilities
  - Provide values to users
  - Are described by a set of functional requirements.
- A feature can encompass multiple user requirements
- Each user requirement leads to certain functional requirements.

Version and Scope Document

2.2 Scope of initial release

- Summarize the capabilities that are planned for inclusion in the initial product release.

2.3 Scope of subsequent releases

- Build a release roadmap that indicates which functionality chunks will be deferred and the desired timing of later releases

2.4 Limitations and exclusions

- *"The new system will not provide mobile platform support."*

Version and Scope Document

3.1 Stakeholder profiles

- Different categories of customers and other key stakeholders.
  - The major value or benefit: cost saving, improved productivity, rework and waste reduction, ability to perform new tasks.
  - Major features and characteristics
  - Their attitudes

3.2 Project priorities

- To determine the priorities
- To adjust the degrees of freedom to achieve the project's success within the limits imposed by the constraints.

Version and Scope Document

3.3 Deployment considerations

- Summarize the information and activities that are needed to ensure an effective deployment of the solution into its operating environment.
  - Describe the access that users will require to use the system (Location and Time zones)
  - Describe infrastructure changes for capacity, network access, data storage
  - Record the knowledge needed for users (Training)

Review

- Business Requirements
  - Product Vision and Project Scope
  - Requirement Conflict
  - Vision and Scope Document
    - Business Requirements
    - Scope and Limitations
    - Business Context
  - Scope Representation Techniques

Scope Representation Techniques

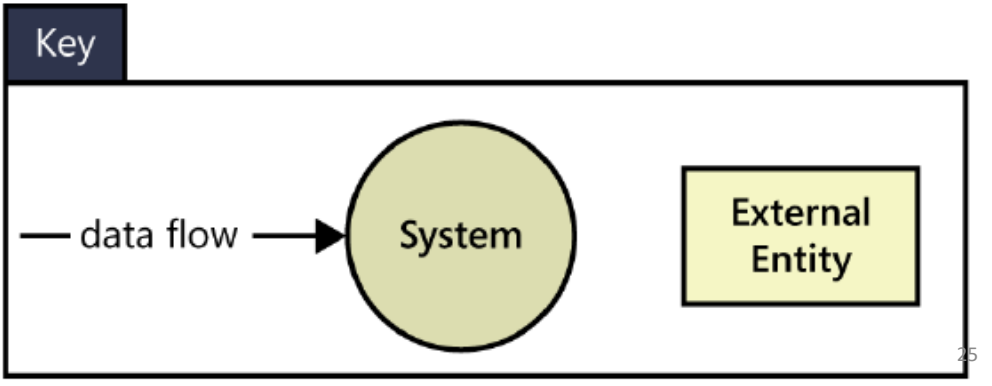
- To represent project scope

Foster clear and accurate communication among the project stakeholders.

- Context Diagram
- Ecosystem Map
- Feature Tree
- Event List

Context Diagram

- Establish the boundary and connections between the system and the environment.
  - The entire system (**a single circle**): no visibility into the system
  - External entities (**rectangles**): user classes, organizations, other systems, or hardware devices
  - Data flow (**arrows**) between the external entities and the system.

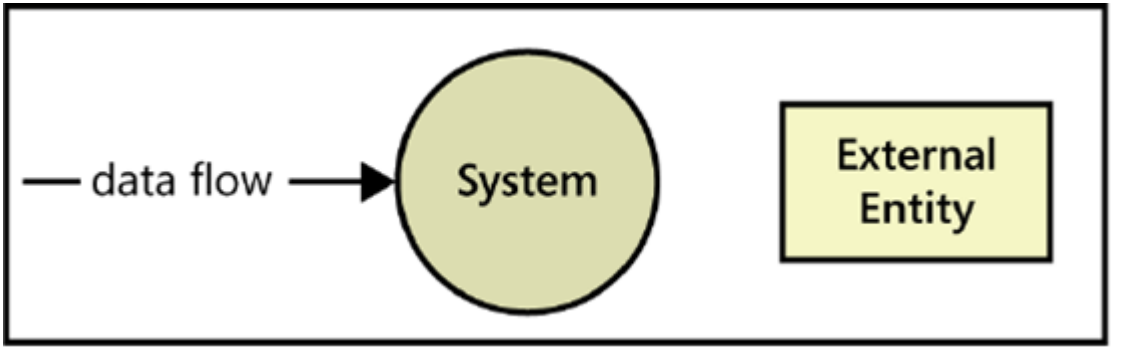


Context Diagram

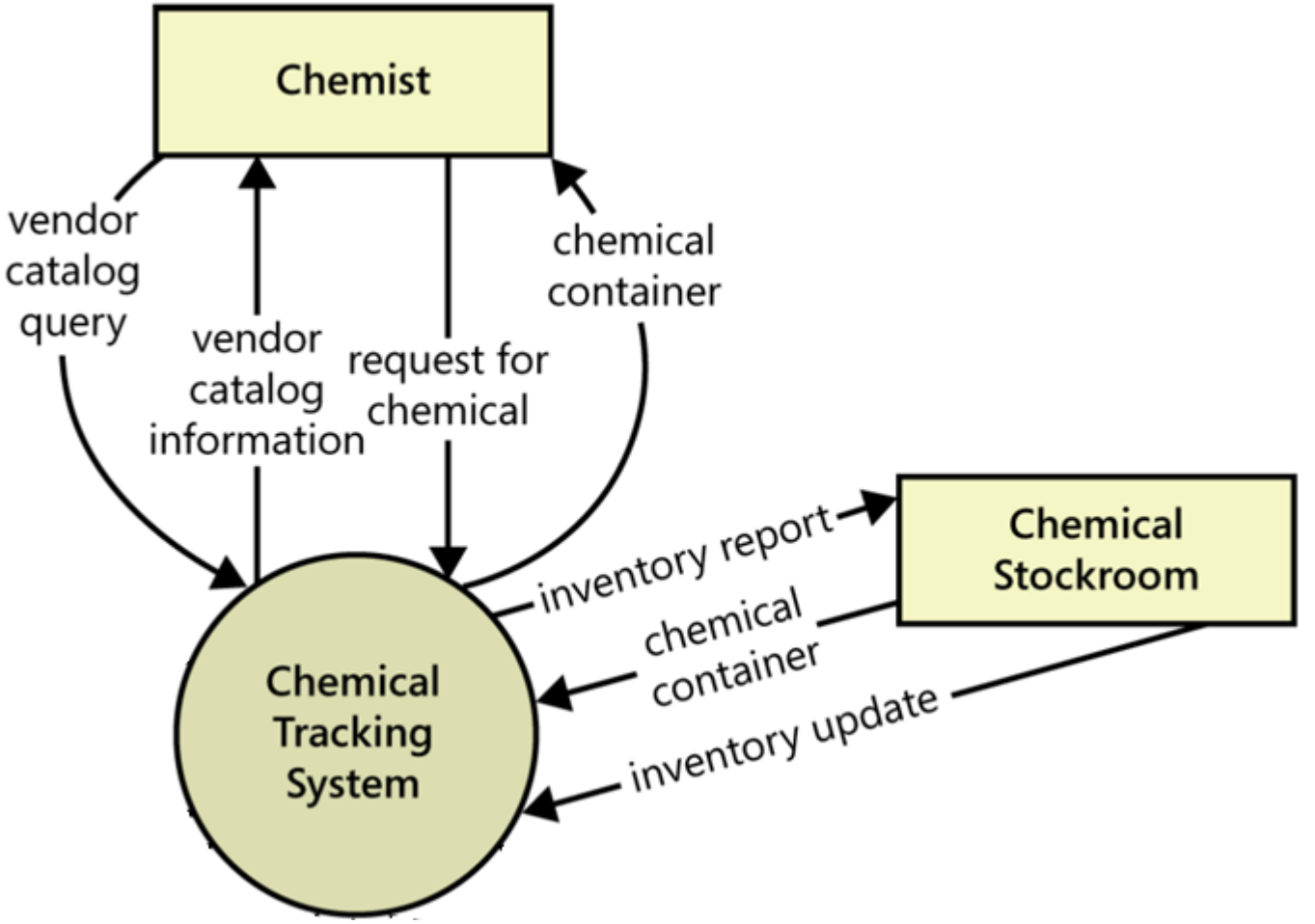
- Chemical Tracking Systems
  - To track the chemical inventories in the stockroom.
  - A chemist searches the chemical catalog and requests the chemicals they need.
  - The chemical stockroom checks the inventory report and updates the inventory in the stockroom.
  - The bar code reader is used to check in or check out the chemical containers.
  - The usage information is reported to the health and safety department.
  - The system can check the training record of the chemists in the training database.
  - The system requests the order status of the chemicals ordered by the buyers in the purchasing department.
  - …

Context Diagram

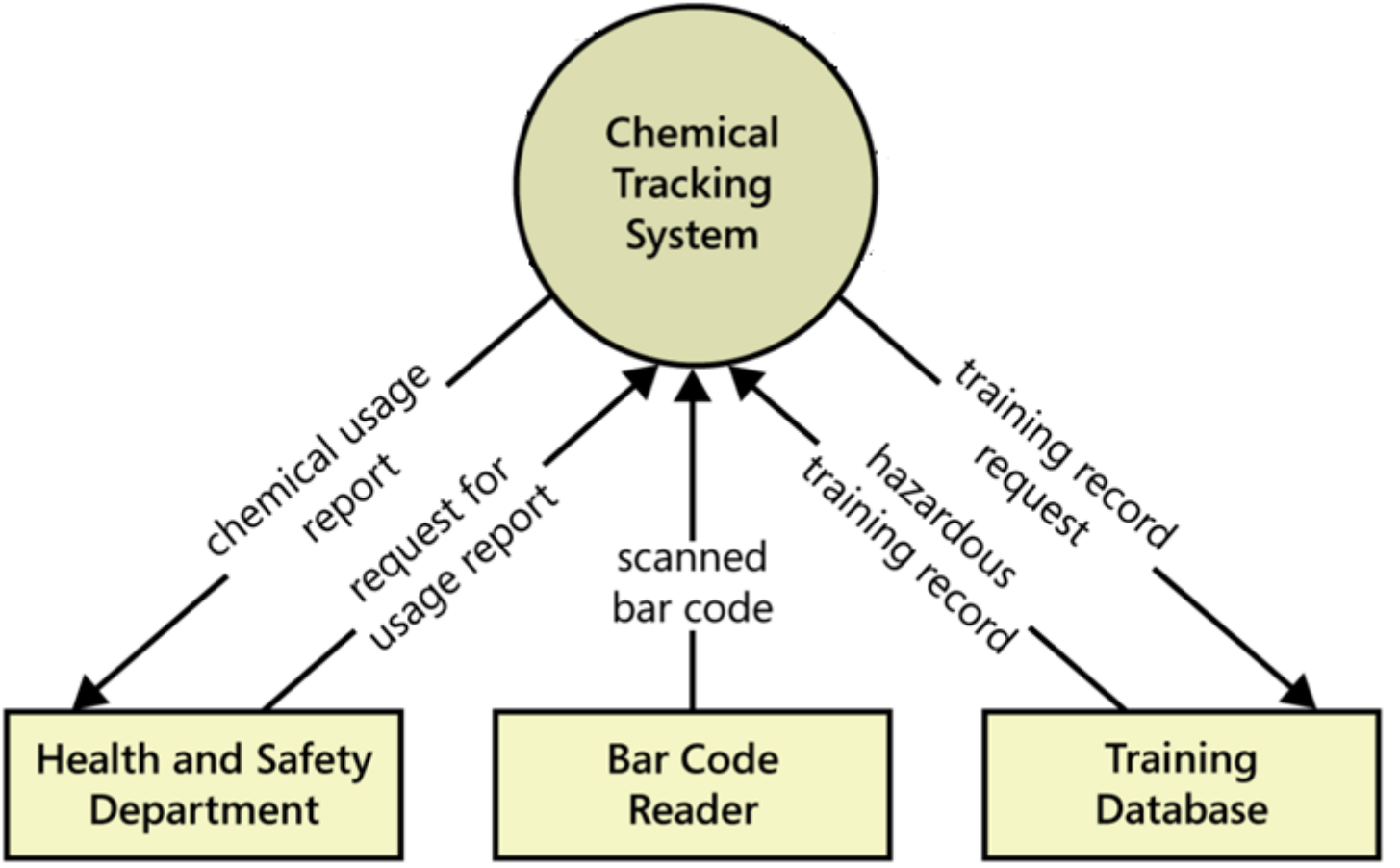
- Chemical Tracking System
  - The entire system (**a single circle**): Chemical Tracking System
  - External entities (**rectangles**): Chemist, Buyer, Chemical Stockroom, Health and Safety Department, Training Database, Bar Code Reader
  - Data flow (**arrows**): Request for chemical, chemical container, vendor catalog query, vender catalog information



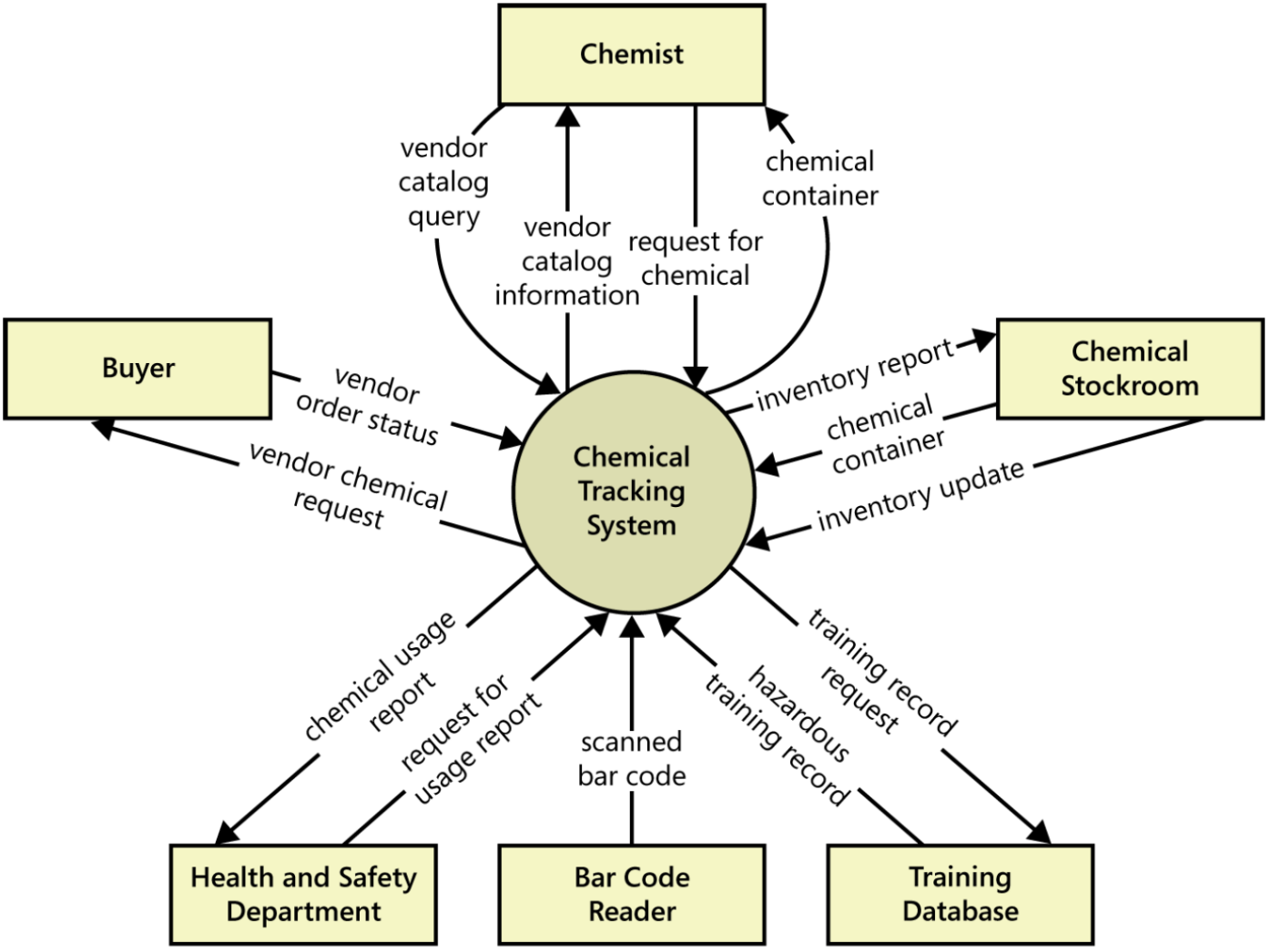
Context Diagram



Context Diagram



Context Diagram

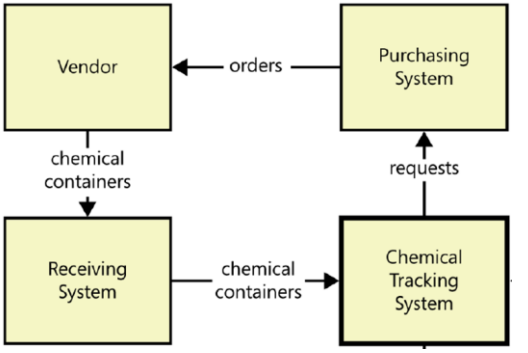


Ecosystem Map

- Show all the systems related to the developing system that interact with one another and the nature of those interactions
  - All the systems that interconnect and that therefore might need to be modified to accommodate the new system
  - Other systems that have a relationship with the system (may not have direct interfaces)

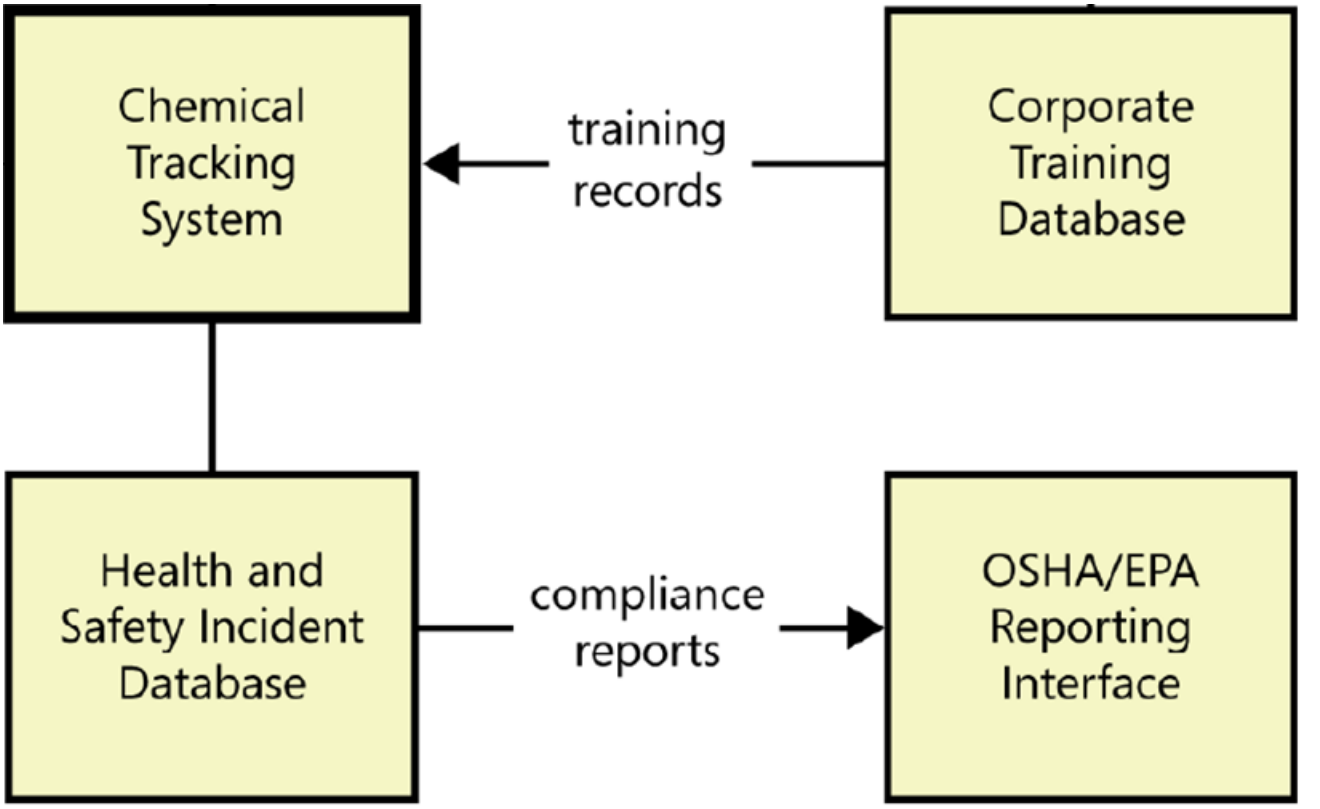
Ecosystem Map

- **A bold box**: the primary system
  Chemical Tracking System
- **Boxes**: other systems that interconnect
  - Purchasing systems
  - Receiving systems
  - Health and safety incident database

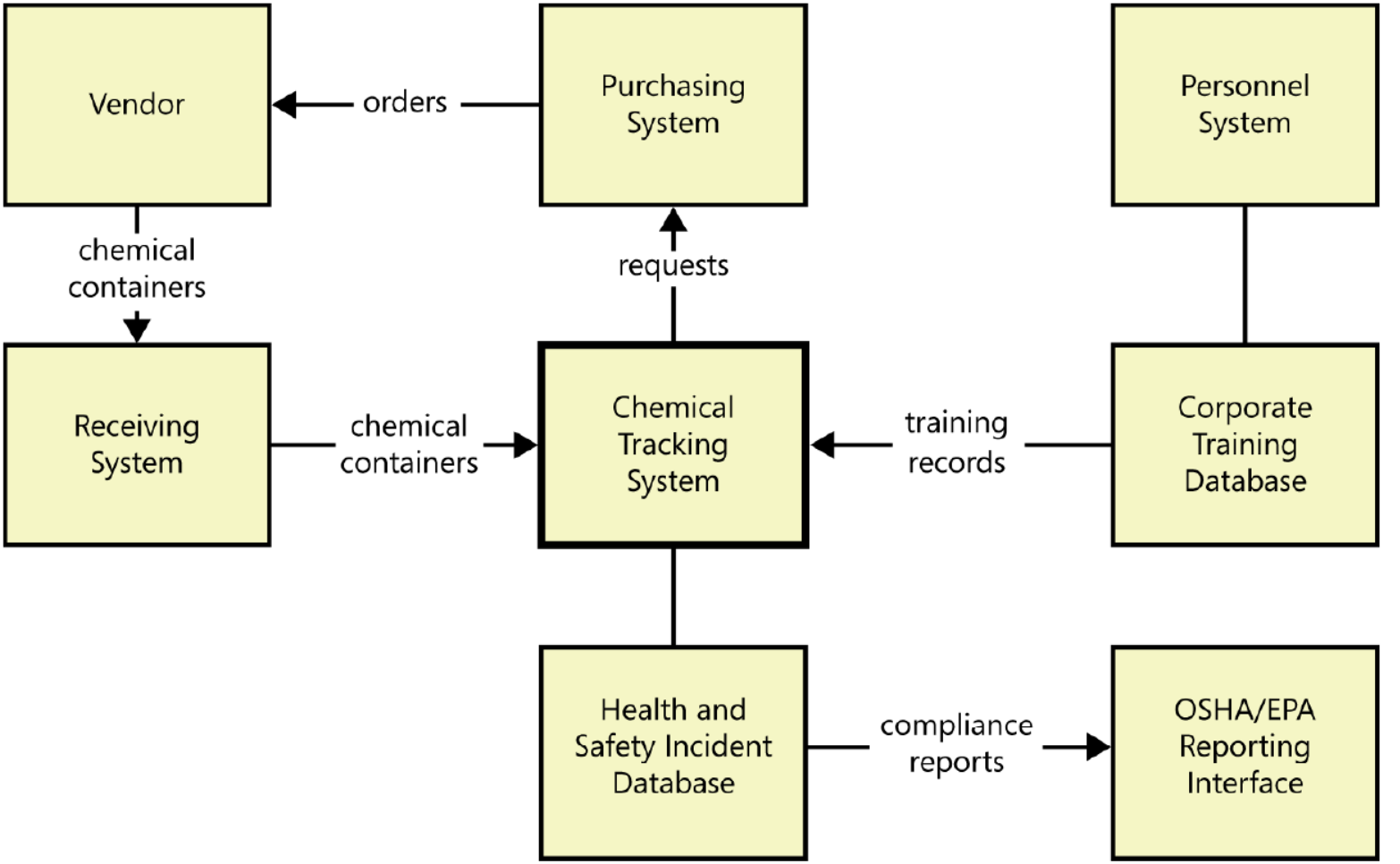


- **Lines**: interfaces between systems
- **Lines with arrows and labels**: data flow from one system to another

Ecosystem Maps



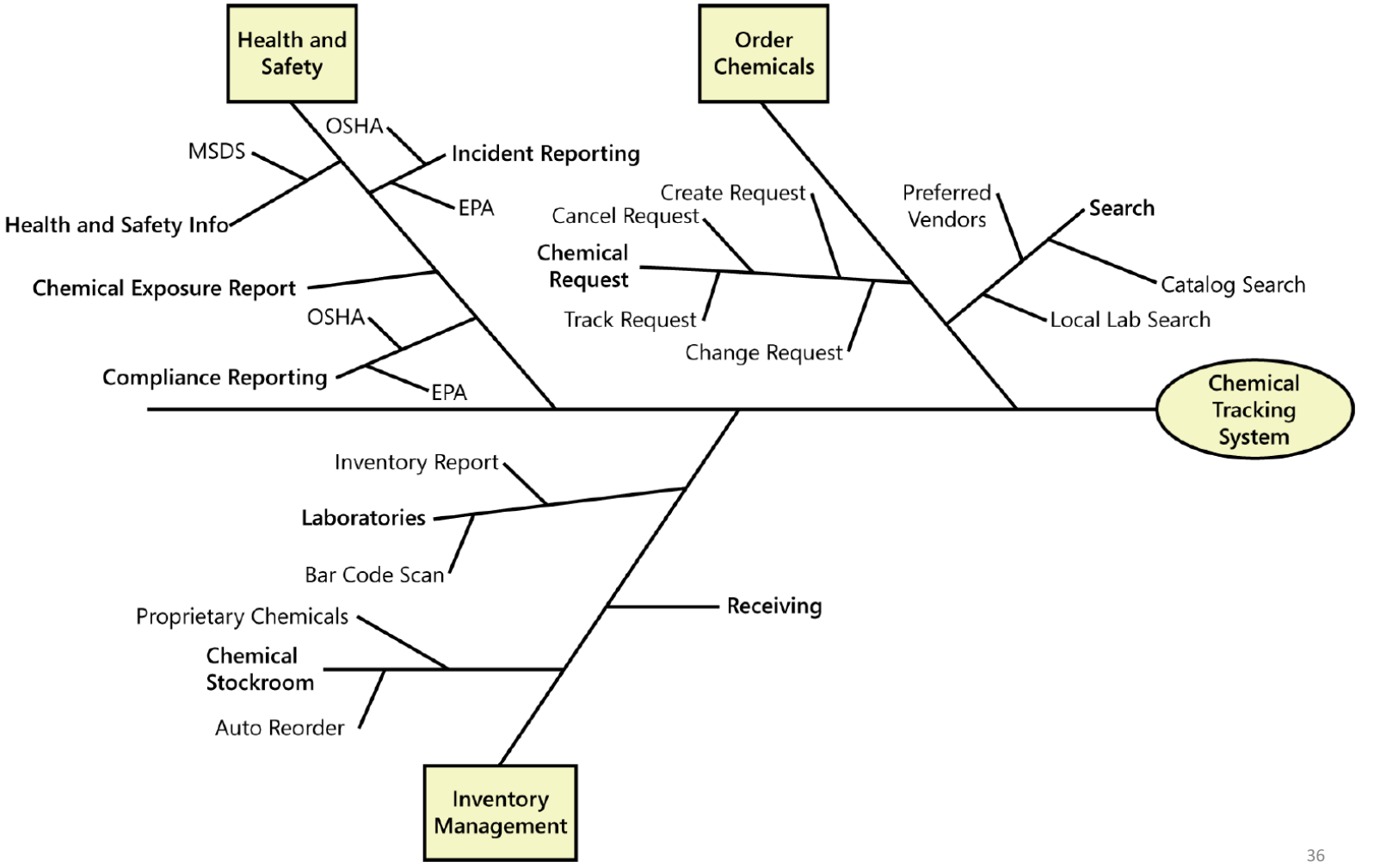
Ecosystem Maps



Feature Tree

- A visual depiction of the product's features organized in logical groups.
- Hierarchically subdivide each feature into further levels of detail
  - A concise view of all of the features planned for a project
  - A feature tree can show up to three levels of features
  - level 1 (L1), level 2 (L2), and level 3 (L3).

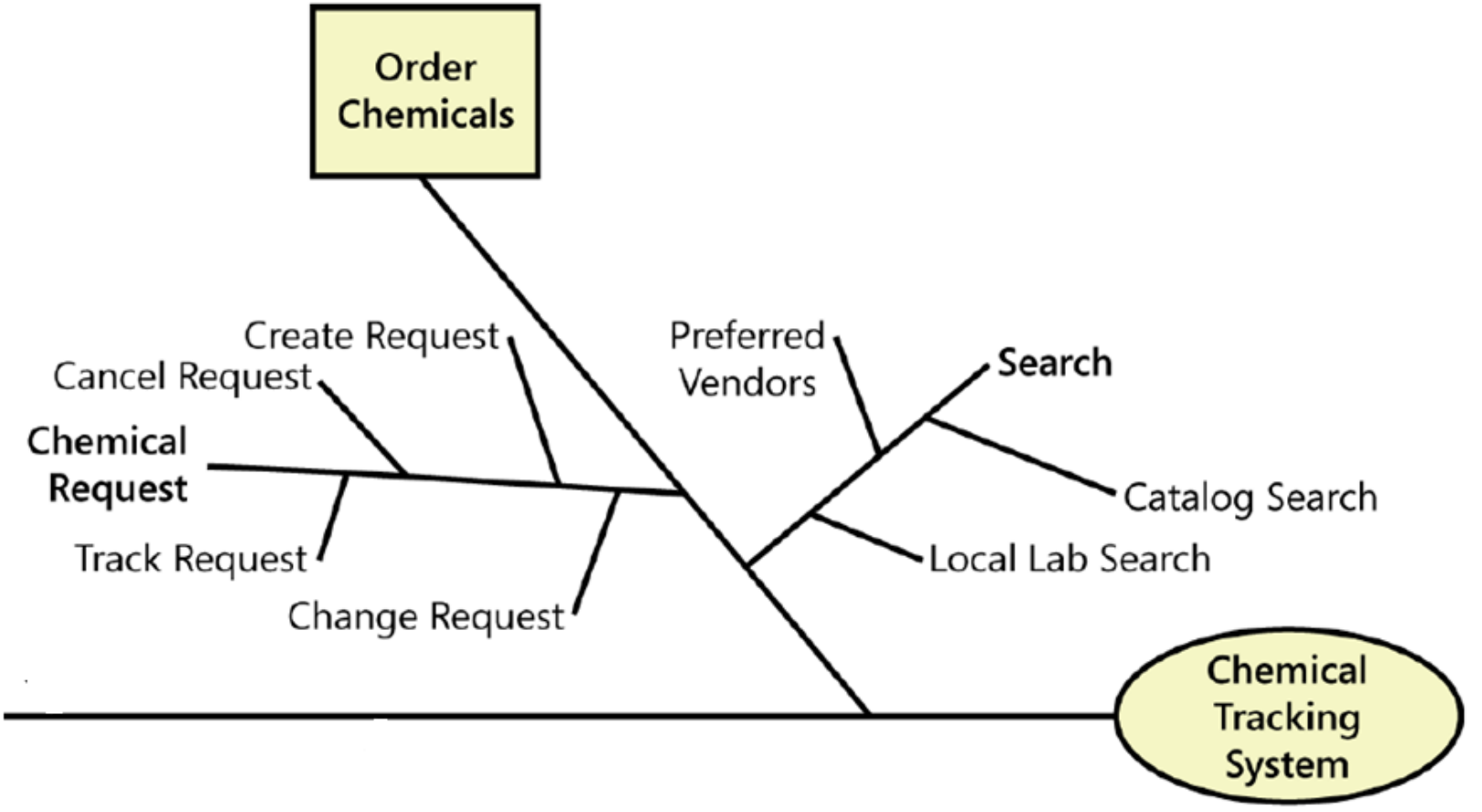
Feature Tree



Feature Tree

- The boxes represent the L1 features
  - Order Chemicals
  - Health and Safety Management
  - Inventory Management
- The lines coming off an L1 branch are L2 features
  - Order Chemicals: Chemical Request, Search
- The branches off an L2 branch are the L3 features
  - Search: Preferred Vendors, Catalog Search, Local Lab Search

Feature Tree



Feature Tree

Define the scope by selecting a specific set of features and sub-features to be implemented

- A feature is implemented entirely in a specific release
- A feature can be implemented in multiple release.
  - implement only a portion of it by choosing certain L2 and L3 sub-features
  - add more L2 and L3 sub-features until each feature is fully implemented.

Event List

- Identify external events that trigger behaviors in system
  - business events triggered by users
  - time-triggered (temporal) events
  - signal events received from external components

External Events for Chemical Tracking System

- Chemist places a chemical request.
- Chemical container bar code is scanned.
- Time to generate OSHA compliance report arrives.
- Vendor issues new chemical catalog.
- New proprietary chemical is accessioned into system.
- Vendor indicates chemical is backordered.
- Chemist asks to generate his chemical exposure report.
- Updated material safety datasheet is received from EPA.
- New vendor is added to preferred vendor list.
- Chemical container is received from vendor.

Event List

- Complement context diagram and ecosystem map
  Context diagram and ecosystem map: external entities and systems involved
  - ➢ Event list: specify the behaviors triggered by the external entities and systems.
    - Whether an external entity on the context diagram is the source of any events
    - Whether any systems in the ecosystem map leads to events for the system.
    - For each event, whether there are corresponding external entities in the context diagram or systems in the ecosystem map

Summary

- Choose scope representation techniques that provide the most useful insight for each project
- Scope change isn't a bad thing
- Determine which features or user requirements have the most value for the business objectives.
- Schedule the important features for the early releases

Review

- Scope Representation Techniques
  - Context Diagram
  - Ecosystem Map
  - Feature Tree
  - Event List

Outline

- Business Requirements
- Finding the voice of users
- Requirement Elicitation
- Elicitation Techniques

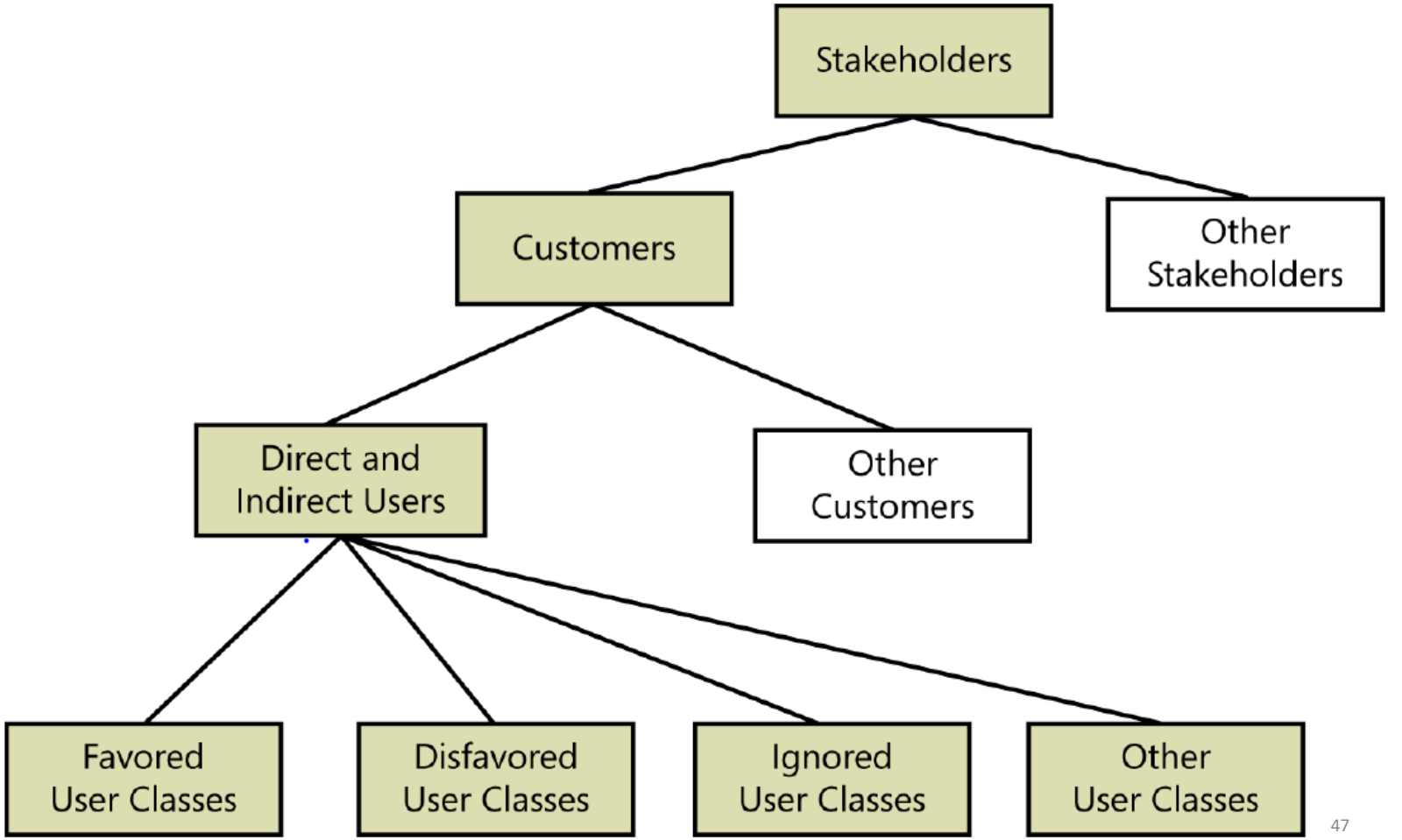
Finding the Voice of Users

- Identify the different classes of users
- Select and work with individuals who represent each user class and other stakeholders.

User Classes

- A user class is a subset of the product's customers, which is a subset of product's stakeholders.
  - Their access privilege or security levels (e.g., ordinary user, guest user, administrator)
  - The tasks and goals they perform with the system
  - 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

User Classes



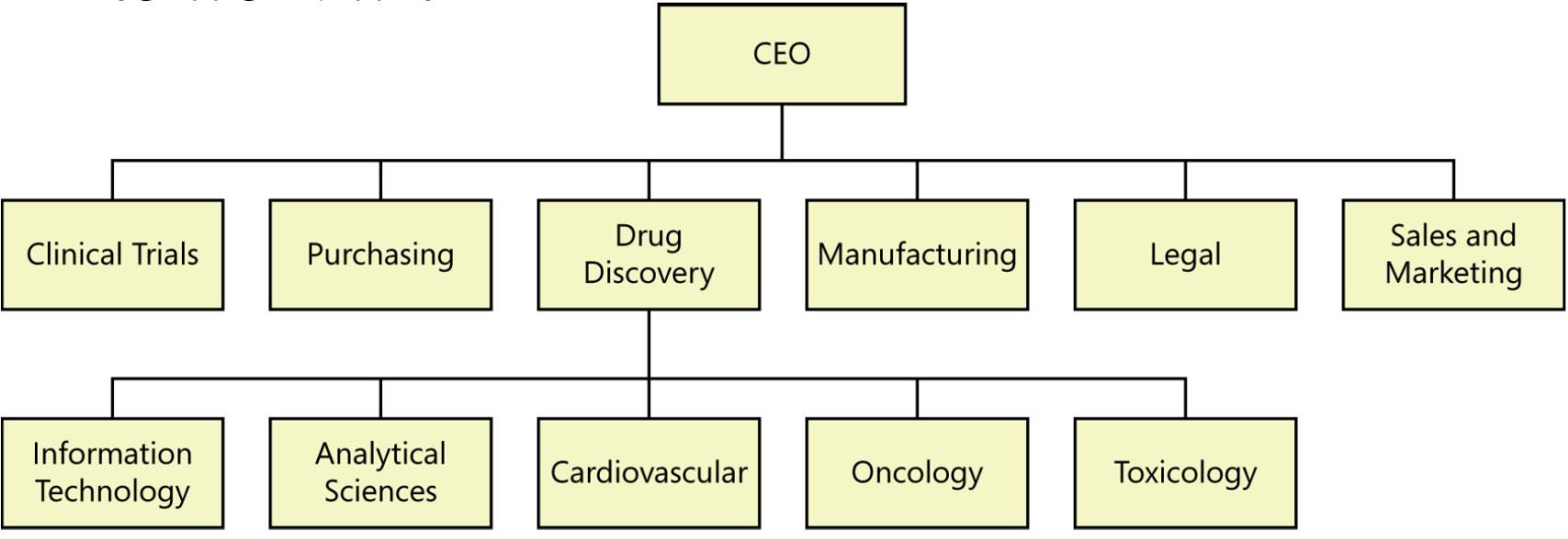
User Classes

- An individual can belong to multiple user classes
- Overlap between the needs of different user classes
- Don't overlook indirect user classes.
- User classes need not be human beings, such as software agency, robots.
- Approaches to identify user classes
  - Ask the project sponsor
  - Brainstorm
  - Use the external entities of the context diagram
  - Analyze the organization chart

User Classes

A organization chart

- Reduce the likelihood that an important class of users is overlooked.
- Help judge how many user representatives need to work with



User Classes

Name	Number	Description
Chemists (Favored)	Approximately 1,000 located in 6 buildings	Chemists request chemicals from vendors and from the chemical stockroom. Each chemist will use the system 6 times per day.
Buyers	5	Buyers in the purchasing department process chemical requests. They place and track orders with external vendors.
Chemical stockroom staff	6 technicians, 1 supervisor	The chemical stockroom staff will supply containers from three stockrooms, request new chemicals from vendors, and track the movement of all containers.
Health department staff (favored)	1 manager	They will use the system only to generate predefined quarterly reports that comply with federal and state chemical usage and disposal reporting regulations.

User Personas

- A description of a representative member of the user class
- to help you understand the requirements
- to design the user experience to best meet the needs of specific user communities, before an actual user is available

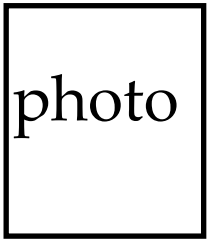
***Fred**, 41, has been a chemist at Contoso Pharmaceuticals.*
He doesn't have much patience with computers.
Fred usually works on two projects in related chemical areas.
His lab contains approximately 300 bottles of chemicals and gas cylinders.
On an average day, he'll need four new chemicals from the stockroom.
Two of these will be commercial chemicals in stock, one will need to be ordered, and one will come from the supply of proprietary Contoso chemical samples.
On occasion, Fred will need a hazardous chemical that requires special training for safe handling.

User Personas

- A way to personalize user requirements
  - fictitious stand ins for real user classes
  - removes obvious abstraction
  - identify user motivations, expectations and goals.
  - enable the design team to stand in the user's shoes
  – focus on users' goals

Simple Layout

Name



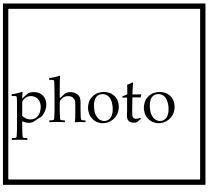
a summary of persona

multiple paragraphs

This layout tends not to be useful as the information tends to be disorganized

Simple Layout 1

Fred



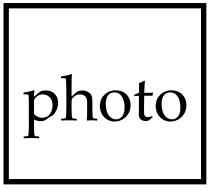
Fred, 41, a chemist at Contoso Pharmaceuticals. He doesn't have much patience with computers.

Fred usually works on two projects in related chemical areas. His lab contains approximately 300 bottles of chemicals and gas cylinders. On an average day, he'll need four new chemicals from the stockroom. Two of these will be commercial chemicals in stock, one will need to be ordered, and one will come from the supply of proprietary Contoso chemical samples.

On occasion, Fred will need a hazardous chemical that requires special training for safe handling.

Simple Layout2

Name

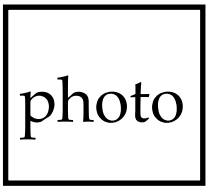


introduction to personalize

category1 description1
category2 description2

Simple Layout2

Fred



Fred, 41, a chemist at Contoso Pharmaceuticals.

Skill	Limited patience with computers
Projects	Two projects in related chemical areas.
Lab	He is in xxx lab, and his lab contains approximately 300 bottles of chemicals and gas cylinders.
Requests	Need four new chemicals from the stockroom. Two of these will be commercial chemicals in stock, one will need to be ordered, and one will come from the supply of proprietary Contoso chemical samples.

Tips

- Give the persona a name and a photo
- Keep the persona to one page
  – can be referred to quickly
- Add personal details, but don't go overboard
- Include goals why they need the product
- Other details
  - work environment (tools used)
  - computer proficiency
  - frequency to use the product

Using Personas

- Identify the features and functionality
- Determine if one user interface will meet the goals of all users, or if there needs to be more than one interface
- Make design decisions about how functionality will work
- Represent behavior patterns, not job descriptions.
  - not a list of tasks and duties.
  - describes the flow of someone's behavior, as well as skills, attitudes, environment, and goals.

User Representatives

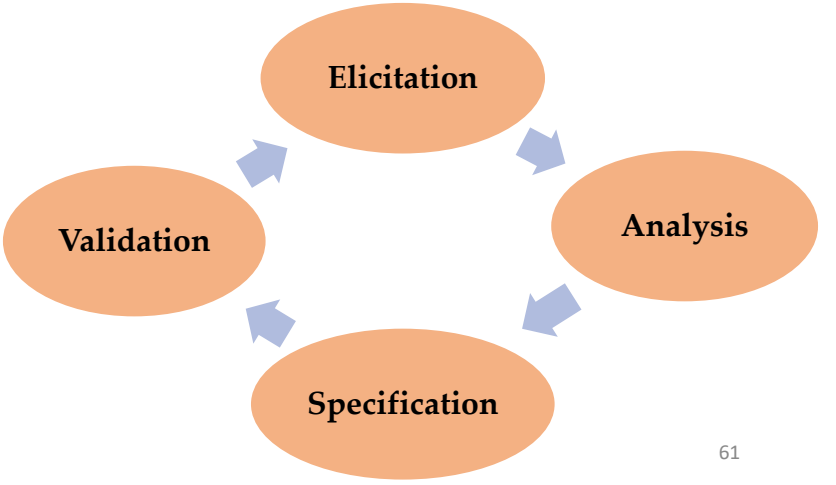
- User representatives provide the voice of the users
- They are involved throughout the development life cycle
- User representatives will be actual users that represent the members of a user class.
- RA talks with representatives instead of guessing the needs of users

Outline

- Business Requirements
- Finding the voice of users
- Requirement Elicitation
- Elicitation Techniques

Requirements Elicitation

- The process of identifying the needs and constraints of the various stakeholders for a product.
- A collaborative and analytical process that includes activities to collect, discover, extract, and define requirements.
- To discover business, user, data, functional, and nonfunctional requirements, along with other types of information.



RA Must:

- Identify the "problem"/"opportunity"
  - Which problem needs to be solved? (identify problem/boundaries)
  - Where is the problem? (understand Context/Problem Domain)
  - Whose problem is it? (Stakeholders)
  - Why does it need solving? (Stkhdrs Goals)
  - How might a software system help? (collect some Scenarios)
  - When does it need solving? (Constraints)
  - What might prevent us solving it? (Feasibility/Risk)

Sources

- Customer specifications
- Documentation for existing systems
- Users of existing systems
- Potential users of new system
- Predecessor products
- Competitors' products
- Domain experts
- Documents for interfacing systems
- Standards and Legislation

Requirements Elicitation Difficulties

- Problem Domain knowledge is spread over many sources
  - textbooks, operating manuals, process manuals, people's brains
  - specialist terminology
  - rarely available in an explicit form
- Tacit Knowledge ("say-do")
  - people find it hard to describe knowledge they regularly use
  - taken for granted and too obvious

Requirements Elicitation Difficulties

- Limited observability
  - people are busy
  - direct access to real users may be difficult
- Ignorance
  - Stakeholders may not know!!
  - May make unreasonable demands (unaware of cost)
- People may not be free to tell you what you need to know
  - ◊ political games

Review

- Finding the Voice of Users
  - User Classes
  - User Personas
  - User Representatives
- Requirement Elicitation
  - Dimensions and Opportunities
  - Elicitation Difficulties

Outline

- Business Requirements
- Finding the voice of users
- Requirement Elicitation
- Elicitation Techniques
- Understanding User Requirements
- Understanding Business Rules
- Specifying Data Requirements
- Specifying Non-functional Requirements
- Setting Requirement Priorities
- Requirement Re-Use

Elicitation Techniques

- Facilitated activities: RAs interact with stakeholders to elicit requirement
  Focus on discovering business and user requirements
- Independent activities: RAs work on their own to discover information.
  Present and reveal needed functionality that users might not aware of

Interviews
Workshops
Focus Groups
Observations
Questionnaires

System Interface Analysis
User Interface Analysis
Document Analysis

Technique – Interview

- A conversation where one participant asks questions, and others provide answers.
- Discuss system with different stakeholders
- Types
  - structured/closed -> agenda of predefined questions
  - open -> no pre–set agenda
    - ◊ free wheeling discussion
    - ◊ adaptive questions
- Start with questions, move to discussion

Technique – Interview

- Advantages
  - rich collection of information
  - users usually happy to talk about tasks
- Disadvantages
  - large amounts of qualitative data
  - hard to compare different respondents
  - interviewing is a difficult skill
  - unrealistic expectations

Technique – Interview

- Preparation:
  - determine objectives in advance
  - make an appointment
  - prepare questions and suggest ideas
  - plan for iteration
    - ◊ extended interviews counter productive (1 hour concentration limit)
    - ◊ second thought (present results from previous interview, move to new info)
  - send the agenda before the interview

Technique – Interview

- Preparation:
  - recording (how will you keep notes?)
    - ◊ Yourself
    - ◊ stenographer
    - ◊ record (audio/visual)
  - have a plan
  - leave time to recap and check objectives at the end

Technique – Interview

- Conduct:
  - polite, friendly, put client at ease
  - flexible
  - tenacious
    - ◊ make sure you understand
    - ◊ lead with questions, not answers
    - ◊ keep discussion focused on objectives
  - listen actively and restate the main ideas

Technique-Interview

- Strengths:
  - broad bandwidth of communication
    - ◊ verbal & body language
    - ◊ drawings, sketches
  - inherent flexibility
    - ◊ unplanned topics, on-the-fly topics
  - broad scope
- Interview different stakeholders
  - common mistake: only direct users
  - development team, management, accounting

Workshop

- Designed to encourage consensus
- Gather several types of key stakeholders
- Short intensive period
- Facilitator: guiding to a successful outcome
- Output: preliminary system definition at the feature level
- Benefits:
  - Team building
  - All stakeholders get a say
  - Achieve agreement
  - Output is available immediately

Workshop

- Participation of right stake holders
- Keep the team small
- Plan an agenda
- Establish and enforce ground rules

Workshop

- Stay in scope
- Timebox discussions
- Keep everyone engaged
- Use parking lots to capture items for later consideration

Focus Groups

- A representative group of users to generate input and ideas on a product's functional and quality requirements
- Interactive, allowing all users a chance to voice their thoughts
- Useful for exploring users' attitudes, impressions, preferences, and needs
- Select the focus group members carefully
- Be facilitated and keep them on topic
- Participants do not have decision-making authority

Observations

- It is hard to precisely describe the complex tasks and remember the details
- Observe exactly how users perform their tasks.
- Observations are time consuming
- Limit each observation time to two hours or less.
- Select important or high-risk tasks and multiple user classes for observations
- Observations can be silent or interactive

Questionnaires

- To survey large groups of users to understand their needs
- Preparing well-written questions is the biggest challenge
  - Provide answer options that cover the full set of possible responses
  - Make answer choices both mutually exclusive and exhaustive
  - Use closed questions with two or more specific choices
  - Consider consulting with an expert in questionnaire design and administration
  - Always test a questionnaire before distributing it.
  - Don't ask too many questions or people won't respond

Questionnaires

- Advantages
  - Inexpensive
  - Can quickly collect info from large number of people
  - Can be administered remotely
    - ◊ paper or electronic
- Disadvantages
  - No context
    - ◊ no room for users to convey their real needs
  - difficult to explore new domains

System Interface Analysis

- An independent elicitation technique
- Examine the systems to which the system connects
- Context diagrams and ecosystem maps are starting points
- Reveal functional requirements regarding the exchange of data and services between systems
- Identify functionality in the other system that might lead to requirements for the developing system

User Interface Analysis

- An independent elicitation technique
- Existing systems are studied to discover user and functional requirements.
- Using the screen shots of the existing systems and identifying a complete list of screens to help you discover potential features
- Learn about the common steps that users take
- Understand how an existing system works
- Do not assume that certain functionality is needed in the new system just because you found it in an existing one

Document Analysis

- An independent elicitation technique
- Examine the existing documentation for potential software requirements

  Requirements specifications, business processes, lessons-learned collections, and user manuals for existing or similar applications
- A way to get up to speed on an existing system or a new domain
  - reduce the elicitation meeting time needed
  - reveal information people don't tell you
- Watch for: The available documents might be not up to date.

Review

- Elicitation Techniques
  - Interview
  - Workshops
  - Focus Group
  - Observations
  - Questionnaires
  - System Interface Analysis
  - User Interface Analysis
  - Document Analysis

Planning Elicitation on Project

- RA plans the project's approach to requirements elicitation
- An elicitation plan includes the techniques RA will use, when RA plan to use them, and for what purpose.
- To address the following items
  - Elicitation Objectives
  - Elicitation strategies and planned techniques
  - Time and resource estimates
  - Documents and systems for independent elicitation
  - Expected products of elicitation efforts
  - Elicitation risks

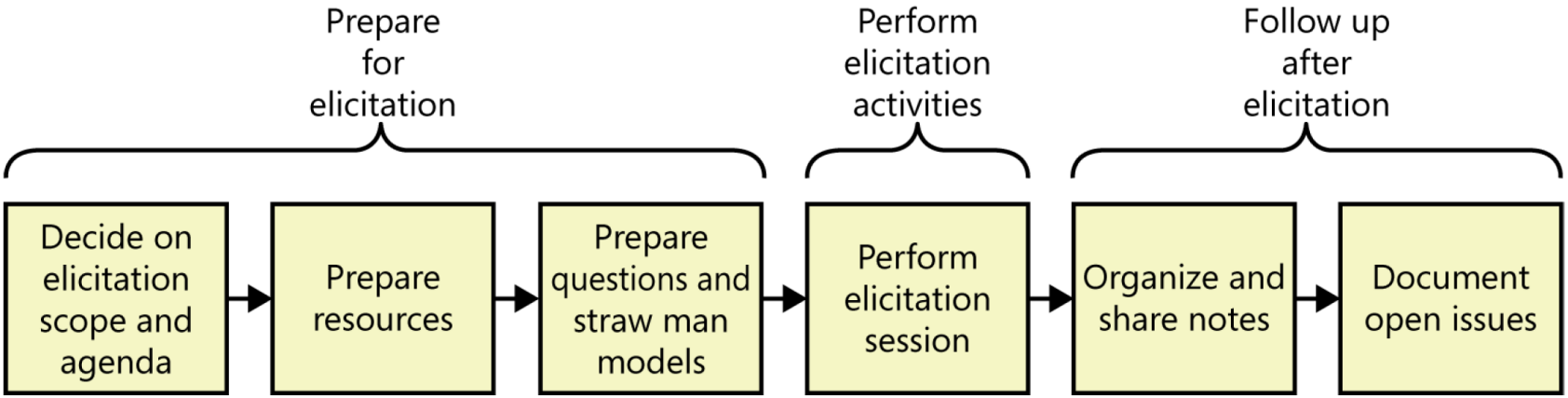
Planning Elicitation on Project

- Suggested elicitation techniques by project characteristic

	Interviews	Workshops	Focus groups	Observations	Questionnaires	System interface analysis	User interface analysis	Document analysis
Mass-market software	x		x		x			
Internal corporate software	x	x	x	x		x		x
Replacing existing system	x	x		x		x	x	x
Enhancing existing system	x	x				x	x	x
New application	x	x				x		
Packaged software implementation	x	x		x		x		x
Embedded systems	x	x				x		x
Geographically distributed stakeholders	x	x			x			

Single Requirements Elicitation Session

- Prepare for elicitation
- Perform elicitation activities
- Follow up after elicitation



Preparing for Elicitation

- Elicitation preparation is to make best use of everyone's time
  - Plan session scope and agenda
  - Prepare resources (equipment and participants)
  - Learn about the stakeholders (cultural and regional preferences)
  - Prepare questions
- Phrase the questions to avoid leading customers down an unintended path

"What do you want?"

"What do you need to do?"

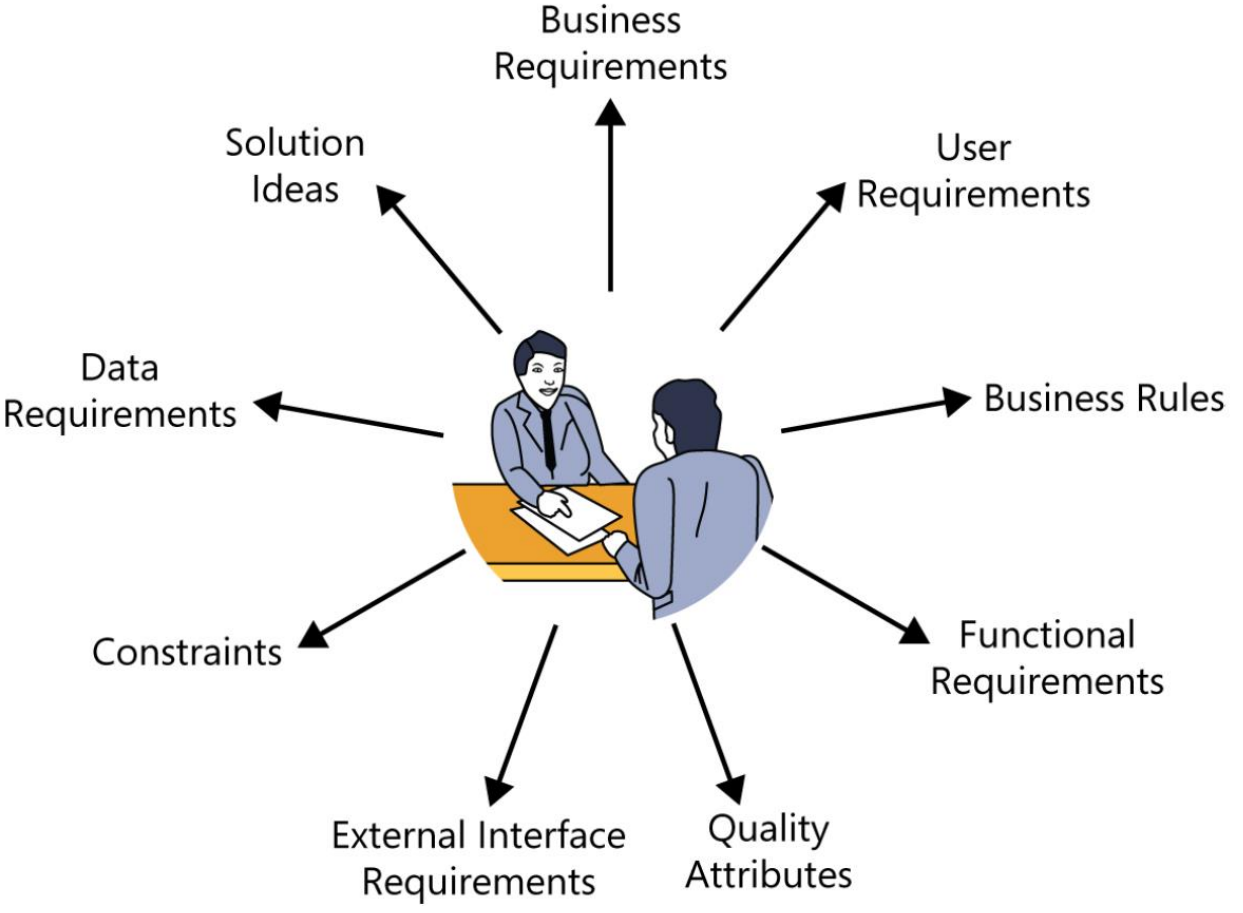
Performing Elicitation Activities

- Using the elicitation techniques to collect information from stakeholders

- Educate stakeholders:
  Teach the approach and why use it
  Explain the techniques
- Take good notes:
  assign someone to take accurate notes
  keep conversation going by questions

Following up after Elicitation

- Organize and share the notes for reviewing and updating
- Classify customer inputs:



Following up after Elicitation

- **Business Requirements**: The financial, marketplace, or other business benefit that either customers or the developing organization wish to gain from the product.

"Reach a sales volume of X units or revenue of $Y within Z months."
"Increase market share in region X by Y percent within Z months."
"Receive no more than X service calls per unit and Y warranty calls per unit within Z months after shipping."

Following up after Elicitation

- **User Requirements**: General statements of user goals or tasks that users need to perform
- "I need to <do something>"

"As the lead machine operator, I need to calibrate the pump controller first thing every morning."
"I need to print a mailing label for a package."

Following up after Elicitation

- **Business Rules**: Only certain users can perform an activity under specific conditions.
- Derive some functional requirements to enforce the rules
- "Must comply with . . . ," "If <some condition is true>, then <something happens>," or "Must be calculated according to . . ."

"A new client must pay 30 percent of the estimated consulting fee and travel expenses in advance."
"Time-off approvals must comply with the company's HR vacation policy."

Following up after Elicitation

- **Functional Requirements**: The observable behaviors the system will exhibit under certain conditions and the actions the system will let users take.

"If the pressure exceeds 40.0 psi, the high-pressure warning light should come on."
"The user must be able to sort the project list in forward and reverse alphabetical order."

Following up after Elicitation

- **Quality Attributes**: How well the system does.
- Describe desirable system characteristics: fast, easy, user-friendly, reliable, secure.

"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."

Following up after Elicitation

- **External Interface Requirements**: The connections between the developing system and the rest of the universe.
- "Must read signals from . . . ," "Must send messages to . . . ," "Must be able to read files in <format>," and "User interface elements must conform to <a standard>".

"The manufacturing execution system must control the wafer sorter."
"The mobile app should send the check image to the bank after I photograph the check I'm depositing."

Following up after Elicitation

- **Constraints**: Design and implementation constraints legitimately restrict the options available to the developer.
- Devices with embedded software often must respect physical constraints such as size, weight, and interface connections.
- "Must be written in <a specific programming language>," "Cannot exceed <some limit>," and "Must use <a specific user interface control>."

"Files submitted electronically cannot exceed 10 MB in size."
"The browser must use 256-bit encryption for all secure transactions."

Following up after Elicitation

- **Data Requirements**: 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."

Following up after Elicitation

- **Solution Ideas**: A specific way to interact with the system to perform some action.
- Probe the surface of a solution idea to get to the real requirement.
- Repeatedly asking "why" the user needs it

"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."