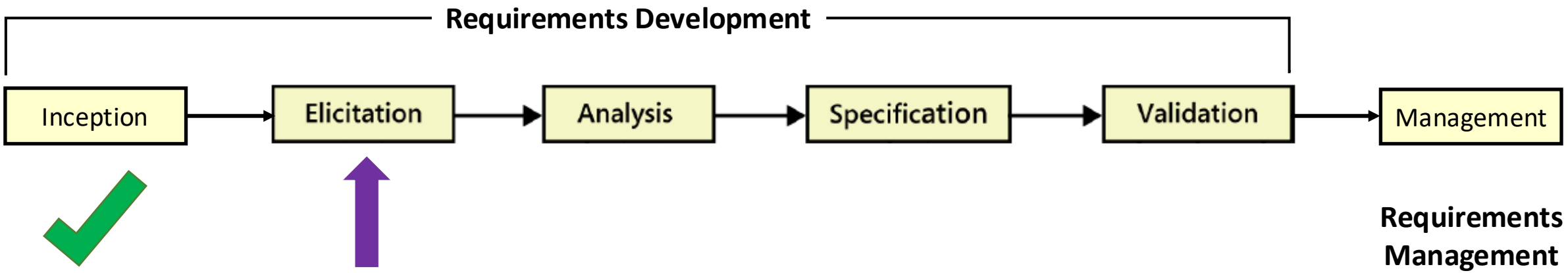
Requirements Elicitation

SE 311

Outline

- Overview of requirements elicitation
- Requirements elicitation techniques
- Use cases

RE Process


Requirements Development
Inception
Elicitation
Analysis
Specification
Validation
Management
Requirements Management

Requirements Elicitation

- Is the process of identifying the needs and constraints of the various stakeholders for a software system

Definition of *elicit* verb	استخرج، استنبط، انتزع
1: to call forth or draw out (something, such as information or a response) // her remarks elicited cheers **2:** to draw forth or bring out (something latent or potential) // hypnotism elicited his hidden fears	

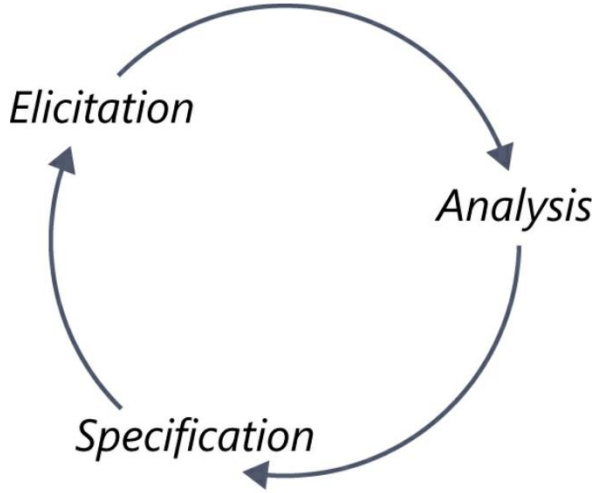
Requirements Elicitation

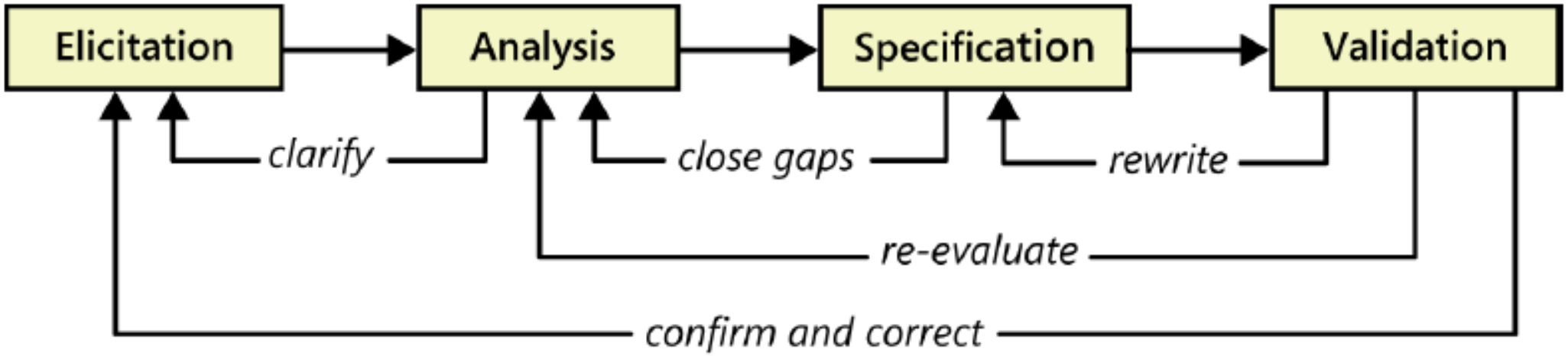
- A collaborative and analytical process that includes activities to collect, discover, extract, and define requirements
- Discovers business, user, functional, and non-functional requirements
- Engaging users in the elicitation process is a key element for success

Requirements Elicitation Goals

- Determine sources of requirements and select appropriate techniques
- Elicit information on domain, problem, needs, and constraints
- Produce a *first document*
  - Mainly user requirements and elicitation notes
  - Potentially incomplete and inconsistent but one must start somewhere

Requirements Elicitation

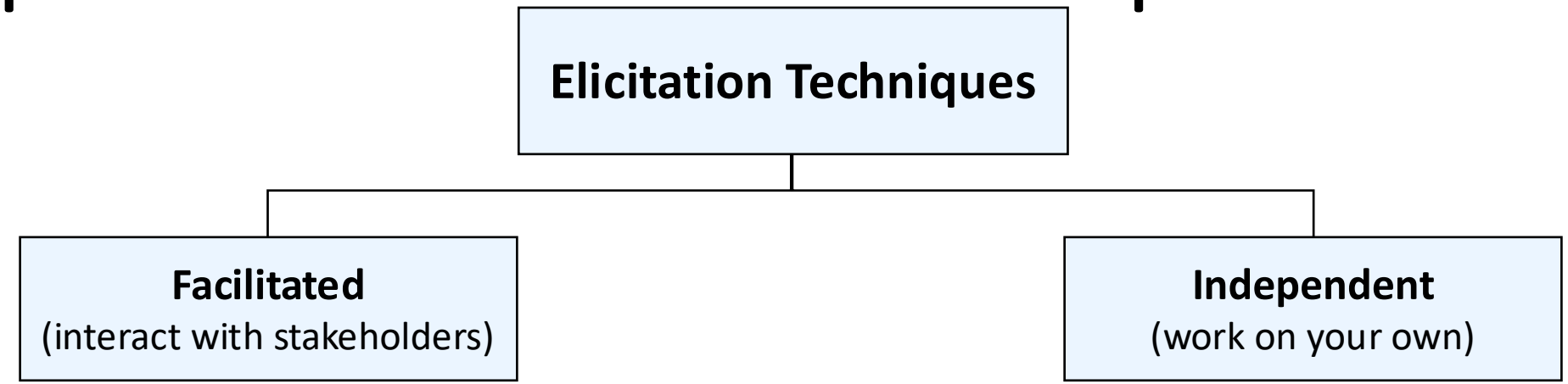




Requirements development is an iterative process: do some elicitation, study what you learn, write some requirements, realize you have conflicting or missing requirements, return to elicitation and repeat.

Your requirements won't be elicited in just 1 or 2 sessions.

Requirements Elicitation Techniques



- Interviews
- Workshops
- Focus groups
- Observations
- Questionnaires

Facilitated techniques primarily focus on discovering user and business requirements.

- System interface analysis
- User interface analysis
- Document analysis

Independentnt techniques focus on discovering needed functionality that the users might not be aware of

Interviews

- May be one-on-one or small group interviews
- Strengths:
  - Effective for getting to know the application domain quickly
  - Interviewees may be more comfortable sharing their thoughts than in a workshop setting
  - Easier to establish user involvement
  - Suitable for busy executives

Interviews - Tips

1. Acquire background knowledge • Domain (e.g. problems, vocab) • Interviewee (e.g. work tasks, attitude)	**2. Prepare questions**
3. Establish rapport • Introduce yourself • Review agenda and session objectives	**4. Stay in Scope**
5. Suggest ideas	**6. Listen Actively** • Active listening • Paraphrasing

Interviews - Tips



Workshops

"Facilitation is the art of leading people through processes toward agreed-upon objectives in a manner that encourages participation, ownership, and productivity from all involved" (Sibbet, 1994)

- Facilitated and structured meetings with a selected group of stakeholders during which requirements are elicited from multiple stakeholders concurrently.
- Include several types of stakeholders and formal roles
- <u>Strengths:</u>
  - Effective in solving disagreements
  - Suitable for tight schedules

Workshops - Tips

1. Establish and enforce ground rules	**2.** Fill all of the team roles
3. Plan an agenda	**4.** Stay in scope
5. Use parking lots to capture items for later consideration	**6.** Timebox discussions
7. Keep the team small but include the right stakeholders	**8.** Keep everyone engaged

Workshops - Tips



"If we want to succeed as a team, we need to put aside our own selfish, individual interests and start doing things my way."

Watch out for such people and don't allow them to dominate the discussion and intimidate others

Focus Groups

- A representative group of users who convene in a facilitated elicitation activity to generate input and ideas on a product's functional and quality requirements
- Useful for exploring users' attitudes, impressions, preferences, and needs

You will need to keep them on topic without influencing their opinions
Include users who have used previous versions or similar products
Select users from different user classes or a bunch of users from the same class and hold several focused groups
Participants in focus groups normally don't have decision-making authority for requirements

Focus Groups

- Genrates subjective feedback that can be further evaluated and prioritized as requirements are developed
- <u>Strengths:</u>
  - Valuable for developing commercial products



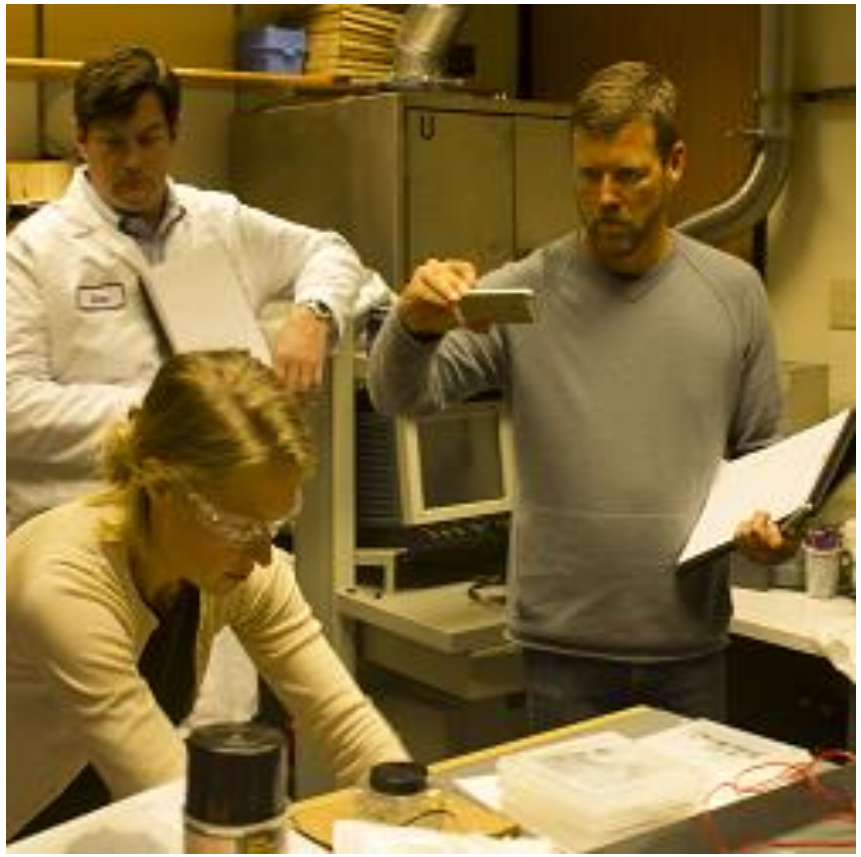
Observations



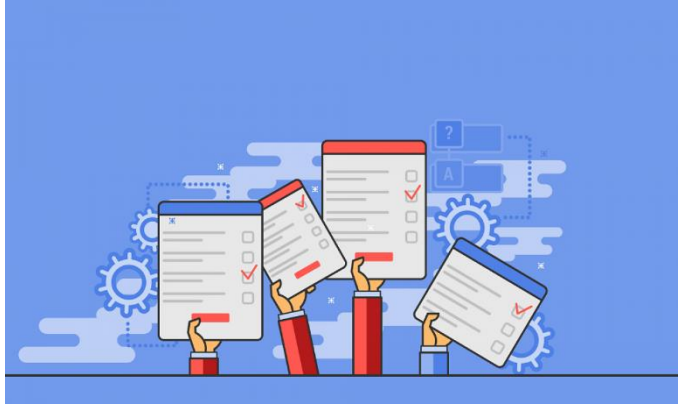
- Get into the trenches and observe specialists “in the wild”
- Shadow important users as they do their work
- Can be silent or interavtive
- May be supplemented later with questionnaires and/or interviews
- Time consuming

Observations

- <u>Strengths:</u>
  - Useful for important or high-risk tasks
  - Facilitates requirements validation
  - Useful for identifying new topics for interviews



Questionnaires

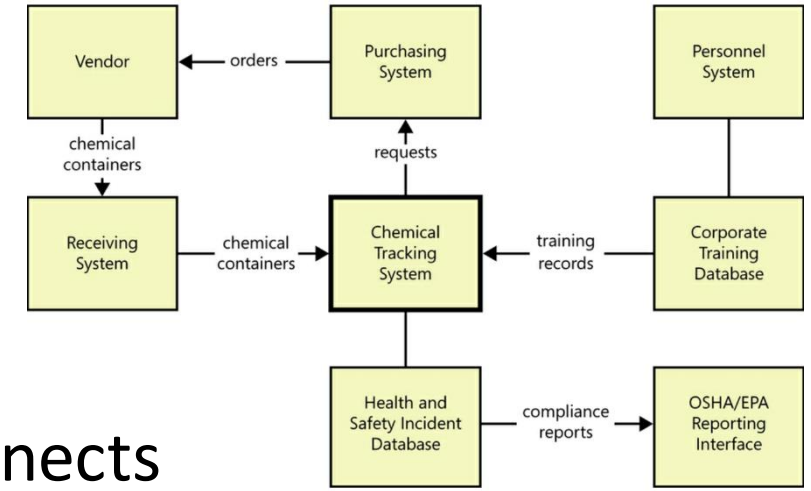


- A way to survey large groups of users to understand their needs
- Analyzed results can be used as input for other elicitation techniques
- Survey commercial product users for feedback
- Strengths:
  - Inexpensive
  - Suitable for eliciting information from large user populations
  - Easily administered across geographical boundaries

Questionnaires – Tips

1. Cover the full set of possible responses	**2.** Make answer choices both mutually exclusive and exhaustive
3. Avoid suggestive questions	**4.** Use scales consistently
5. Use closed questions with a number of choices for statistical analysis	**6.** Consult an expert on questionnaires
7. Test the questionnaire	**8.** Avoid asking too many questions

System Interface Analysis



- Examining the systems to which your system connects
- Reveals functional requirements regarding the exchange of data and services between the systems
- Consult context diagrams and ecosystem maps to identify systems
- Identify functionality in the other system that may lead to requirements for your system

User Interface Analysis



- Study existing systems to discover user and functional requirements
- Great way to get up on speed on how an existing system works
- In absence of an existing system, one may consult user interfaces of similar products
- Don’t assume that you need to stick to all the features and aspects of user interface for the next implementation

User Interface Analysis

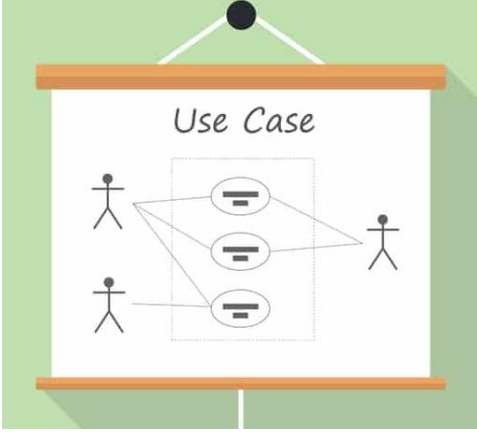
- Strengths:
  - Suitable for building a new version of an existing system
  - Facilitates identification of areas of improvement
  - Relates user input to reality, thereby clarifying it

Document Analysis

- Examining any existing documentation for potential requirements
- Good way for getting up to speed on an exisitng system or a new domain
- Available documents might not be up to date



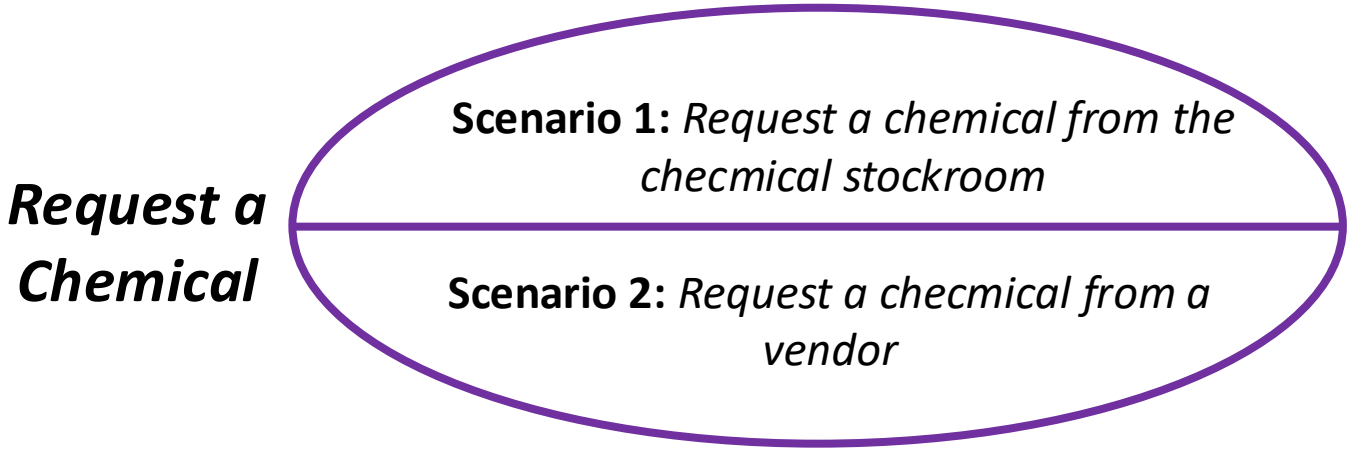
Use Cases



- A sequence of interactions between a system and an external actor that results in the actor being able to achieve some outcome of value
- A user-centric and usage-centric technique for exploring user requirements
- Shifts from product-centric perspective (asking users what they want the *system* to do) to discussing what the *users* need to accomplish

Usage Scenario

- A *scenario* is a description of a single instance of a usage of a system
- A use case is a collection of related usage scenarios, and a scenario is a specific instance of a use case
  - Specific actor, at a specific time, with specific data



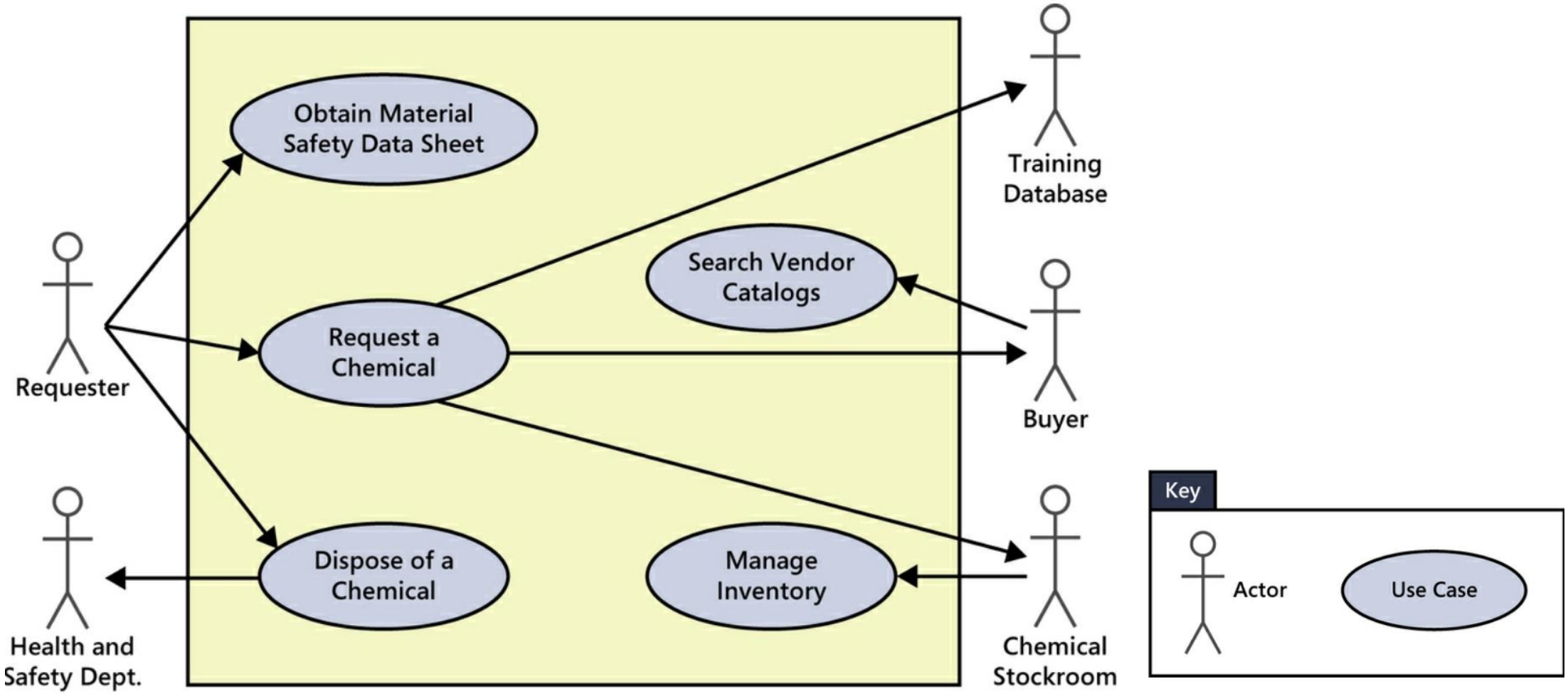
Representation of Use Cases - Textual

- Narrative form – a paragraph focusing on the primary scenario and some secondary ones

  A User inserts a card in the card reader slot. The system asks for a personal identification number (PIN). The user enters a PIN. After checking that the user identification is valid, the system asks the user to chose an operation...

- Useful when stakeholders first meet

Representation of Use Cases – Visual



Partial use case diagram for the Chemical Tracking System

Use Cases

- Strengths:
  - Users have clearer expectations of what the system will do
  - BA's and developers better understand user's business
  - Prevents implementation of unneeded functionality
  - May be turned into object models for object-oriented design
- Risks:
  - Too many use cases
  - Highly complex use cases

Misuse Cases

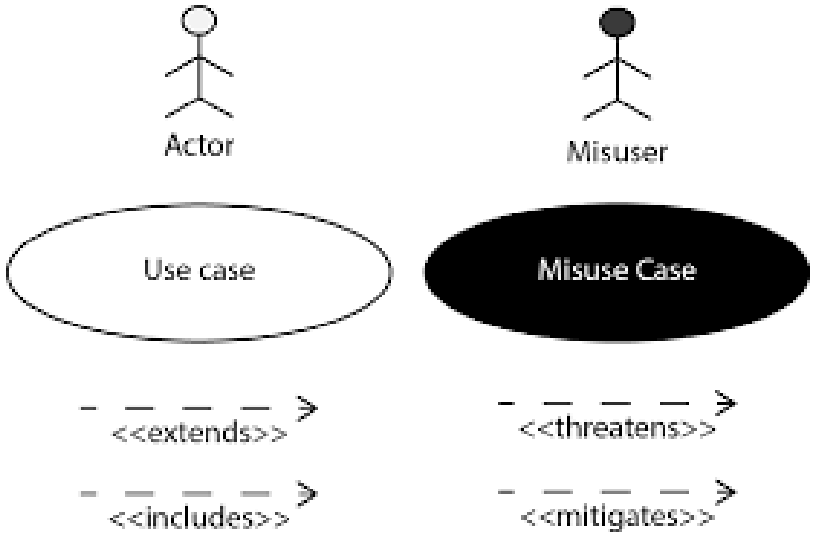
- A negative scenario is one with a goal that is:
  - Undesirable from a business obejctive's perspective
  - Desriable from a hostile agent's perspective
- Can be handled using *misuse cases*
  - Proposed by Sindre & Opdahl (2000)
  - Capture use cases that a system must be protected against
  - Obvious applications for security and risk analysis

Finding Misuse Cases

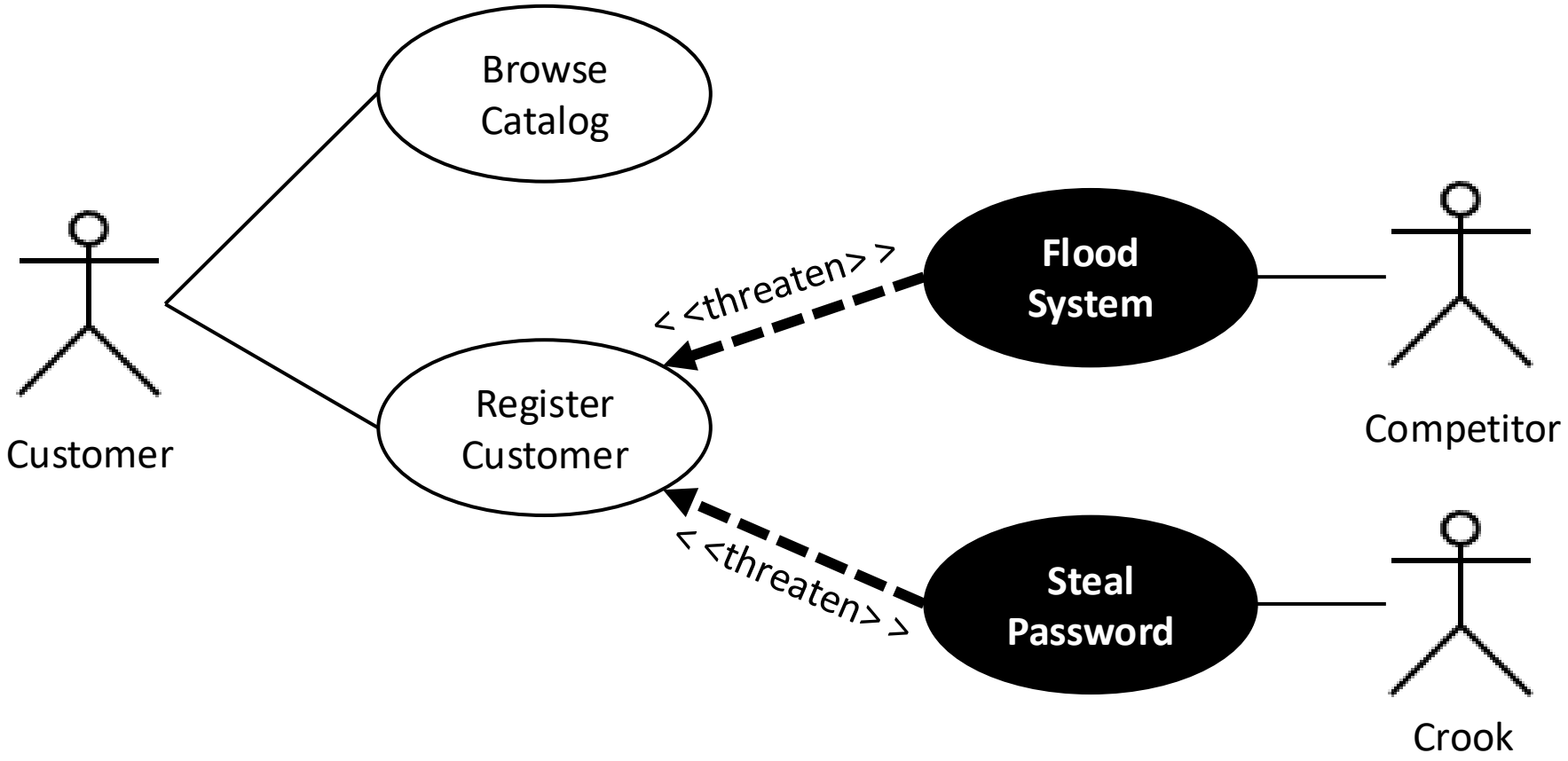


Agents with intentions to harm the system, its stakeholders, or their resources OR achieve goals incompatible with system goals

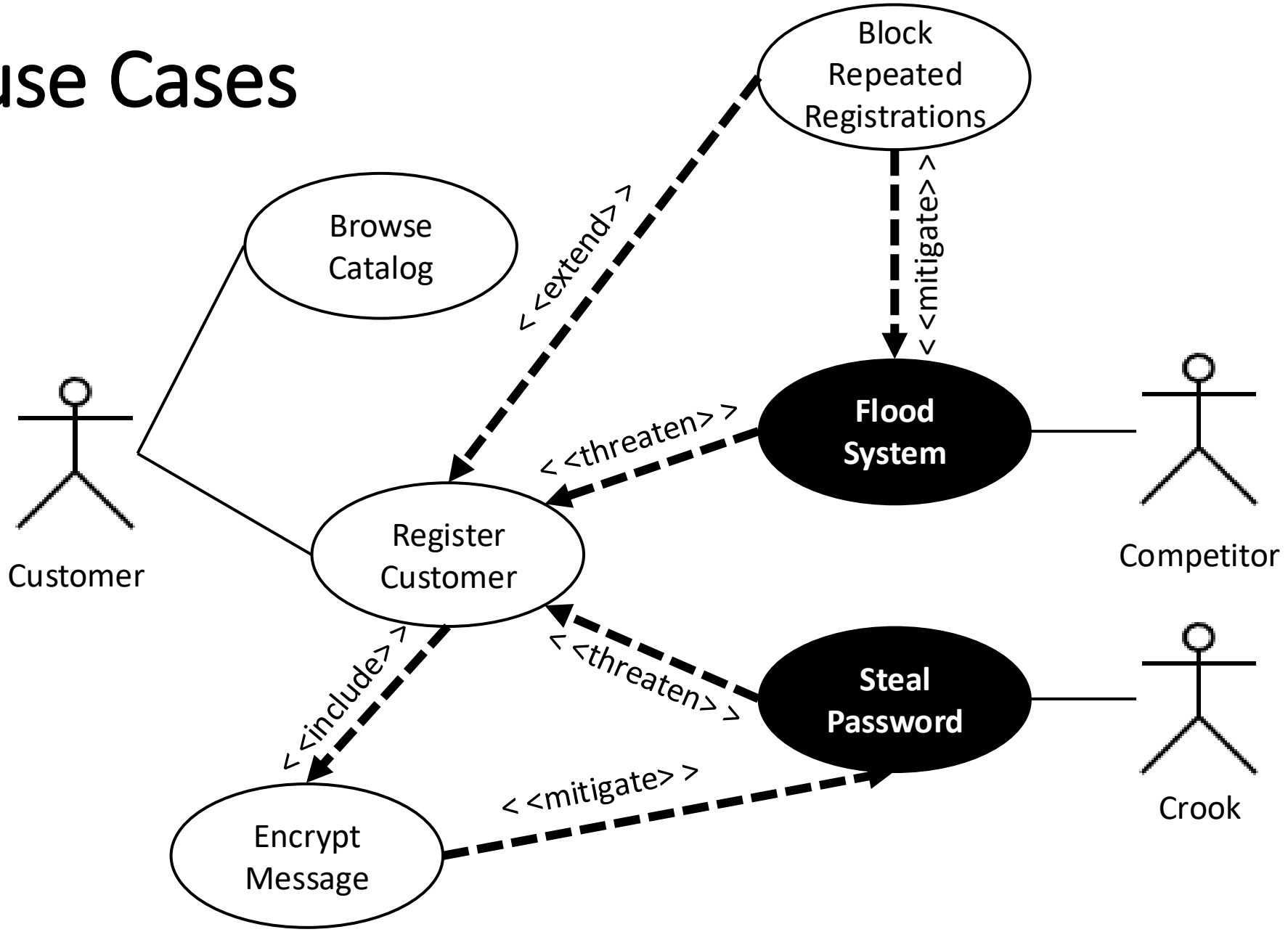
- Identify potential *misactors* for a given use case diagram
- Identify misuse cases – what would actors do to harm the system?
- Mitigate misuse cases



Misuse Cases



Misuse Cases



Misuse Cases

- Strengths:
  - Useful for eliciting security anf safety requirements
  - Early identification of threats, mitigations, and exceptions
- Risks:
  - If used during elicitation, it could lead to premature design solutions when identifying mitigations
  - Potential for missing misactors or threats if consulting partial view of system

M. Zarour, M. Alenezi: Software Security Specifications and Design How Software Engineers and Practitioners Are Mixing Things up

Prototyping

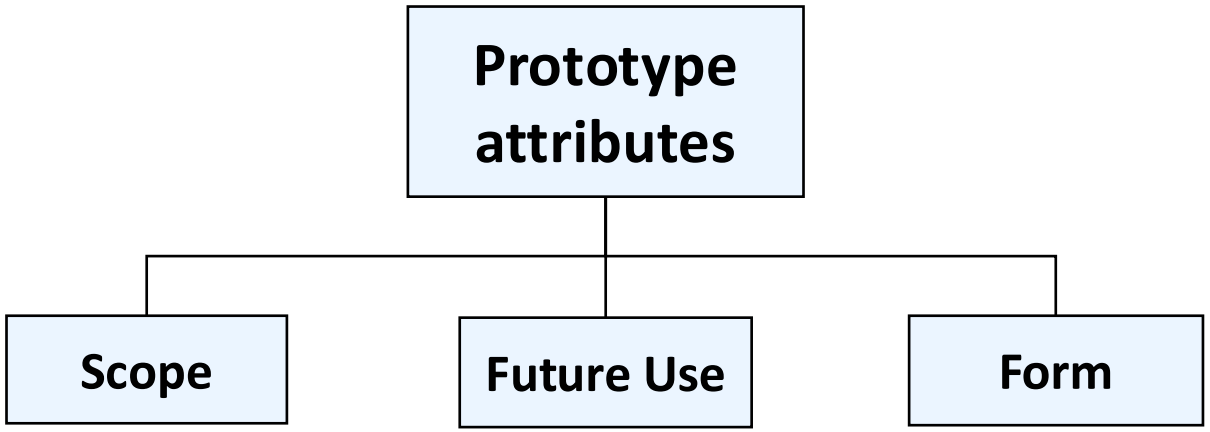


- A software prototype is a partial, possible, or preliminary implementation of a proposed new product
- It can be static designs or working models; quick sketches or highly detailed screens; visual displays or full slices of functionality; or simulations
- Takes a tentative step into the solution space

Prototyping - Purpose

- Prototypes can serve three major purposes, and that purpose must be made clear from the very beginning:
  - Clarify, complete, and validate requirements – *requirements tool*
  - Explore design alternatives – *design tool*
  - Create a subset that will grow into the ultimate product – *construction tool*

Prototyping – Attributes & Applications

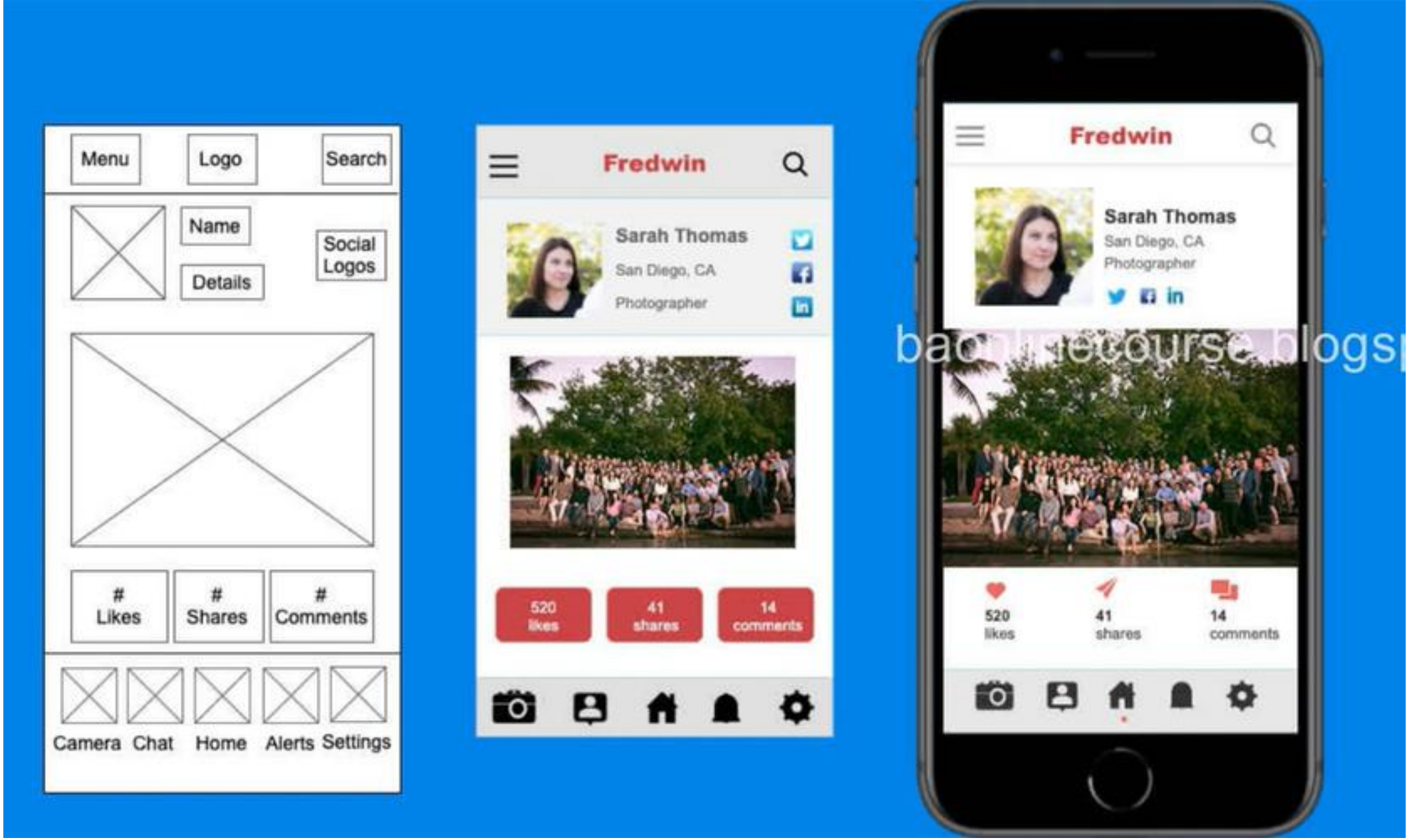


- Mock-up
- Proof-of-concept

- Throwaway
- Evolutionary

- Paper
- Electronic

	Throwaway	Evolutionary
Mock-up	▪ Clarify and refine user and functional requirements. ▪ Identify missing function-ality. ▪ Explore user interface approaches.	▪ Implement core user requirements. ▪ Implement additional user requirements based on priority. ▪ Implement and refine websites. ▪ Adapt system to rapidly changing business needs.
Proof of concept	▪ Demonstrate technical feasibility. ▪ Evaluate performance. ▪ Acquire knowledge to improve estimates for construction.	▪ Implement and grow core multi-tier function-ality and communica-tion layers. ▪ Implement and optimize core algorithms. ▪ Test and tune perfor-mance.



[Rapid Prototyping: Digital](#) [Figma](#)

Prototyping – Fidelity

- *Fidelity* is the extent to which the prototype is real and reactive
- It varies in level for throwaway prototypes

Fidelity	Description	Pros	Cons
Low	Static or non-functional	• Easy, quick, & cheap • Excellent for interfaces	• Not interactive • May deter some stakeholders
High	Fully functional	• Deeper understanding of functionality • More precise decisions	• Time consuming & costly • May be difficult to change

Prototyping



- <u>Strengths:</u>
  - Enhances user involvement
  - Closes expectation gaps
  - Resolves uncertainties early in the development process

Prototyping - Risks



- Pressure to release the prototype
- Distraction by detail
- Unrealistic performance expectations
- Investing excessive effort in prototypes

Prototyping – Success Factors

- ✔ Include them in project plans
- ✔ State the purpose of each prototype before you build it
- ✔ Plan to develop multiple prototypes
- ✔ Create throwaways as quickly and cheaply as possible
- ✔ Don’t prototype requirements that you already understand, except to explore design alternatives
- ✔ Don’t expect a prototype to replace written requirements

Final Remarks

Factors to consider when choosing elicitation techniques:

- The nature of the information being elicited in terms of consciousness
- The time and budget constraints
- The availability of stakeholders
- The experience of the BA with a particular elicitation technique

Appendix I – Sample Elicitation Questions

- Functional requirements
  - What will the system do?
  - When will the system do it?
  - Are there several modes of operations?
  - What kinds of computations or data transformations must be performed?
  - What are the appropriate reactions to possible stimuli?
  - For both input and output, what should be the format of the data?
  - Must any data be retained for any period of time?

Appendix I – Sample Elicitation Questions

- Design Constraints
  - Physical environment
    - Where is the equipment to be located?
    - Is there one location or several?
    - Are there any environmental restrictions, such as temperature, humidity, or magnetic interference?
    - Are there any constraints on the size of the system?
    - Are there any constraints on power, heating, or air conditioning?
    - Are there constraints on the programming language because of existing software components?

Appendix I – Sample Elicitation Questions

- Design Constraints
  - Interfaces
    - Is input coming from one or more other systems?
    - Is output going to one or more other systems?
    - Is there a prescribed way in which input/output need to be formatted?
    - Is there a prescribed way for storing data?
    - Is there a prescribed medium that the data must use?
  - Standards
    - Are there any standards relevant to the system?
  - Laws, policies, and regulations
    - Are there any laws, policies, or regulations applicable here?

Appendix I – Sample Elicitation Questions

- Performance
  - Are there constraints on execution speed, response time, or throughput?
  - What efficiency measure will apply to resource usage and response time?
  - How much data will flow through the system?
  - How often will data be received or sent?

- Usability and Human Factors
  - What kind of training will be required for each type of user?
  - How easy should it be for a user to understand and use the system?
  - How difficult should it be for a user to misuse the system?

Appendix I – Sample Elicitation Questions

- Security
  - Must access to the system or information be controlled?
  - Should each user's data be isolated from data of other users?
  - Should user programs be isolated from other programs and from the operating system?
  - Should precautions be taken against theft or vandalism?

Appendix I – Sample Elicitation Questions

- Reliability and Availability
  - Must the system detect and isolate faults?
  - What is the prescribed mean time between failures?
  - Is there a maximum time allowed for restarting the system after failure?
  - How often will the system be backed up?
  - Must backup copies be stored at a different location?
  - Should precautions be taken against fire or water damage?

Appendix I – Sample Elicitation Questions

- Maintainability
  - Will maintenance merely correct errors, or will it also include improving the system?
  - When and in what ways might the system be changed in the future?
  - How easy should it be to add features to the system?
  - How easy should it be to port the system from one platform (computer, operating system) to another?

- Precision and Accuracy
  - How accurate must data calculations be?
  - To what degree of precision must calculations be made?

Appendix II – Comparison of Elicitation Techniques

	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			

Appendix II – Comparison of Elicitation Techniques

Technique	Good for	Kind of data	Plus	Minus
Questionnaires	Answering specific questions	Quantitative and qualitative data	Can reach many people with low resource	The design is crucial. Response rate may be low. Responses may not be what you want
Interviews	Exploring issues	Some quantitative but mostly qualitative data	Interviewer can guide interviewee. Encourages contact between developers and users	Time consuming. Artificial environment may intimidate interviewee
Focus groups and workshops	Collecting multiple viewpoints	Some quantitative but mostly qualitative data	Highlights areas of consensus and conflict. Encourages contact between developers and users	Time consuming
Naturalistic observation	Understanding context of user activity	Qualitative	Observing actual work gives insight that other techniques cannot give	Very time consuming. Huge amounts of data
Studying documentation	Learning about procedures, regulations, and standards	Quantitative	No time commitment from users required	Day-to-day work will differ from documented procedures

[1] Preece, Rogers, and Sharp "Interaction Design: Beyond human-computer interaction", p214

Appendix III – Prototyping Tools

- https://medium.theuxblog.com/11-best-prototyping-tools-for-ui-ux-designers-how-to-choose-the-right-one-c5dc69720c47
- https://www.justinmind.com