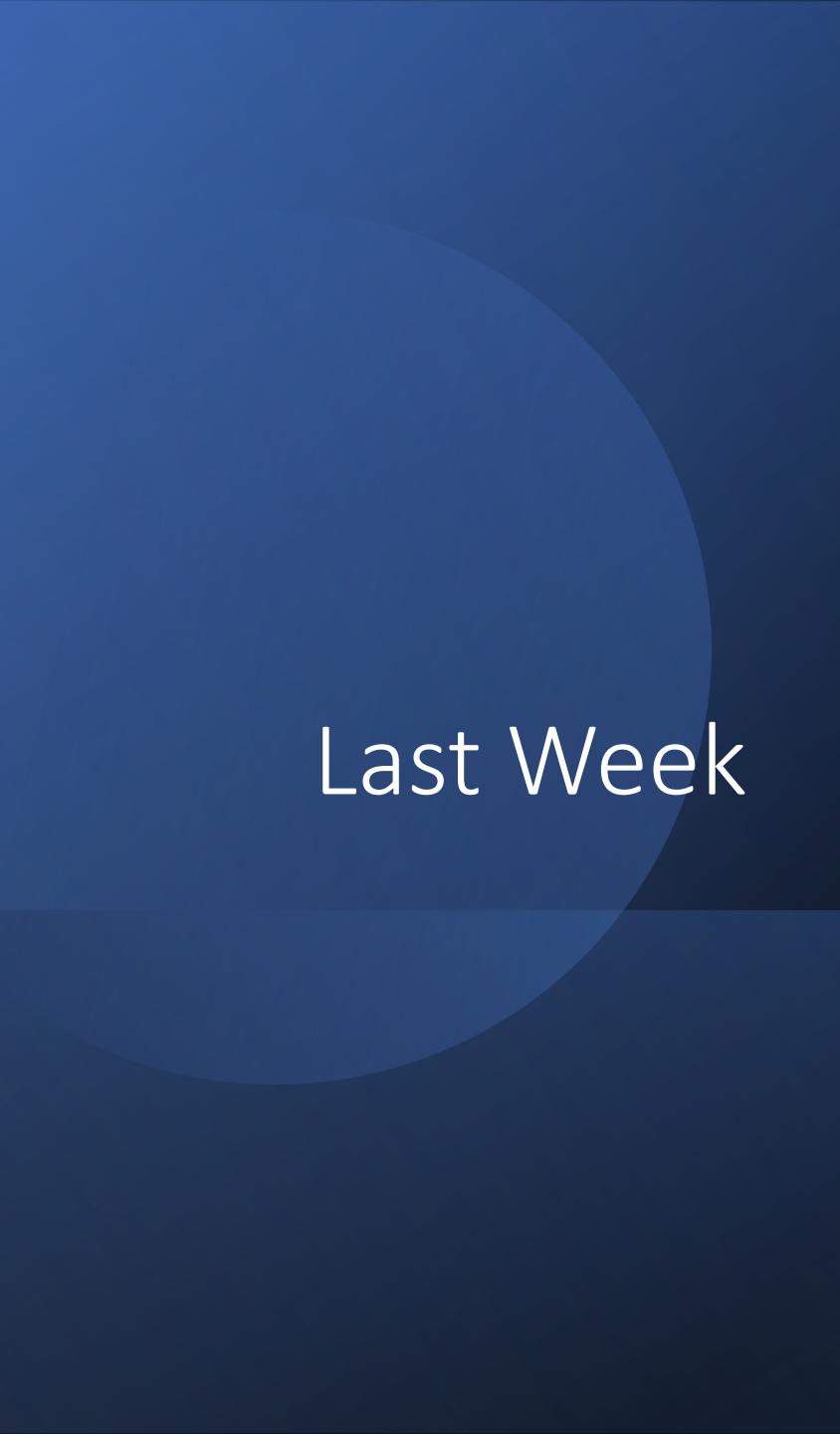




COE691: Software Requirements and Specifications

Dr. Rasha Kashef
Dept. of Electrical, Computer Engineering, and
Biomedical Engineering





Last Week

- Requirements Inception
 - Primary investigation procedure
- Requirements Elicitations
 - Requirement Resources
 - Stakeholder Analysis
 - Elicitation Techniques



Week 4: Agenda

- Requirements Elicitation Techniques
 - Classical
 - Modern

Elicitation: Basic Points



Elicitation is not Acquisition



Requirements are not available like sensor data
Not just read them systematically !!



Elicitation is not specification and modelling



RE Determines the **success** of the mission

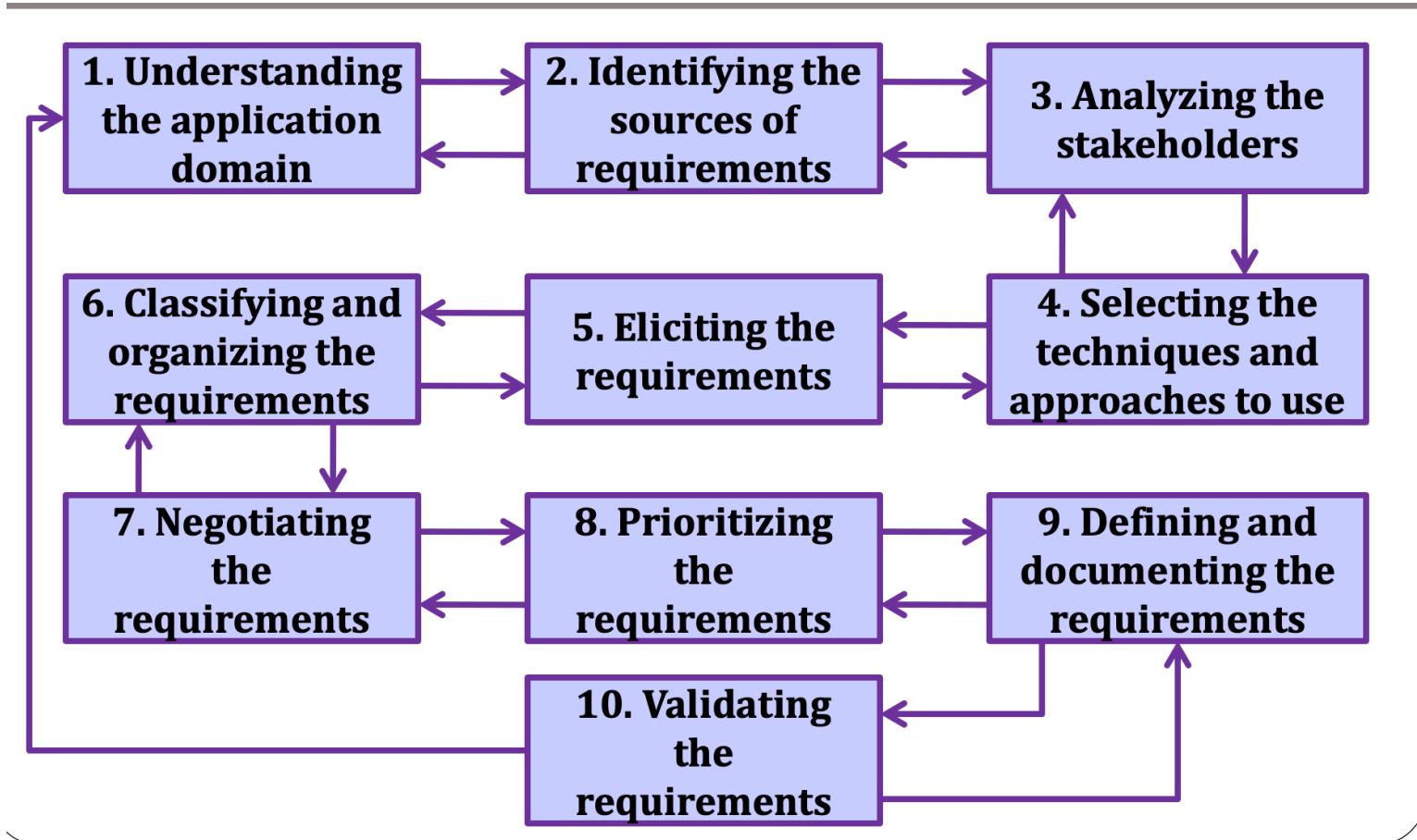


Elicitation determines the **success** of the RE process

Requirement Elicitation

- **Definition**
 - The process of acquiring (eliciting) all the relevant knowledge needed to produce a requirements model of a problem domain
 - The above definition implies that requirements elicitation is all about 'understanding' some particular problem domain. Only after understanding the nature, features and boundaries of a problem can the analyst proceed with a formal statement of the problem (requirements specification) and subsequently with its validation by the user (requirements validation)

The Process of User Requirements Elicitation



Elicitation Techniques

- There are many elicitation techniques on the market. It is very difficult to say which elicitation technique is suitable for all projects. Not all elicitation techniques can be executed for every project.
- Is there a universally best methodology?
 - No
- Can I combine methodologies?
 - Maybe
- When selecting an elicitation method, factors such as the nature of the project, organizational structure and type of stakeholders are taken into account by the business team before deciding which technique works best. Having said that, brainstorming, document analysis, interviews, prototyping and workshops are the most widely used requirement elicitation techniques.

Elicitation Techniques

- Questionnaires
- Interviewing
- Use cases and scenarios
- Analysis of existing systems
- Documentation Analysis
- Discourse analysis
- Task observation (Analysis),
- Brainstorming
- Ethnography
- Joint Application Development (JAD)
- Prototyping
- Pilot system
- Reverse Engineering

Classic Requirements Elicitation techniques

- These requirements elicitation techniques have been used for a long time. These are tested and proven methods
 - Interviews
 - Questionnaire
 - Introspection
 - Analysis of Existing Systems
 - Document Analysis
 - Observation and Social Analysis

Modern Requirements Elicitation Techniques

- Scenarios
- Brainstorming
- Prototyping
- Joint/Rapid Application Development
- Task Analysis
- Workshops
- Reverse Engineering
- Domain Analysis
- Goal modeling
- Etc.



Introspection

Introspection

- Requirement's analyst “imagines” what kind of system is required.
- Introspection requires the analyst to develop requirements based on what he or she believes the users and other stakeholders want and need from the system.
- Introspection is mainly used only as a starting point for other requirements elicitation efforts.
- Advantages:
 - It helps the other elicitation techniques. So, it is a good starting activity for requirement elicitation.
 - There are almost no costs of this technique.
- Disadvantages:
 - can be very inaccurate
 - unlikely to reflect stakeholder's goals
 - In case of using the introspection, the analyst should not only be familiar with the domain and goals of the system, but also should be expert in the business processes of the users.
 - In other words, this technique requires a huge experience of the requirement analyst.

Analysis of Existing Systems

Analysis of Existing Systems

- Useful when building **a new improved version** of an existing system
- Important to know:
 - What is used, not used, or missing
 - What works well, what does not work
 - How the system is used (with frequency and importance) and it was supposed to be used, and how we would like to use it

Why analyze an existing system?

- Users do not like the new system if it is too different or does not do what they want (risk of nostalgia for old system)
- To catch obvious possible improvements (features that are missing or do not currently work well)
- To find out which "legacy" features can/cannot be left out

Document Analysis

Document Analysis

- Document analysis is one of the most helpful elicitation techniques in understanding the current process. Documents like user manuals, software vendor manuals, process documents about the current system can provide the inputs for the new system requirements.
- Steps involved in document analysis are:
 1. Evaluating whether the existing system and business documents are appropriate to be studied.
 2. Analysing the documents to identify relevant business details.
 3. Reviewing and confirming identified details with subject matter experts.

Documents Analysis



Start with reading available documentation

User documents (manual, guides...)

Development documents

Requirements documents

Internal memos

Change histories

...



Of course, often these are out of date, poorly written, wrong, etc., but it's a good starting point



Discourse analysis

Use of words and phrases is examined in written or spoken language



Observation

Observation and Social Analysis



- Social analysis is also known as Observation. Observation is the method of collecting requirements by observing the people doing their normal work. This method is generally used to find the additional requirements needed by the user, when the user is unable to explain their expected requirements from the new product and problems with the existing product

Observations

- Can be supplemented later with **questionnaires**
 - Based on what you know now – the results of observation
 - To answer questions that need comparison or corroboration (confirmation)
 - To obtain some statistics from a large number of users (look for statistical significance!), e.g.:
 - *How often do you use feature X?*
 - *What are the three features you would most like to see?*
- Can be supplemented later with **interviews**
 - After getting a better idea of what is to be done, probably some questions require more detailed answers
 - You will not be wasting other people's time or your own
 - This is very labour intensive!

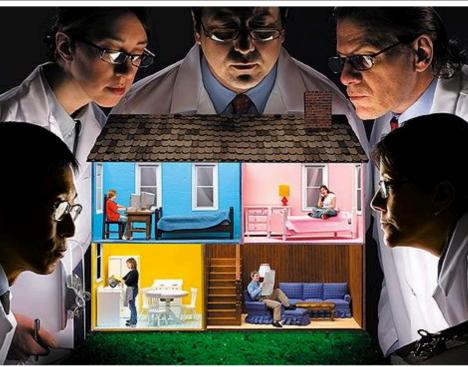
Social Analysis Types

- **Passive observations-** This social analysis is carried out without direct involvement of the observer in the society. The observation of the people work is carried out by recording using videotapes, video cameras and surveillance cameras. The documentation of the problems and the requirements are prepared from the recorded data.
- **Active observation-** This observation is carried out with the direct involvement of the observer. The people are provided with the new product prototype or existing product to perform the operations on the product. The observer provides the domain knowledge to the user to work with the product and he makes the report of the requirements of the people by observing their work with the product.
-

Social Analysis Types

- **Explanatory Observations**- In this type of observation, the users talk loudly, explaining what they are doing, while using the product. The observer takes notes using the explanation given by the user.
- **Ethnography** - In this method the observer is completely immersed in the society. The observer goes through in depth observation of the society and their works. There is no particular formula to carry out this method but it is time consuming and expensive method to gather the requirements.

Ethnography



- Comes from anthropology, literally means "writing the culture"
- People often find it hard to describe what they do because it is so natural to them
 - *Sometimes the best way to understand is to observe them at work*
- Ethnography is a technique from the social science which has proved to be valuable in understanding the actual work processes. Social scientists are trained in observation and work analysis
- Ethnography aims to describe the nature of those who are studied through writing.
- Ethnography attempts to discover social, human, and political factors, which may also impact requirements
- Essentially seeks to explore the human factors and social organization of activities. Studies have shown that work is often richer and more complex than is suggested by simple models derived from interviews

Ethnography

- Discoveries are made by observation and analysis, workers are not asked to explain what they do
 - Collect what is ordinary/what is it that people do (aim at making the implicit explicit)
 - Study the context of work and watch work being done
- Useful to discover:
 - What does a nuclear technician do during the day?
 - What does his workspace look like?

Ethnography

- These techniques are especially useful when addressing contextual factors such as usability, and when investigating collaborative work settings where the understanding of interactions between different users with the system is paramount.
- In practice, ethnography is particularly effective when the need for a new system is a result of existing problems with processes and procedures, and in identifying social patterns and complex relationships between human stakeholders.

Ethnography – Example (1)

- Sommerville et al. were involved in a project where they had to elicit the requirements of an air traffic control system
- They observed the air traffic controllers in action with the existing system
- Surprising observations
 - Controllers often put aircrafts on potentially conflicting headings with the intention of fixing them later
 - System generates an audible alarm when there is a possible conflict
 - The controllers close the alarms because they are annoyed by the constant warnings
- Incorrect conclusion
 - The controllers do not like audible alarms because they close them
- More accurate observation
 - The controllers do not like being treated like idiots

Advantages and Disadvantages of Ethnography

- Advantages of Ethnography
 - Ethnography is useful to collects the **quality attributes requirements** such as usability and efficiency etc which are necessary for the success of the project.
 - Ethnography much effective when to determine the social factors and patterns in the solution of the problem.
- Disadvantages of Ethnography
 - Ethnography fails in many cases because there are so much diverse communities of people belonging to different social and ethical sects.
 - It is difficult to analyze the social requirements of the people and hence the **psychologists are required** to provide their services

Task Analysis

Task analysis

- Task analysis employs a top-down approach where high-level tasks are decomposed into subtasks and eventually detailed sequences until all **actions and events** are described.
- The primary objectives of this technique are to construct a **hierarchy of the tasks performed by the users and the system**, and determine the knowledge used or required to carry them out.
- Task analysis provides information **on the interactions of both the user and the system** with respect to the tasks as well as a contextual description of the activities that take place.
- In most cases considerable effort is required to perform thorough task analysis, and it is important to establish what level of detail is required and when components of the tasks need to be explored further.

Advantages and Disadvantages of Task Analysis

- Advantages of Task Analysis
 - Task analysis provides the interaction of both user and the system with respect to some tasks that takes place.
 - Task analysis is used by the project manager to manage the user and system tasks.
- Disadvantages of Task Analysis
 - The task analysis requires a lot of effort as compared to interview.
 - The detail of level is mandatory in task analysis and hence it needs a lot of detail for the low-level tasks.



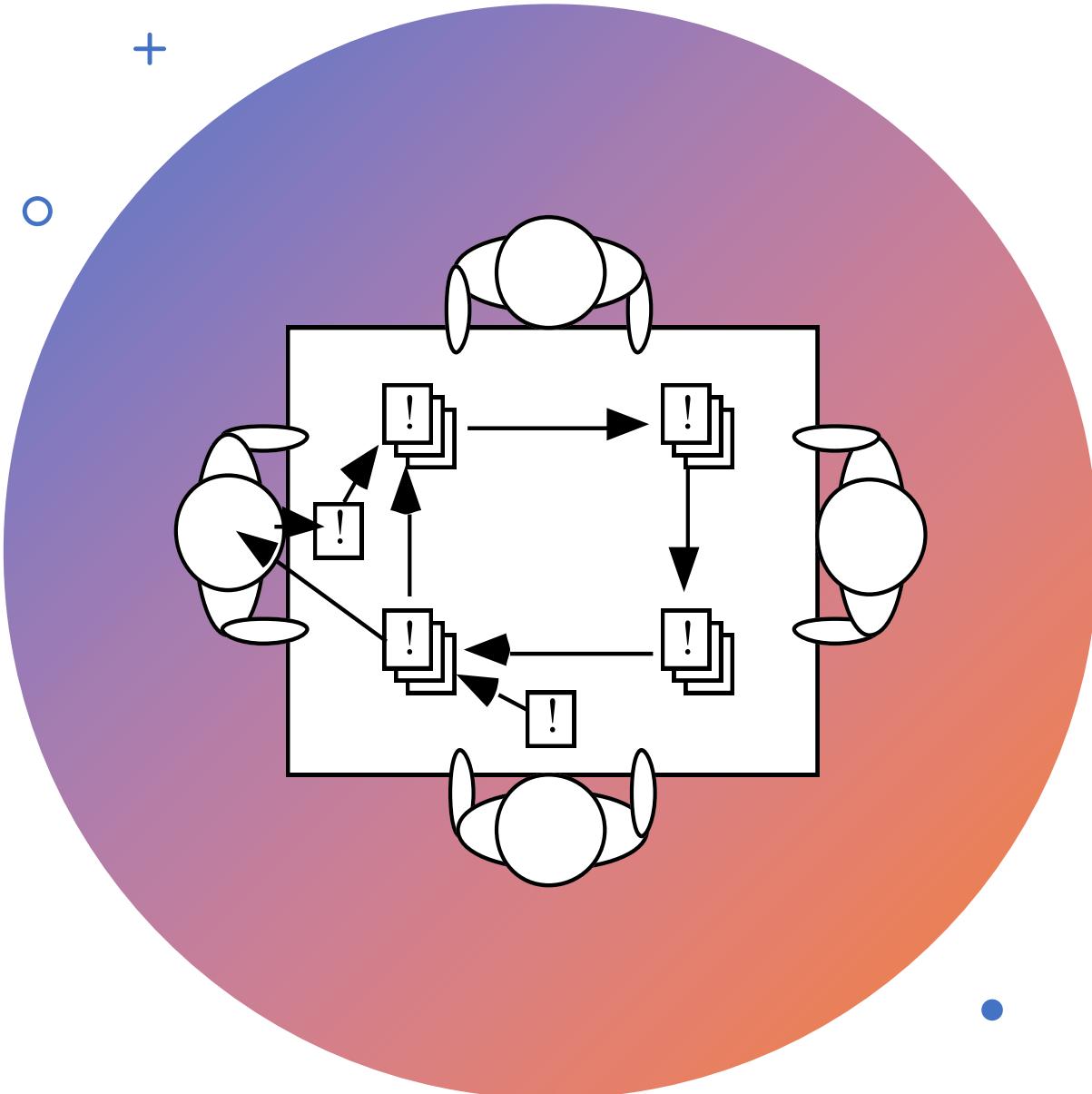
Brainstorming

Brainstorming

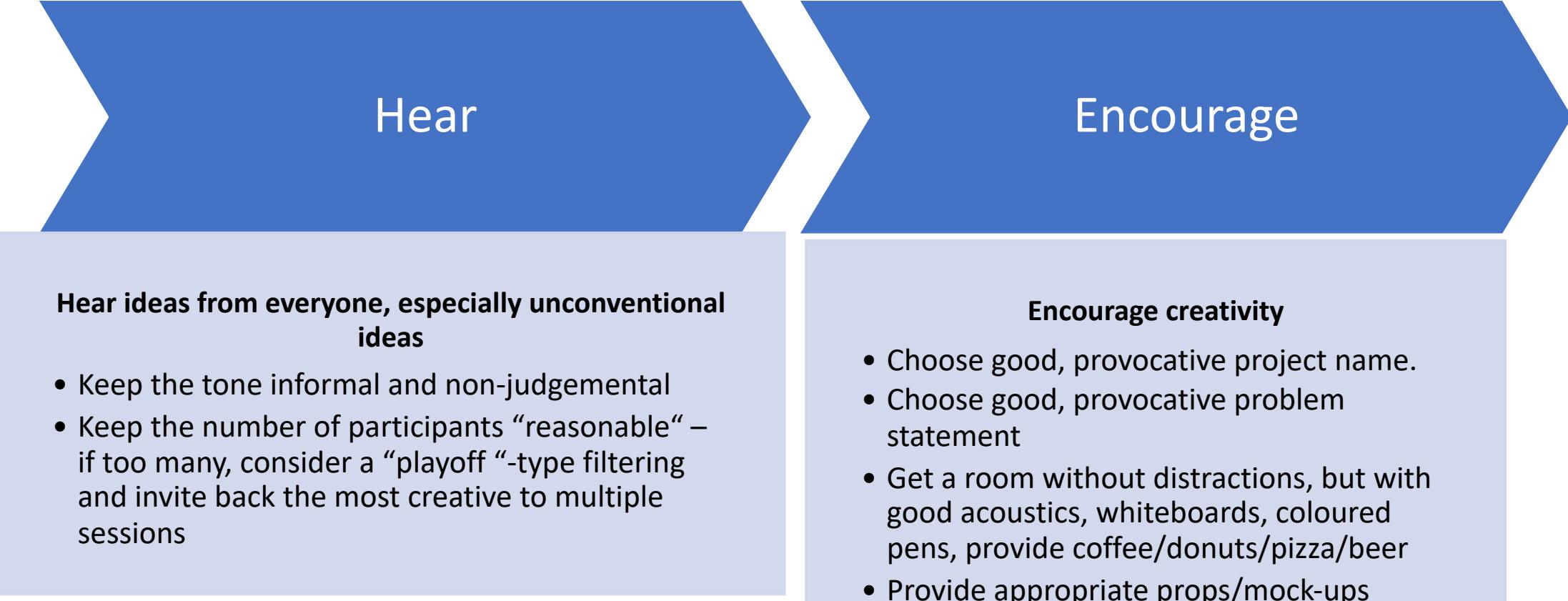
- Brainstorming is a process where participants from different stakeholder groups engage in informal discussion to rapidly **generate as many ideas as possible** without focusing on any one in particular.
- It is important to **avoid exploring or critiquing ideas in great detail**. It is not usually the intended purpose of brainstorming sessions to resolve major issues or make key decisions.
- This technique is often used to **develop the preliminary mission statement** for the project and target system.
- One of the advantages in using brainstorming is that it promotes freethinking and expression, and **allows the discovery of new and innovative solutions to existing problems**

Brainstorming

- To **invent new way** of doing things or when much is unknown
 - When there are few or too many ideas
 - Early on in a project particularly when:
 - There is little expertise for the type of applications
 - Innovation is important (e.g., novel system)
- Two main activities:
 - **The Storm:** Generating as many ideas as possible (quantity, not quality) – wild is good!
 - **The Calm:** Filtering out of ideas (combine, clarify, prioritize, improve...) to keep the best one(s)
 - may require some voting strategy



Brainstorming – Objectives



Hear

Hear ideas from everyone, especially unconventional ideas

- Keep the tone informal and non-judgemental
- Keep the number of participants “reasonable” – if too many, consider a “playoff” -type filtering and invite back the most creative to multiple sessions

Encourage

Encourage creativity

- Choose good, provocative project name.
- Choose good, provocative problem statement
- Get a room without distractions, but with good acoustics, whiteboards, coloured pens, provide coffee/donuts/pizza/beer
- Provide appropriate props/mock-ups

Advantages of Brainstorming

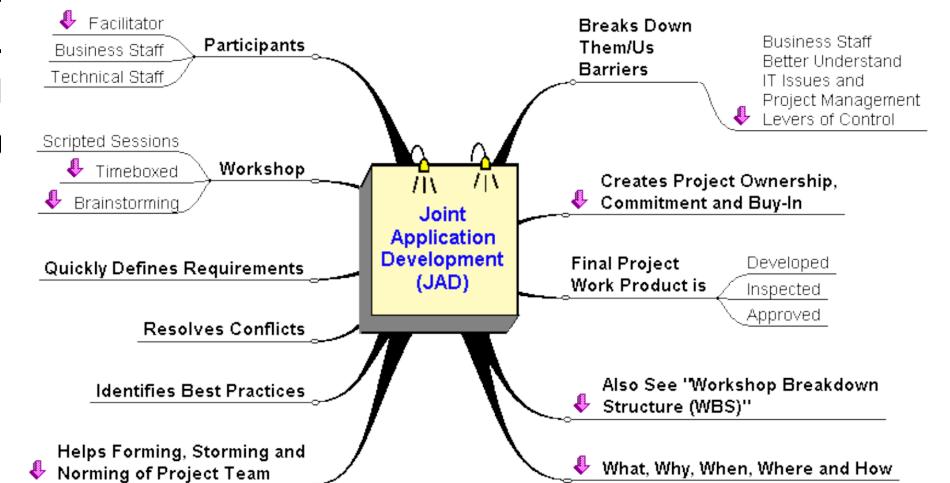
- **Advantages of Brainstorming**

- Brainstorming are mostly used for the innovative sort of projects where each participant provides his or her own ideas after their personal research about the project to be started.
- This technique is often used make the key decisions about the requirements of the project.
- It promotes free thinking and expression of ideas.
- Brainstorming provides the innovative ideas about the project to be developed.

Joint Application Development (JAD)

Joint Application Development (JAD)

- Developed at IBM in the 1970s
 - Lots of success stories
- "Structured brainstorming", IBM-style
 - Full of structure, defined roles, forms to be filled out...
- Joint Application Development (JAD) technique is an extended session to the workshop. In the JAD session stakeholders and project team works together to identify the requirements.
- These sessions allow the business team to gather and ~~consolidate large amounts of information~~
Identification of stakeholders is the critical to the overall success of the JAD session. The team includes business process owners, client representatives or managers, business analysts, project managers, IT experts (developers, quality assurance, etc.)
- A more structured and intensive brainstorming approach
- Several activities and six (human) roles to be played
- JAD session may last few days



Joint Application Development – Four Main Principles

- Effective use of group dynamics
 - Facilitated and directed group sessions to get common understanding and universal buy-in
- Use of visual aids
 - To enhance understanding, e.g., props, prepared diagrams
- Defined process
 - I.e., not a random hodgepodge (confusing mixture.)
- Standardized forms for documenting results

Joint Application Development (JAD)

- Used for making decisions on different aspects of a project
- Any process where consensus-based decision making across functional areas is required, e.g.,
 - Planning a project
 - Defining requirements
 - Designing a solution

Joint Application Design – Activities



Preparation

Pre-session Planning

Pre-work



Working Session



Summary

Follow-up

Wrap-up

Joint Application Design

– Pre-session Planning

Preparation is essential – this is not an informal session

Evaluate project

- Identify contentious issues and scope of JAD session

Select JAD participants

Create preliminary agenda

Determine deliverables for the working session

Enable participants to prepare for the session

The 6 “P”s

1. **Purpose** - Why do we do things? (Goals, needs, motivation)
2. **Participants** - Who is involved? (People, roles, responsibilities)
3. **Principles** - How do we function? (Guidelines, working agreements, ground rules)
4. **Products** - What do we create? (Deliverables, decisions, plans, next steps)
5. **Place** - Where is it located? (Venue, logistics)
6. **Process** - When do we do what? (Activities, sequence)

Joint Application Design – Pre-work

Gather information

Clear schedules for the working session

Refine session agenda

Finalize pre-session assignments

Prepare material for session (flip-charts,
presentations, markers, pizza...)

Joint Application Design – Working Session



Set-up stage

Session leader welcomes participants, presents task to be discussed, establishes rules and what is on/off topic...



Generate common understanding

Brainstorming...



Achieve consensus on decisions



Generate ownership of results



Create the deliverables (using standard JAD forms)



Identify open issues and questions



Joint Application Development – Follow-up and Wrap-up

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

Joint Application Development – Roles

- **Session leader**
 - Organizer, facilitator, JAD expert
 - Good with people skills, enthusiastic, sets tone of meeting
- **Analyst**
 - Produces official JAD documents, experienced developer who understands the big picture, good philosopher/writer/organizer
- **Executive sponsor**
 - Manager who has ultimate responsibility for product being built
 - Provides strategic insights into company's high-level goals/practices, makes executive decisions later on as required
- **User representatives**
 - Selection of knowledgeable end-users and managers
 - Come well-prepared with suggestions and ideas of needs, will brainstorm for new or refined ideas, eventually review completed JAD documents
- **Information system representatives**
 - Technical expert on ISs
 - Helps users think big, know what is easy/hard/cheap/expensive, mostly there to provide information rather than make decisions
- **Specialists**
 - Technical expert on particular narrow topics, e.g., security, application domain, law, UI issues...

Advantages and Disadvantages of JAD

- **Advantages of JAD**

- JAD provides the rapid decision making about the problem and also the solution of the problem.
- JAD handles the rapid changeability in requirements.
- JAD provides the well formatted structured approach.
- JAD provides the direct communication between all stakeholders of the project.

- **Disadvantages of JAD**

- As JAD should provide the rapid solution so hence sometime this technique is unable to provide the exhaustive validation in so least amount of time.
- JAD team requires huge experience and expertise in the problem domain



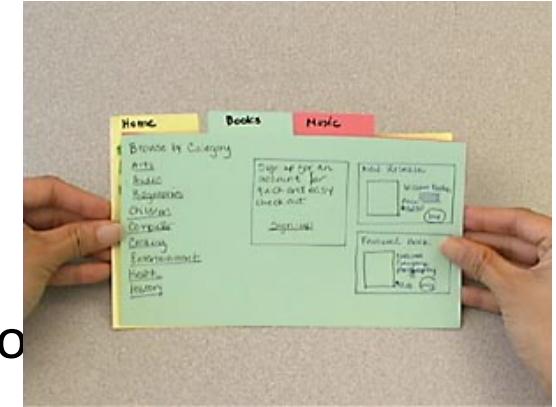
Prototyping

Prototyping

- Screen mock-ups can support the requirement gathering process, when introduced at the correct time. Mock-ups help stakeholders visualize the functionality of a system. This can be an advantage to business analysts and stakeholders since this allows them to identify gaps/problems early on.
- A software requirements prototype is a mock-up or partial implementation of a software system
 - Helps developers, users, and customers better understand system requirements
 - Helps clarify and complete requirements
 - Provides early response to “I’ll know it when I’ll see (or won’t see) it” attitude
 - Effective in addressing the “Yes, But” and the “Undiscovered Ruins” syndromes
 - Helps find new functionalities, discuss usability, and establish priorities
- Prototyping is effective in resolving uncertainties early in the development process
 - Focus prototype development on these uncertain parts
 - Encourages user participation and mutual understanding

Prototyping – Realizations

- Prototypes can take many forms:
 - Paper prototypes (see <http://www.paperprototyping.com/>)
 - Prototype on index card
 - Storyboard
 - Screen mock-ups
 - Interactive prototypes
 - Using high-level languages (e.g., Visual Basic, Delphi, Prolog)
 - Using scripting languages (e.g., Perl, Python)
 - Using animation tools (e.g., Flash/Shockwave)
 - Models (executables)
 - Pilot systems
 - ...



Prototyping – Types

- **Horizontal**: focus on one layer – e.g., user interface
- **Vertical**: a slice of the real system
- **Evolutionary**: turned into a product incrementally, gives users a working system more quickly (begins with requirements that are more understood)
 - intended to deliver a workable system quickly to the customer.
 - Therefore, the requirements which should be supported by the initial versions of this prototype are those which are well-understood and which can deliver useful end-user functionality. It is only after extensive use that poorly understood requirements should be implemented.
- **Throw-away**: less precise, thrown away, focusing on the less well-understood aspects of the system to design, designed to elicit or validate requirements.
 - intended to help elicit and develop the system requirements.
 - The requirements which should be prototyped are those which cause most difficulties to customers and which are the hardest to understand. Requirements which are well-understood need not be implemented by the prototype.

Prototyping – Fidelity (1)

- Fidelity is the extent to which the prototype is real and (especially) reactive
- Fidelity may vary for throw-away prototypes
- **High-fidelity**
 - Applications that "work" – you press a button and something happens
 - Often involves programming or executable modeling languages
 - Advantages:
provides an understanding of functionality, reduce design risk, more precise verdicts about requirements
 - Disadvantages:
takes time to build, more costly to build, sometimes difficult to change, false sense of security, often focuses on details rather than on the goals and important issues

Prototyping – Fidelity (2)

Low-fidelity

- It is not operated – it is static
- Advantages:
easy and quick to build, cheaper to develop, excellent for interfaces, offers the opportunity to engage users before coding begins, encourage creativity
- Disadvantages:
may not cover all aspects of interfaces, are not interactive, may seem non-professional in the eyes of some stakeholders (sigh!)

Prototyping – Risks and Problems

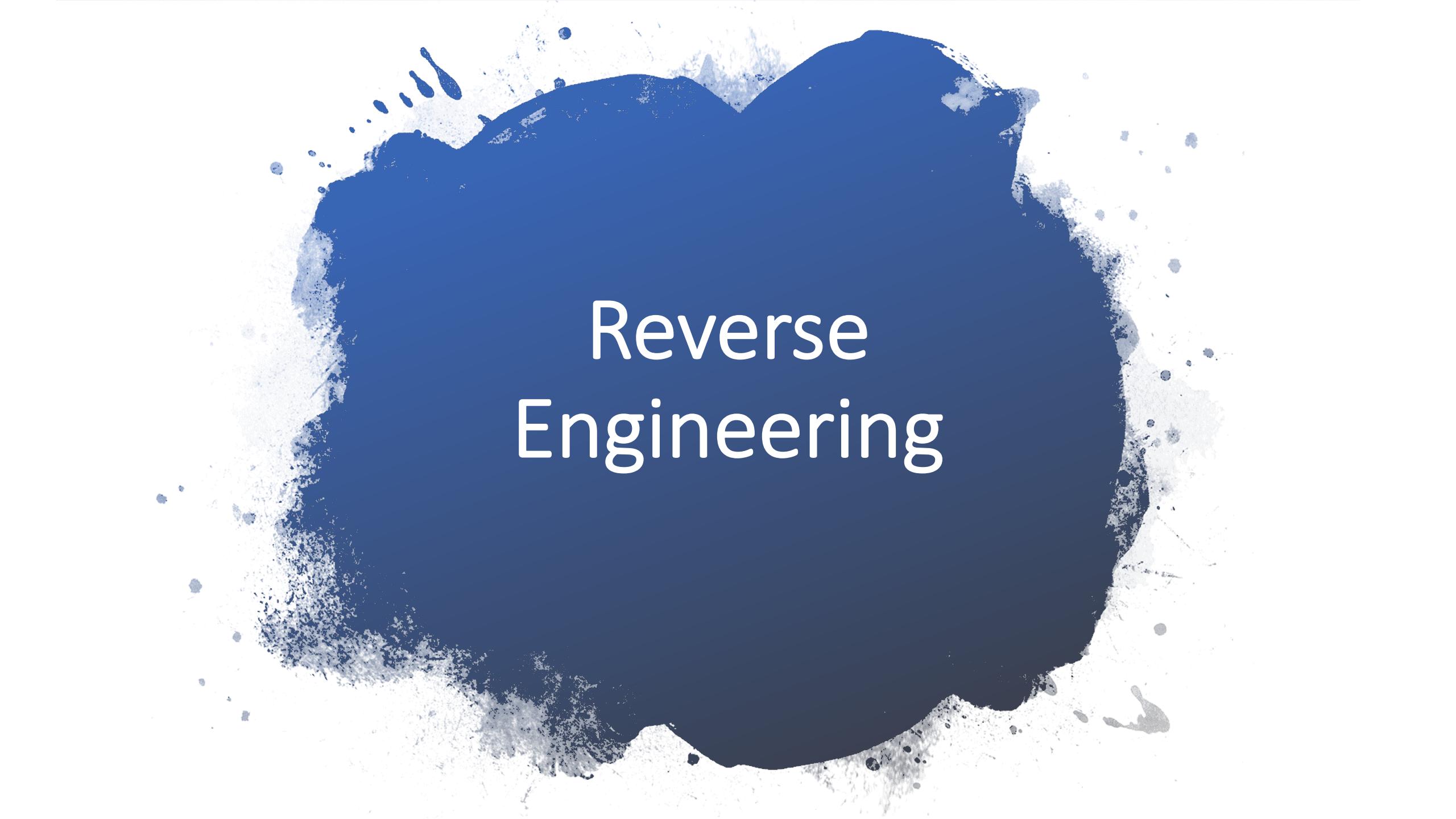
- Prototypes that focus on user-interface tends to lose the focus of demonstrating/exploring functionality
- Prototypes can bring customers' expectations about the degree of completion unrealistically up
- Do not end-up considering a throwaway prototype as part of the production system
 - Always clearly state the purpose of each prototype before building it
- Training costs - prototype development may require the use of special purpose tools
- Development costs - depend on the type of prototype being developed
- Extended development schedules - developing a prototype may extend the schedule although the prototyping time may be recovered because rework is avoided
- Incompleteness - it may not be possible to prototype critical system requirements



Workshop

Workshops

- Workshops comprise a group of users or stakeholders working together to identify requirements. A requirement workshop is a structured way to capture requirements. Workshops are used to scope, discover, define, and prioritize requirements for the proposed system.
- They are the most effective way to deliver high-quality requirements quickly. They promote mutual understanding and strong communication between users or stakeholders and the project team



Reverse Engineering

Reverse Engineering

This elicitation technique is generally used in migration projects. If an existing system has outdated documentation, it can be reverse engineered to understand what the system does. This is an elicitation technique that can extract implemented requirements from the system.

There are two types of reverse engineering techniques.

- **Black box reverse engineering:** The system is studied without examining its internal structure (function and composition of software).
- **White box reverse engineering:** The inner workings of the system are studied (analysing and understanding of software code)

Comparison of Data-Gathering Techniques¹

Technique	Good for	Data Types	Pros.	Cons.
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	Possibility of dominant characters
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

Future Directions in Requirements Elicitation

- Despite the successes and progress to date, many important topics remain open for investigation with respect to providing appropriate techniques, approaches, and tools for requirements elicitation, including specific assistance for novice analysts, cognitive support through intelligent tools, and methods that involve direct inter-action with stakeholders.
- Potential requirements elicitation research areas ::
 - Increasing the awareness and education of analysts and stakeholders in industry
 - Developing guidelines for technique selection and managing the impact of factors on the process
 - Investigating ways of collecting and reusing knowledge about requirements elicitation
 - Integration and use of new technologies including web and agent-based architectures into the next generation of support tools
 - Produce and publish case studies and industrial experience reports on how requirements elicitation contributed to successes and failures of projects
 - Exploring how requirements elicitation activities relates to new and developing fields of software engineering such as agent-based systems, agile development methodologies, and web systems