**Nguyễn Tuấn Khang - 21520963**

**Nguyễn Đức Thành Duy - 21520774**

Lab 1.1.

Answer the below questions in a text file (md, docx, txt) named Lab1.1 :

## What criteria should be used in choosing an appropriate requirements engineering tool?

1. Functionality: The tool should support all of the requirements engineering activities that your organization needs to perform. This may include requirements elicitation, analysis, specification, validation, and management.
2. Ease of use: The tool should be easy to learn and use, both for experienced and inexperienced users.
3. Scalability: The tool should be able to scale to meet the needs of your organization, regardless of its size or complexity.
4. Integration: The tool should be able to integrate with other tools that your organization uses, such as project management tools, configuration management tools, and testing tools.
5. Cost: The tool should be affordable and within your budget.

## Are there any drawbacks to using certain tools in requirements engineering activities?

1. Complexity: Some tools can be complex and difficult to learn and use.
2. Cost: Some tools can be expensive, especially for small or medium-sized organizations.
3. Vendor lock-in: Some tools are proprietary and require you to purchase additional products or services from the vendor.
4. Lack of support: Some tools are open source and may not have adequate support from the community.

## When selecting an open-source tool, what characteristics should you look for?

1. Active community: The tool should have an active community of users and developers who can provide support and contribute to the development of the tool.
2. Maturity: The tool should be mature and well-tested.
3. Documentation: The tool should be well-documented, so that users can learn how to use it and troubleshoot problems.
4. Licenses: The tool should be licensed under a permissive open source license, such as the MIT License or the Apache License 2.0.

## How can tools enable distributed, global requirements engineering activities? What are the drawbacks in this regard?

Some ways in which tools can enable distributed, global requirements engineering activities:

1. Collaboration tools: Tools such as wikis, chat tools, and video conferencing tools can enable team members to collaborate on requirements engineering activities, even if they are located in different parts of the world.
2. Version control tools: Tools such as Git and SVN can enable team members to track changes to requirements and merge their work together.
3. Requirements management tools: Tools such as DOORS and Jama can enable team members to store, manage, and track requirements in a central repository.

Some drawbacks to using tools for distributed, global requirements engineering activities:

1. Communication challenges: It can be difficult to communicate effectively with team members who are located in different parts of the world.
2. Cultural differences: It is important to be aware of cultural differences when working with team members from different cultures.
3. Time zone differences: It can be difficult to schedule meetings and coordinate work with team members who are in different time zones.

## If an environment does not currently engage in solid requirements engineering practices, should tools be introduced?

## If an environment does not currently engage in solid requirements engineering practices, it is still possible to introduce tools. However, it is important to first implement the basic principles of requirements engineering, such as requirements elicitation, analysis, specification, validation, and management. Once the basic principles are in place, tools can be introduced to help automate and improve the requirements engineering process

## .What sort of problems might you find through a traceability matrix that you might not see without one?

1. Inconsistencies: A traceability matrix can help you to identify inconsistencies between requirements and other artifacts, such as design documents and test cases.
2. Missing requirements: A traceability matrix can help you to identify missing requirements, i.e., requirements that are not implemented in the system.
3. Unnecessary requirements: A traceability matrix can help you to identify unnecessary requirements, i.e., requirements that are not used by the system.

## How is AI being proposed for knowledge acquisition and representation in requirements specifications?

1. Inconsistencies: A traceability matrix can help you to identify inconsistencies between requirements and other artifacts, such as design documents and test cases.
2. Missing requirements: A traceability matrix can help you to identify missing requirements, i.e., requirements that are not implemented in the system.
3. Unnecessary requirements: A traceability matrix can help you to identify unnecessary requirements, i.e., requirements that are not used by the system.