**Bùi Thị Hoàng Giang – 21520794**

**Nguyễn Hoàng Hy – 21520946**

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

* **Requirement Types**: Different projects may have various types of requirements, such as functional, non-functional, user stories, use cases, etc. Ensure that the tool can handle the types of requirements you need to manage.
* **Ease of Use**: The tool should be user-friendly and easy for team members to learn and use. A complex or unintuitive tool can lead to resistance and lower productivity.
* **Cost and Licensing**: Understand the pricing structure of the tool, including licensing, support, and any additional costs. Evaluate whether the tool's pricing aligns with your budget.
* **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.
* **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.
* **Reporting and Documentation**: Check if the tool offers robust reporting and documentation capabilities. Good reporting features can help in project management and communication.

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

* **Customization**: While customization is an advantage, it can also be a drawback if it requires significant effort. Some tools may be too rigid or require extensive customization to adapt to specific project requirements.
* **Overhead**: Introducing a new tool can add overhead to the project in terms of training, administration, and maintenance. It may take time to set up and configure the tool to suit your project's needs.
* **Integration Challenges**: Ensuring seamless integration with other project management and development tools can be a challenge. Poor integration can lead to data silos and manual data transfer.
* **Performance Issues**: In some cases, complex tools may suffer from performance issues, particularly when handling a large number of requirements, documents, or users.

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

* **Documentation**: Comprehensive and well-maintained documentation is crucial for understanding how to use the tool effectively. Look for documentation that includes installation guides, tutorials, user manuals, and API references.
* **Updates and Maintenance**: Check the frequency of updates and releases. A tool that is regularly updated is more likely to stay current and address issues and vulnerabilities promptly.
* **Customization**: The ability to customize and extend the tool to meet your specific project requirements is essential. Evaluate how flexible the tool is in terms of configuration and plugin support.
* **Performance**: Evaluate the tool's performance, especially when dealing with large datasets or high loads. Ensure that it meets your project's performance requirements.
* **Use Cases**: Determine whether the tool has been successfully used in projects similar to yours. Case studies and testimonials can provide insight into its effectiveness.

*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:

* **Cloud-Based Tools**: Cloud-based requirements management tools (e.g., Jira, Trello, ReqView) offer accessibility from anywhere with an internet connection, providing a common platform for requirements documentation and tracking
* **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.
* **Communication Tools:** Messaging and video conferencing tools (e.g., Slack, Microsoft Teams, Zoom) facilitate real-time communication and discussions, making it easier to clarify requirements and address questions among global team members.
* **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:

* **Communication challenges**: It can be difficult to communicate effectively with team members who are located in different parts of the world.
* **Cultural differences**: It is important to be aware of cultural differences when working with team members from different cultures.
* **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?*

* **Incomplete Requirements**: A traceability matrix can expose gaps in requirements coverage. If certain requirements are not linked to any design or test artifacts, it suggests that those requirements may be missing or not adequately addressed.
* **Orphaned Artifacts**: On the flip side, it can identify design or test artifacts that have no corresponding requirements. This may indicate that some work has been done without clear requirements or that certain artifacts are no longer needed.
* **Ambiguities and Inconsistencies**: When you trace a requirement to multiple design or test artifacts, inconsistencies and contradictions in the requirement's interpretation become evident. This allows you to address ambiguities and ensure that everyone is on the same page.

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

AI is being proposed for knowledge acquisition and representation in requirements specifications in various ways to enhance the requirements engineering process. Here are some approaches and techniques where AI plays a role:

* **Natural Language Processing (NLP):** NLP techniques are used to analyze natural language documents, such as stakeholder requirements, user stories, or domain knowledge, and extract structured information from unstructured text. AI-powered NLP can automatically identify entities, relationships, and concepts, turning textual information into structured data that can be used to build requirements models.
* **Semantic Analysis**: AI can perform semantic analysis to identify and understand the meaning of words, phrases, and sentences in requirements documents. This helps in disambiguating terms and ensuring that requirements are clear and well-defined.
* **Pattern Recognition**: AI systems can recognize patterns and recurring themes in requirements specifications. By identifying common patterns, AI can assist in generating templates or suggesting existing requirements that might be applicable to the current project.

AI in requirements engineering is not meant to replace human experts but to enhance their capabilities, reduce errors, and improve the efficiency and quality of the requirements engineering process. By leveraging AI for knowledge acquisition and representation, organizations can better manage the complexity of modern software projects and deliver high-quality software that meets stakeholder needs.