1.1.

**What criteria should be used in choosing an appropriate requirement engineering tool? (Page 207)**

The capabilities of requirements engineering of any tools are organized along the following areas: elicitation, analysis, specification, verification and validation, management, and other capabilities.

Sud and Arthur (2003) evaluated a few requirements management tools using the following dimensions:

* Requirements traceability mechanism
* Requirements analysis mechanism
* Security and accessibility mechanism
* Portability and backend compatibility
* Configuration management approach
* Communication and collaboration mechanism
* Change management support.
* Online publishing support
* Usability features such as word processor compatibility
* SRS documentation format

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

* Some tools can be overly complex, making it difficult for team members, especially beginners, to navigate and use them effectively. This complexity can slow down the requirement gathering process.
* Many advanced requirements engineering tools come with a hefty price tag. For students or smaller organizations, these costs might be prohibitive, limiting access to necessary features.
* Sometimes, tools might not integrate well with other software used in the development process. This lack of integration can lead to data inconsistencies and communication gaps between team members.
* Tools require maintenance and updates. If a tool isn't well-maintained, it can become outdated, leading to potential security vulnerabilities or compatibility issues with newer technologies.
* Some tools might not be flexible enough to accommodate unique or changing project requirements. This lack of flexibility can stifle creativity and hinder the development of innovative solutions.

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

* The tool should provide a range of features and capabilities that align with your requirements engineering activities. It should support tasks such as requirements elicitation, documentation, analysis, traceability, and collaboration.
* Look for a tool that allows ​customization to fit your specific requirements engineering processes and methodologies. This may include the ability to define custom fields, templates, workflows, and reports.
* ​Consider whether the tool can seamlessly integrate with other tools commonly used in requirements engineering, such as version control systems, project management software, or issue tracking systems. Integration capabilities can streamline your workflow and enhance collaboration.
* Ensure the tool provides capabilities for generating clear and comprehensive documentation and reports. These outputs should be customizable and easily shareable with stakeholders, fostering effective communication and understanding.
* Consider the tool's ability to handle large and complex sets of requirements without compromising performance. It should be able to scale with your project's growing needs and accommodate increasing numbers of users and requirements.
* Open-source tools should prioritize security and data privacy. Consider the measures in place to protect sensitive requirements information and ensure compliance with relevant regulations and standards.
* Assess the longevity of the tool and the maintenance efforts invested by its developers. A regularly updated tool ensures ongoing support, bug fixes, and feature enhancements, minimizing the risk of compatibility issues or abandonment.

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

* Tools facilitate asynchronous collaboration, allowing team members from different time zones to work on requirements at their convenience. This ensures continuous progress without waiting for everyone to be online simultaneously.
* Many tools offer real-time collaboration features like live editing and commenting, enabling instant communication and feedback, even if team members are geographically dispersed.
* Tools provide versioning capabilities, allowing changes to be tracked, compared, and reverted if necessary. This ensures that everyone is working with the latest version of the requirements document.
* Online tools provide a central location for storing requirements, ensuring that all team members have access to the most recent documentation. This reduces the risk of working with outdated or conflicting information.
* Tools allow administrators to set access permissions, ensuring that sensitive requirements are only accessible to authorized team members. This helps maintain data security and confidentiality.
* Requirements tools often offer traceability features, allowing team members to track changes and understand the impact of requirements on other project aspects. This is crucial for managing complex projects with interconnected requirements.

*Drawbacks:*

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

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

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