1. **What criteria should be used in choosing an appropriate requirement engineering tool?**

There are many commercial and open - source tools, and it is important to carefully evaluate the features of any tool before adopting it in the enterprise.

Carrillo de Gea et al. (2021) provided an alternative list of 17 items to be used in the evaluation of RE tool capabilities:

* Organization of requirements with metadata, attributes, and reuse
* Reports, database queries, and open interface language
* Internal checks, that is, consistency, dependencies, and history.
* Traceability support, that is, drag and drop (horizontal and vertical)
* Providing support for reuse
* Remote working, cloud only
* Multiple views of requirements
* Performance
* Collaboration, workflow management
* Easily adapted and integrated into business processes.
* Federation and notification with ALM/PLM tools
* Export/import with standard formats
* Macros for repeated commands
* Training and learning curve effort.
* Agile, CI/CD, and DevOps
* Intelligent support
* Scalability

This checklist (or an adapted version) and an appropriate consensus management approach (e.g., Wideband Delphi, AHP) can be used to select the right tool for a team or enterprise.

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

**Yes**, there may be some drawbacks include:

**Complexity**: Some tools can be complex and difficult to use. They may require extensive training and expertise to operate effectively.

**Lack of integration**: Some tools may not integrate well with other software development tools such as project management, testing, and version control tools.

**Limited collaboration**: May not provide adequate collaboration features such as commenting, notifications, and real-time updates.

**Security risks**: Some tools may pose security risks due to outdated security protocols and lack of access control mechanisms.

**High cost**: Can be expensive to maintain and upgrade. They may require significant investment in licensing fees, maintenance costs, and training costs.

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

There are hundreds of thousands of open - source projects many of which are full - featured requirements management tools, and you are encouraged to turn first to open - source repositories to look for tools before purchasing or trying to develop them from scratch. There are also utilities or resources for requirements engineering.

When selecting an open-source tool, there are several characteristics you should consider:

* **Functionality and flexibility**: The tool should be able to perform the tasks you need it to do, and it should be flexible enough to adapt to your changing needs over time.
* **Security**: Open-source tools are often more secure than proprietary software because they are subject to peer review and can be audited by anyone. However, you should still look for tools that have a good track record of security and that are regularly updated to address any vulnerabilities.
* **Maintenance and support**: You should look for tools that are actively maintained and have a strong user community. This will ensure that any issues or bugs are quickly addressed, and that you have access to resources and support when you need it.
* **User community**: A strong user community is important because it means that there are other people using the tool who can help answer questions, provide guidance, and share best practices.
* **Documentation**: Good documentation is essential for any tool, but it is especially important for open-source software. Look for tools that have clear and comprehensive documentation that is easy to understand.
* **Licensing**: Finally, you should pay attention to the licensing terms of any open-source tool you consider using. Make sure that the license is compatible with your intended use case, and that you understand any restrictions or requirements associated with the license.

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

RE tools can enable distributed, global requirements engineering activities by enabling collaboration, communication, and coordination among stakeholders who are geographically dispersed. Some of the benefits of using RE tools in a distributed process are:

-Provide a central and accessible repository for storing and sharing requirements, as well as their attributes, dependencies, and traceability links

-Support the creation and maintenance of various types of requirements artifacts, such as use cases, user stories, scenarios, models, diagrams, etc.

-Enable the management of requirements changes by providing features such as version control, configuration management, change tracking, impact analysis, etc.

-Integrate with other tools in the software development lifecycle, such as project management tools, design tools, testing tools, etc.

-Support the communication and negotiation of requirements among stakeholders by providing features such as comments, annotations, reviews, feedback, etc.

-Support the verification and validation of requirements by providing features such as test case generation, test execution, test coverage, test results, etc.

However, RE tools are also having some drawback such as tool selection, tool integration, tool adoption, tool maintenance, etc.