1. Criteria should be used in choosing an appropriate requirements engineering tool.

* 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

These evaluation dimensions can be used by engineers to compare various commercial and open-source requirements management tools prior to adoption.

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

* **Licensing:** Open source doesn't mean free. While you have free access to the source code, you may need to comply with certain conditions depending on the license. Therefore, you should always check the license of the open-source tool and ensure it aligns with your intended use. If you plan to modify or distribute the software, make sure the license permits these actions.
* **Community Support:** Open-source tools are often supported by communities of developers. A strong, active community can be a good sign of a healthy project. Look for frequent updates, a large number of contributors, and active discussions. These signs indicate that the tool is being actively maintained and that help will likely be available if you encounter problems.
* **Documentation:** Good documentation is crucial for understanding how to use and adapt the tool. It can also provide insight into the tool's reliability and the community's commitment to supporting new users. Check if the tool has clear, comprehensive, and up-to-date documentation.
* **Stability and Maintenance:** Check the tool's update history. Frequent updates can indicate active maintenance, which can lead to better stability and security. However, if a tool is updated too frequently with breaking changes, it could cause instability in your own project.
* **Security:** Open-source tools can be more secure because their code is publicly available for scrutiny, which can lead to the early detection and fixing of security vulnerabilities. However, not all open-source projects actively manage security issues, so it's important to check the project's history of dealing with security issues.
* **Features and Functionality:** Ensure the tool provides the features and functionality you need. Consider how well it meets your requirements now and whether it can be adapted to meet your future needs.

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

On one hand, tools can provide structure and support for requirements engineering activities, helping to establish good practices. They can facilitate collaboration, automate repetitive tasks, and help manage and track requirements. This can lead to improved efficiency, consistency, and quality in the requirements engineering process.

On the other hand, the introduction of tools can also present challenges. There can be a learning curve associated with using new tools, which requires time and training. If the tools are not intuitive or user-friendly, this can cause confusion and frustration among the team members. There can also be resistance to change, especially if the team members are used to their current ways of working. Furthermore, the tools may not fully meet the needs or expectations of the team members, or they may not integrate well with the other tools or systems used in the organization.

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

* **Commonsense Knowledge Acquisition and Representation:** AI can be used to automatically extract commonsense knowledge from text and contextualize it quantitatively and qualitatively. Large-scale models, such as BERT, GPT-2, and T5, can learn to implicitly represent an abundance of commonsense knowledge from reading the web. This knowledge can then be extracted through carefully-designed language prompting, or through fine-tuning on knowledge graph tuples. This can be particularly useful in requirements engineering to understand the implicit assumptions and expectations of the stakeholders, and to ensure the alignment and coherence of the requirements.
* **Knowledge-Enhanced Machine Learning:** AI can be used to imbue systems with the ability to reason about moral value, blame, and intentionality. This can be crucial in requirements engineering to understand the ethical implications of the requirements, and to guide the decision-making and trade-off analysis in the requirements engineering process.
* **Knowledge Acquisition for Second Generation Expert Systems:** AI can be used to elicit, model, and represent human problem-solving knowledge. This can be particularly useful in requirements engineering to understand the problem domain, the problem constraints, and the problem solutions. This can also be useful to guide the elicitation, modeling, and representation of the requirements.
* **Knowledge Acquisition and Design Using Semantics and Perception for Autonomous Robots:** AI can be used to acquire and design knowledge based on semantics and perception. This can be particularly useful in requirements engineering for autonomous robots to understand their sensory inputs, their operational contexts, and their behavioral outputs. This can also be useful to guide the specification, validation, and verification of the requirements.