**Case Entry**

1. MUST be able to support multiple and customizable case types (e.g. Bugs, Features, Work Items, etc)
2. MUST be able to support a flexible workflow, meaning we can design our own lifecycle for each case and identify valid states that each case can be in and the available actions (transitions to other states) for each state.
3. MUST be able to support different workflows for different case types (e.g. the workflow for a Bug may be very different than that for a Feature Request)
4. MUST be able to customize the available fields for every case type.
5. MUST be able to support unique sets of fields for each case type (e.g. the available fields in a Bug case may be very different than the fields in a Feature Request case)
6. MUST be able to support attachments (and without a small file size limitation – up to 100 MB)
7. MUST be able to link cases based on parent/child relationships, duplicate, prerequisite dependency, file change dependency, or reference to another case.
8. SHOULD be capable of tracking cases for the current FAR/PMR DB.
9. SHOULD support the ability to require additional info to be entered (e.g. via  a pop-up window) during specific state transitions.
10. SHOULD support multiple data types for each field, including single-select pulldown lists, multi-select pulldown lists, radio button selections, and free-form text fields.
11. SHOULD support an enhanced text field for comments which allows formatted texts, pictures and graphics, tables, etc to be inserted into the comments.
12. SHOULD be able to support customizing a subset of mandatory fields that must be set for every new case being opened; other fields may be set to be optionally selected or filled in
13. SHOULD be able to clone a case.
14. SHOULD track all changes in some sort of event log.
15. SHOULD provide a way to record the risk associated with a case.

**Integration with other tools**

1. MUST support integration with major source control tools (e.g. SVN, GIT)
2. MUST support an open set of APIs whereby other tools and applications can access the case management database directly.
3. MUST contain an integrated peer review tool or be able to link seamlessly to 3rd party peer review tools
4. SHOULD contain web based documentation support (like wiki) or be able to link to 3rd party tool.
5. SHOULD integrate with a discussion forum tool.

**Notifications:**

1. MUST support configurable email notifications (e.g. State changes, required actions, etc…)
2. MUST support the ability to subscribe to cases.

**Reporting / Query / Search Mechanism**

1. MUST support a very flexible and easy-to-use means to create customized filters for viewing subsets of cases.
2. MUST be able to save and share custom filters
3. MUST support a rich search function to find pervious occurrences of an issue.
4. MUST be capable of accessing the backend database for advance queries and joins with data not in the case management tool.
5. SHOULD support a rich set of built-in reports and graphs for visualizing case info.
6. SHOULD support the ability to create a user customable Dash Board
7. SHOULD allow organization of Filters.
8. SHOULD be able to export report in a CSV format.

**Work / Project Management**

1. MUST support an embedded capability to visually view and manage cases, or support a link to external tools to accomplish the same
2. MUST provide a mechanism to size the work for a case or link to external tool that does this.
3. SHOULD support the ability to manage and monitor work estimates (e.g. support the ability to estimate time needed, percent complete, work remaining, tabulate values over specific projects or products, etc)
4. SHOULD support the ability to create project plans (rather than using an external tool like MS Project).
5. SHOULD have a mechanism to prioritize cases for a team.
6. SHOULD have a mechanism to prioritize cases for an entire project.

**Code Release Management**

1. MUST have the capability to identify the change dependencies (based on file modifications)
2. SHOULD have the capability to manage the case content of a code release.

**User Management**

1. SHOULD have a way of grouping users by a team (and allow users to be members of more than one team)
2. SHOULD have access control based on a user or team.
3. SHOULD have the ability to link file ownership to a team.

**Performance**

1. MUST perform at least as well as our current tool.

**User Interface**

1. MUST provide a web based interface.
2. MUST provide a command line interface on Windows and Linux.
3. SHOULD provide a plugin to standard IDE environments (like Eclipse)
4. SHOULD provide an interface from smart devices