**Priority Definitions**

When logging an issue, it is important to properly classify the issue according to its priority or its importance.  Priority definitions address the impact of an issue from the developer's perspective and helps to answer the question, "How important is this fix and as such should it escalate to the top of the queue?"  Please review the priority classifications below to determine which one to pick but be aware that this value may be changed by AS.

**P1:** Critical-Impact, fix must be completed immediately.  This rating typically aligns itself with Blocker and/or Critical severity issues.

**P2:** High-Impact, fix should be made quickly.

**P3:** Medium-impact, fix should be made in the normal course of work.

**P4:** Low-impact, fix ought to be made when time permits.

**P5:** Time permitting, fix can wait, it is nice to have.

**Set me**:Awaiting prioritization.

**Severity Definitions**

When logging an issue, it is important to properly classify the issue according to its severity.  Severity definitions address the impact of an issue from the user perspective and helps to answer the question, "What impact is the issue to the user if the issue is NOT fixed."

Please review the severity classifications below to determine which one to pick.  I personally reference this list throughout a project as sometimes an issue might straddle two classifications.

**Blocker:** Blocks development, testing work and/or a release.

* If the tester or developer cannot proceed on to other tests or other tasks and is completely blocked from doing productive work, then a Blocker rating should be used. This classification is sometimes referred to as Showstopper. Typically this severity rating is used the closer one gets to release as the other tasks that can be worked on start to diminish. If the logged issue prevents further work in one area only, please use the Critical rating (see below).

**Critical:** The customer experiences real or perceived data loss or data corruption. An essential part of the system is unusable for customers. No workaround or workaround is clearly unacceptable.

* Application crashes
* Application or service does not work as intended or is unusable; impossible to use the application or a main function of the application; critical functionality lost. Unusable means that users can't or won't use an essential part of the system because of its design. Essential parts of the system are those that users need to use the system effectively, i.e., installation, benchmark tasks used for competitive evaluations, or any part that is required to support the primary tasks for which users have purchased the system.
* Stored data corrupted
* Conflicts with a high-priority (must-have) accepted requirement
* Severe memory leak
* Compromises Stanford's standards or policy (e.g. security)

**Major:** The customer's effectiveness is severely compromised for an essential part of the system, although all essential parts of the system can be used. Workaround is present but difficult or awkward.

* Operation, effectiveness or satisfaction is significantly impacted. Effectiveness refers to the user's productivity and satisfaction with the work process provided by the system. Satisfaction with the work process includes concerns such as unpleasant or frustrating processes that affect the system's fitness for use.
* User is possibly led to creating future problems.
* Conflicts with a medium-priority accepted requirement.
* Help file is missing.
* Help file is incorrect and misleads the user.

**Medium:** The customer's effectiveness is compromised, though not severely. All essential parts of the system can be used. Workaround is present and acceptable.

* Loss of function, effectiveness, but essential parts of the system can be used and/or there is a clear workaround
* Unfriendly behavior that is hindering, but workable, for the user or service.
* Inconsistent implementation across the system that detracts from usability or usefulness.
* Unclear, inaccurate or missing recovery in error messages.
* Spelling or grammatical mistakes in the user interface.

**Minor:** The customer can circumvent the problem and use the system with only slight inconvenience. Permanent workaround is acceptable to the user and to Customer Support.

* Cosmetic problems in the user interface; i.e. font sizes, placement of controls (edit boxes, list boxes, etc.) that are not hindering.
* Inconsistent implementation across the system that does not detract from usability or usefulness.
* Unfriendly behavior that is merely annoying to the user or poses slight inconvenience.
* Explanation of feature missing; spelling or grammatical mistakes in the Help File.
* No loss of functionality.