

Releases



☆

Your Rating: ☆☆☆☆☆ Results: ★★★★★☆☆ 101 rates

Trying to simplify, consolidate and streamline our release process, we'll document some of the "rules" we're trying to apply.

TODO - merge with other release-related pages from [Release](#)

- [Release Candidates](#)
- [Blocker issues](#)

Release Candidates

Release Candidates (RC) are build of the software we do before a final release. Their name implies that they are "candidates" for being a final release. If validated (QA), they become the final release. (technically, we're rebuilding them, but they're essentially a 1:1 copy)

If an RC is not validated, fixes have to be made, upon which another RC is built and released, and so on until we validate.

We do these RCs to have a **stable** basis to do QA. If unnecessary changes are introduced, this means more review and more QA work, which is both workload and risk which we can't afford at that moment.

Blocker issues

While starting the development cycle for a "next" version, there are multiple reasons why an issue, feature or bugfix, would be considered a "blocker"; the most common ones would be along the lines of:

- we need feature X for project Y which has to roll out with Magnolia n.m next month
- we need X to be able to implement Y in this or the next version
- feature Z of module M is planned for the next release, but needs X in Magnolia-core to work ... and so on.

What we put in a releases is guided by customer/user needs... and marketing. We're **trying** to be transparent and expose our plans upfront. (see the [Roadmap](#) pages)

Roadmaps and priorities can change during the course of a cycle, but the closer we get to to a release, the more precise it has to be.

As soon as we're in "RC mode", the only reason code changes happen, or the only way any issue deserves the title of "blocker" is:

- a newly introduced bug that we've only just discovered
- we realise (too late) that a feature/fix we thought was completed actually isn't.

- bundling issues. These are usually fairly hard to spot and fix on snapshots.

There are cases where we've cheated about this. If you're looking at this from an external point of view, it might be surprising. But we **do** discuss and debate each and every change occurring in between RCs, so there is no surprise or unweighted risk.

Contributions are always appreciated, but we have to be fairly strict. If fixes are SO urgent they **really** have to be included in between RCs (we're only humans, we might have missed something), to avoid the waves of panic, report them on Jira - maybe with a reviewable patch - and ping the dev-list; don't be offended if we ask you to wait until the next release - the criteria above will still apply. If you monitor the dev-list, though, there's a good chance you've seen a "release plan" e-mail passing by, and have reacted in time 🤖

Obviously, we'd love to improve this and maybe be more flexible; feel free to suggest improvements to be made !