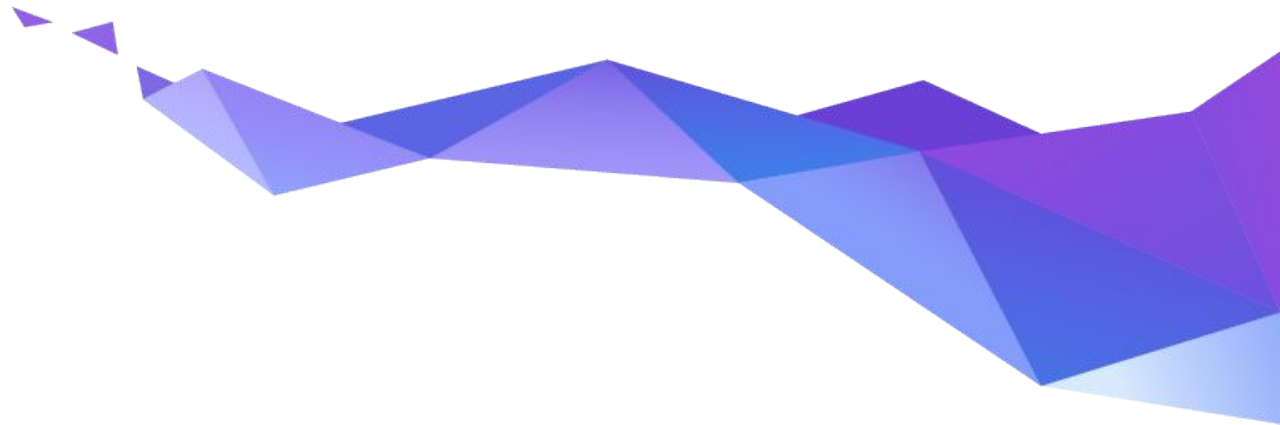


Proposal for a Standard Repeatable Zephyr Schedule

Mark Linkmeyer
12-19-2016



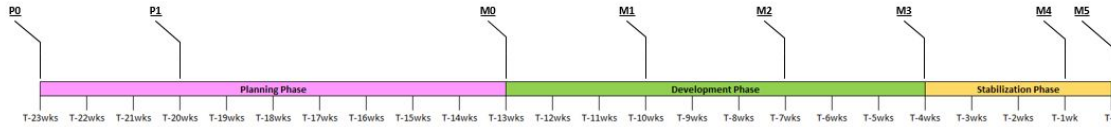
Purpose

- Standardize a repeatable schedule for planning, developing, testing, and releasing Zephyr
- Drive better community involvement by utilizing standard schedules and processes from one release to the next
- Drive consistent high-quality releases of Zephyr through repeatable processes and practices
- Drive clearer understanding of roles/responsibilities

Schedule & Milestones (High-Level View)

See next slides for details and breakout by phases

Zephyr Release X



Zephyr Release X+1



Time

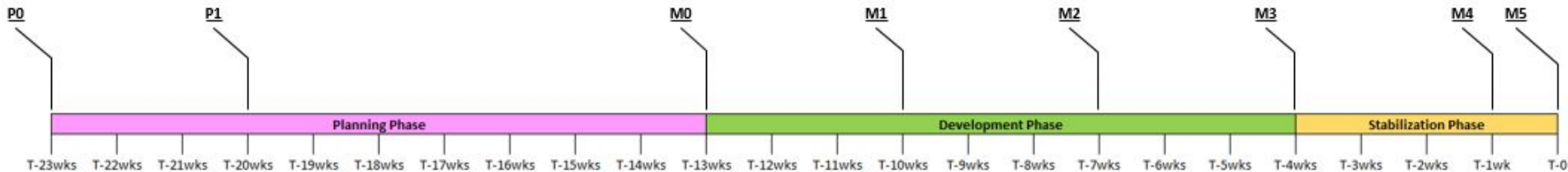


Schedule & Milestones



| | |
|----|--|
| P0 | Reminder Email Sent to Kickoff Planning Start Entering Epics, Stories, Tasks |
| P1 | TSC Agrees on Major Features / Epics and Schedule |
| M0 | All Stories & Tasks Prioritized, Sized, and Assigned All Stories Are Associated with Appropriate Epics All Major Feature Stories Are Labeled Accordingly Merge Window Is Opened |
| M1 | Checkpoint for P1s and P2s |
| M2 | Major Features Ready for Gerrit Reviews Test Plans Reviewed and Approved |

| | |
|----|---|
| M3 | Feature Freeze Feature Development Complete (including Gerrit Reviews and Unit Tests Passing) P1 Stories Implemented Feature Merge Window Is Closed Test Development Complete Technical Documentation Created/Updated and Ready for Review CCB Control Starts |
| M4 | Code Freeze RC3 Tagged and Built |
| M5 | TSC Reviews SWQRC and Approves Release Final RC Tagged Make the Release |



Planning Phase

- While the next slide shows what occurs during the planning phase of release X, planning is performed continuously throughout the entire lifecycle of the release in that issues entered into Jira are constantly being reviewed to determine if the new entries should be included as part of the current release that's in development or whether they should be planned for a future release.
 - It's much like Bug Triage which is constantly assessing priority of bugs and determining in which release each bug should be fixed.
- Epics/Stories entered into Jira and subsequently planned before the M0 milestone have a much higher likelihood of getting into release X than do epics/stories entered after M0.
- The Release Manager owns management of the scope of what will be included in release X, sets the FixVersion field accordingly, and determines if/when a change in scope must be taken the TSC for review/approval.

Planning Phase

P0

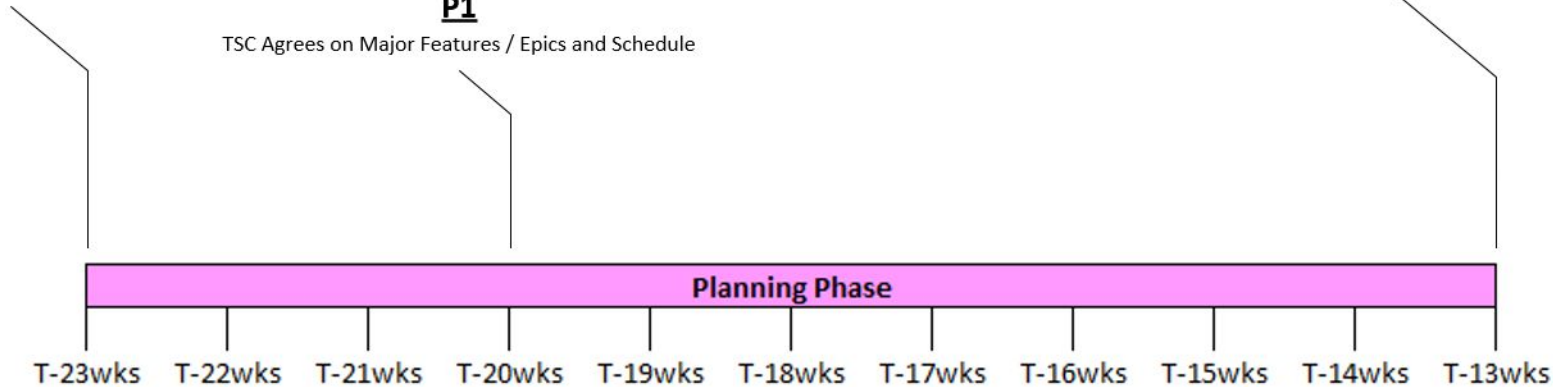
Reminder Email Sent to Kickoff Planning
Start Entering Epics, Stories, Tasks

P1

TSC Agrees on Major Features / Epics and Schedule

M0

All Stories & Tasks Prioritized, Sized, and Assigned
All Stories Are Associated with Appropriate Epics
All Major Feature Stories Are Labeled Accordingly

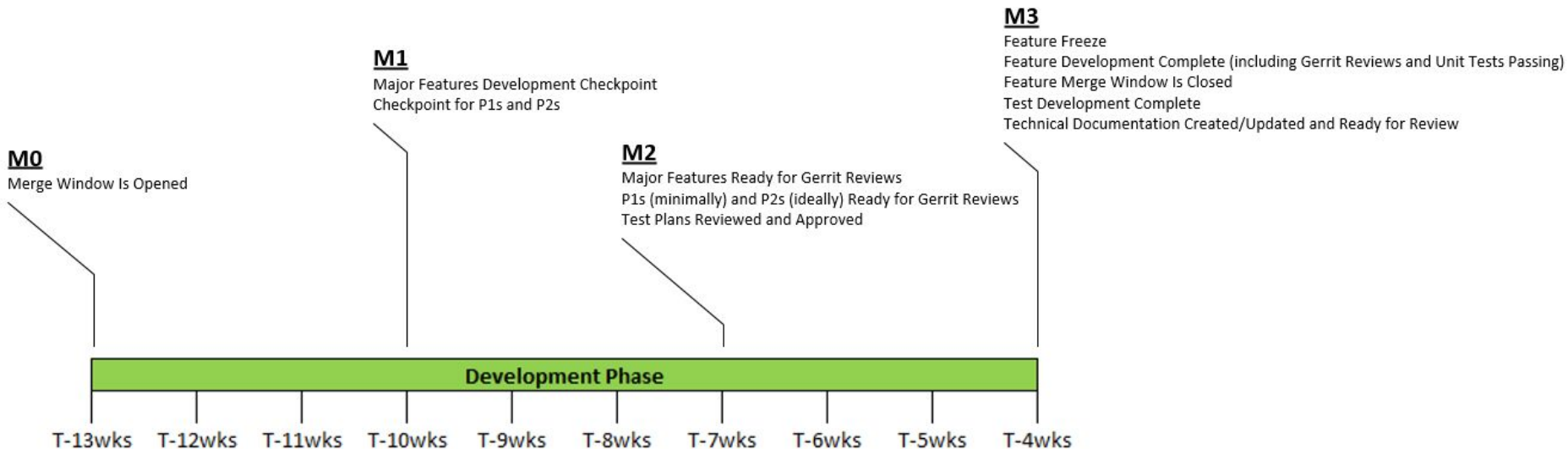


Assumptions



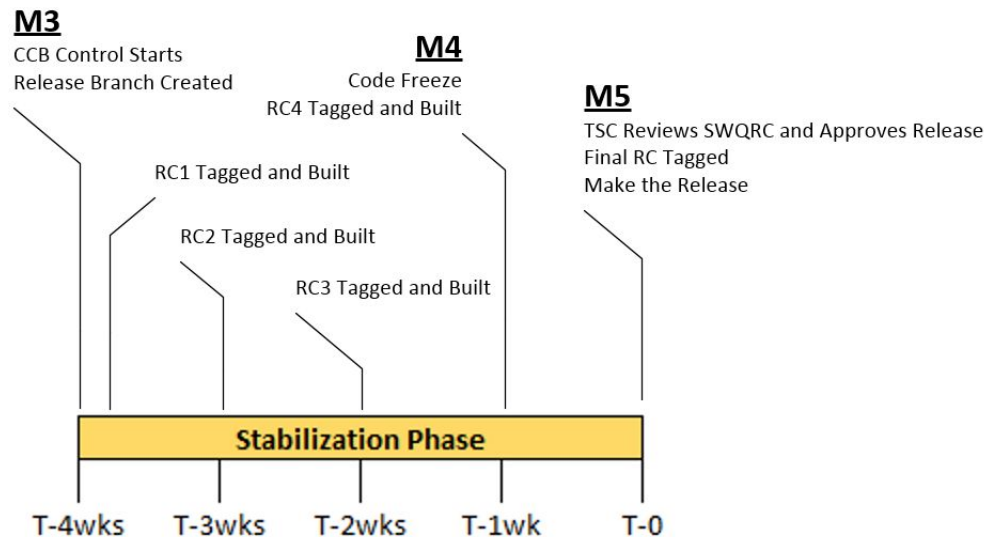
- New Priority definitions:
 - P1 = must-have
 - P2 = should-have
 - P3 = nice-to-have
- Bug Triage occurs on an ongoing basis throughout all phases of the project to ensure bug priorities are properly set
- Coverity and ProtexIP scans are automatically run on a regular basis so that's why there's no mention of running them at any milestone. Results of the scans will be available at least on a weekly basis.
 - Work is currently planned for 1.7 to make this so
 - Worst-case, as a contingency, if this automation work is not completed, the scans should be run manually on a weekly basis from M3 through M5
- CCB (Change Control Board) = TSC

Development Phase



Stabilization Phase (High-Level)

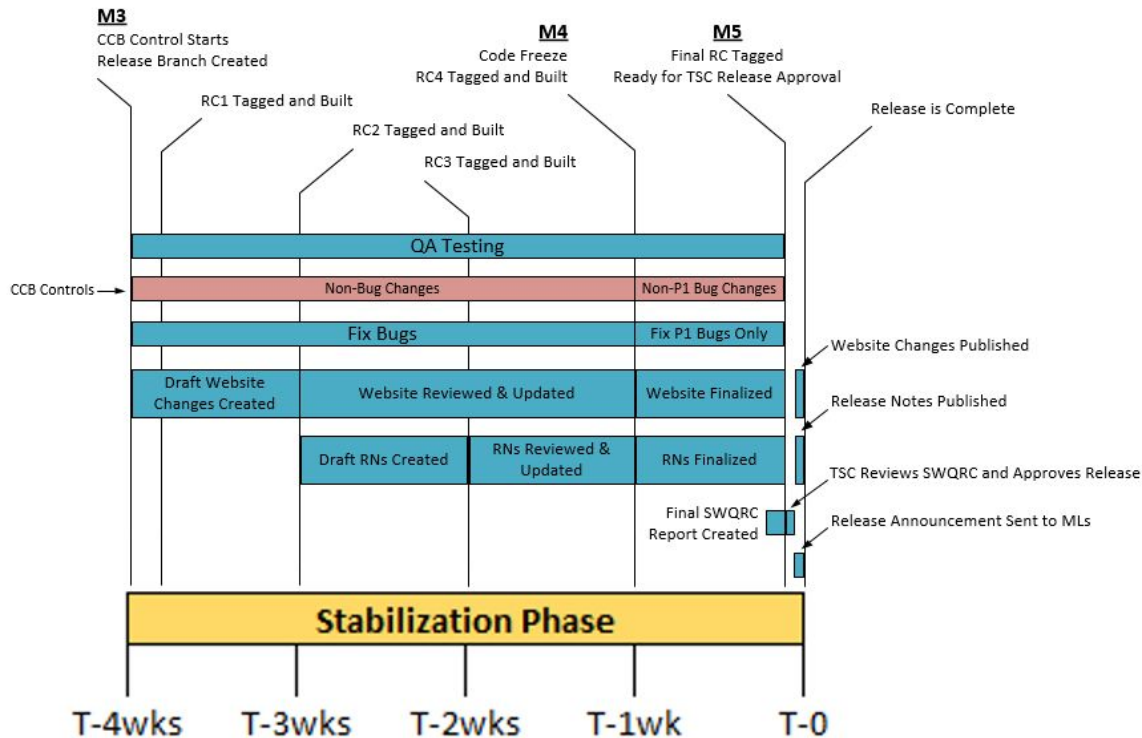
more details on next slide



Notes:

- During the Stabilization Phase:
 - Code changes for release X are made on the Release Branch which is created at M3
 - Code changes for release X+1 are made on Master
- After the release at M5, all code merges are again made in Master until the next release reaches its M3 milestone where a new Release Branch is again created.
- SWQRC = Software Quality Release Criteria
- Generally, RCs are planned to occur once per week at a minimum. More might be needed, but these additional RCs are not shown above.

More Details of Stabilization Phase



Jira Usage Changes Needed to Support Proposal

Reliable and accurate measuring of progress towards meeting Release Criteria drives the need for data to be queryable from Jira.

- Need to be able to query Jira for ‘major features’ versus ‘non-major features’
 - Use issue type and priority field as follows:
 - Major Feature = Epic or Story with P1 priority
 - Non-Major Feature = Epic or Story with non-P1 priority

Notes

- P1s (a.k.a must-haves) are epics/stories/tasks that are so important to the release that if they aren’t all done the release would be delayed in order to get them done.
- Major Feature Epics will certainly have must-have stories and/or tasks linked to them, but they could also have nice-to-have stories and/or tasks which are desirable for the release, but not having them completed would not gate the release.

Release Dates

- Releases are planned to occur by the end of the week in which the last day of the month falls
 - Plan to release on the Wed of the week in which the last of the month falls (wwXX.3). This gives a couple of extra days to get the release out by Friday if any issues are encountered.
 - Closure of the Feature Merge Window is planned to occur four weeks prior to the planned release (wwXX-4.3)

| Release | Planned Feature Merge Window Closure | Planned Release | End of Month |
|---------|--------------------------------------|------------------------|------------------------|
| 1.7 | Feb 1st, ww05.3 (Wed) | Mar 1st, ww09.3 (Wed) | Feb 28th, ww09.2 (Tue) |
| 1.8 | May 3rd, ww18.3 (Wed) | May 31st, ww22.3 (Wed) | May 31st, ww22.3 (Wed) |
| 1.9 | Aug 2nd, ww31.3 (Wed) | Aug 30th, ww35.3 (Wed) | Aug 31st, ww35.4 (Thu) |
| 1.10 | Oct 1st, ww44.3 (Wed) | Nov 29th, ww48.3 (Wed) | Nov 30th, ww48.4 (Thu) |

- See this link for a complete and more visual view of all of the proposed milestones:
<https://drive.google.com/open?id=17bcHbMBUT-oOpIHjI4VuE6K9VwVJD32ZR6nQepY6P4Q>

Questions / Comments to Be Addressed Later

- What additional QA/Testing needs to be highlighted/addressed in the schedule?
- Should the following fields be renamed/added in Jira?
 - “Fix Version/s” changed to “Planned Version”
 - Indicates in which release an issue (epic, story, task, or bug) is planned to be delivered/fixe
 - “Actual Version” added
 - Indicates in which release an issues is actually delivered/fixe
 - *Note: It was mentioned in the Dec 19th Process WG meeting that a field named “Release Version” was used for this today, but MarkL can’t find the field in Jira.*
- “Subsystem Owners” should have a say in what goes into a release and should be able to make decisions on lower priority issues (like P2s and P3s)
 - Subsystem Owners can set Fix Version/s field for P2s and P3s?
 - Only the Release Manager can set the Fix Version/s field for P1s (and the Release Manager owns determining it the TSC needs to review/approve)?
- Are rules (based on issue priority and complexity) needed to determine if/when new issues (including Community Contributions) can be added to a release already past M0?
 - For example, after M2 no more P1s can be added to the release
- Adding a “Complexity” field in Jira (with defined levels for consistent usage) would be helpful in planning when a story or task should be released (i.e. in setting the Fix Version/s field)

Thank you

