Gromacs - Bug #949
Release plan for 5.0
05/30/2012 10:43 PM - Roland Schulz

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Category:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
<tr>
<td>Affected version:</td>
<td>5.0</td>
</tr>
</tbody>
</table>

Affected version - extra info: 5.0

Description
After the 4.6 release we should agree on release plan for 5.0.

I think there are two general options to do that:
- Have a list of features as goal for 5.0 and release when those are done (as we did for 4.6)
- Have a date for a deadline for new features (RC1) (as is done by e.g. Eclipse)

Of course both options would allow exception when decided so by e.g. the majority/special circumstances.

If we would have a yearly deadline for RC1 this would give everyone much more planning certainty and would avoid having the urge of adding many more features while the agreed features are still getting ready.

History
#1 - 05/30/2012 11:34 PM - Erik Lindahl
Hi,
Good initiative!
First a caveat: I'll be a bit more quiet in the discussions, since I need to focus on coding for 4.6 :-) I very much favor a fixed date, but also having everybody take part in the bug-squashing the last few weeks.

I think there are two main problems we have had with feature goals: First, we have frequently had 1 or 2 major features drive a release, and then the people working on those features have obviously done a large part of the release work. If their feature code is late, it's not reasonable to ask they should spend all the time on the release, but reject their own code that was the entire reason for doing it => delays.

Second, we (all) sometimes work on large features in separate trees, and do a lot of work to get this into the release branch (even before any freezes). At that point it always feels a bit tough on people to say "sorry, you were 2 weeks late, so your code has to wait for the next major release in a year". However, the consequence is that we don't do that for any code, and suddenly some code is almost done 2 months later (after a LOT of hard work by the developer), and then it obviously feels even worse to reject it.

There is no blame-game here - I think everybody's actions are very reasonable when isolated, but it's the combination that leads to problems :-) At least in Sweden, january and august (or the entire summer) are usually relatively quiet months, so my personal preference would be to have RC1 for each release on august 1, RC2 on august 15, and the full release by the end of august. I think we could accomplish that with a feature freeze on july 1 if there are no features that are guaranteed to go in!
I think that this discussion would also benefit also from a bit more general discussion on how we do Gromacs development. Since I don't know the reason for the delays (and don't want to start guessing, as that would easily seem as blaming someone), and am also otherwise a bit out of the loop, I will just write my own thoughts here without that much analysis (that also keeps the texts to a manageable length...). Also, I don't know how is the recruitment going on for the project manager position.

In general, I think it would be nice to move towards a bit more "agile" development methods. Jenkins and Gerrit are already a nice step in the correct direction, so below I will focus mostly on managing the features, and not on actual coding or technical solutions for that.

At some point, there was brief discussion of Scrum, but I think that Scrum's short, strictly timeboxed sprints with a fixed goal may not suit Gromacs that well for a few reasons:

- Most people working on Gromacs are doing it voluntarily, when their time allows it, and it's not very predictable how much time they will be able to spend in the next few weeks.
- Geographical distribution, combined with the fact that people are volunteers, also brings challenges to day-to-day communication, which is crucial for getting something done fast.
- With a geographically distributed team and with people having very different interests and skills, I think it is not realistic to assume that everyone would have enthusiasm to focus on developing just one feature at a time.

But with that said, I think several components from Scrum or agile development would be beneficial to have:

- A clearly defined product backlog that everyone knows, and that is actively kept prioritized and up to date. And features being actively worked on are split into manageable tasks such that progress is visible for everyone (and volunteers can help on tasks that no one else is working on at the moment).
- To make the above work, there should be a clear “product owner” (just one, or a group of people), who has the mandate to do the prioritisation, and has enough time to actually do that.
- Develop features in small pieces and integrate often, keeping the main branch always “release quality”. It is unavoidable that some features will have an unfinished look or not be fully functional all the time when doing it like this, but as long as they don’t break existing functionality, it should still be possible to do a release even if something is not completely finished.

With the above components, I think it would be much easier (and more predictable) to do release planning. Both a scope box (fixed set of features), or a time box (a cutoff date) are possible, as is a combination (tagging some features as must-have, which should then be on top of the priority list). And the list could evolve all the time when priorities change or additional work comes up, keeping everyone on the map.

As a bit separate issue (but related to at least maintaining the product backlog), I think we really should also discuss how to best take use of Redmine. At least my idea of pushing the move away from Bugzilla was that some other tool would be much more flexible for managing also other tasks than just bugs. And one idea would be to maintain the product backlog in Redmine, so that work really is visible to people. Currently, there are several Redmine issues under the Support Platforms, suggesting improvements to Redmine, but hardly any discussion has taken place in the past two years...
#3 - 05/13/2014 09:37 AM - Mark Abraham

- Status changed from New to Closed
- Target version deleted (5.0)
- Affected version set to 5.0