

Optimizing Feature Management in Autoware: A Proposal for Deprecation and Deletion Process #3906

yukkysaito started this conversation in **General**



yukkysaito on Oct 16, 2023

Maintainer

edited ▾

Currently, Autoware has accumulated numerous features, including some that are not being used and libraries that are not actively employed. However, the current approach makes it easy to add features but challenging to remove them. Autoware is designed to accommodate various use cases by providing a wide range of features and allowing users to launch the functionalities they need. Consequently, just because a feature is not used in the default autoware_launch does not necessarily mean it's unnecessary. Therefore, I believe some process is required to remove features.

I propose the following approach; what do you think?

1. Every 3 months or semi-annually, create a list of candidate packages for deprecation and conduct a survey in discussions to determine whether they can be deleted.
2. If there are no significant comments, proceed with the deletion.

If issues arise with this method, it might be worth considering notifying users that a feature is deprecated and providing a transition period by printing a message at node startup.



2



6



3

Category



General

Labels

type:guidelines

4 participants



3 comments · 5 replies

Oldest

Newest

Top



yukkysaito on Oct 16, 2023

Maintainer

Author

[@mitsudome-r](#) [@xmfcx](#) Do you have an opinion?



1

4 replies



mitsudome-r on Oct 17, 2023

Maintainer

I agree that we can consider workflow of removing packages. Otherwise, packages would infinitely increase in the Autoware repository.

About the frequency of creating the list of candidate packages, maybe 3 months is too much. I was thinking something like once a year.



yukkysaito on Oct 17, 2023 Maintainer Author

@mitsudome-r How about making a judgment on the frequency as we observe the situation? For now, would it be okay to go with a six-month interval?



xmfcx on Oct 17, 2023 Maintainer

We don't need to impose a strict schedule on ourselves for such an effort just yet.

We can do it once and see its impact.

Then we can judge the effort that it took and decide when the next cycle can be done accordingly. What do you think?



yukkysaito on Oct 18, 2023 Maintainer Author

@xmfcx Thank you for your comment 🙌

I agree with you. Let's try it and then decide.



xmfcx on Oct 17, 2023 Maintainer

Dear @yukkysaito -san, your insightful post on the challenges surrounding Autoware's feature set underscores a deeper concern: our limited visibility into who our users are and how they leverage Autoware Universe. Our prime goal is to cater to our users' needs, and it becomes challenging without an understanding of our user base.

Recognizing this blind spot, here are some non-invasive, user-respecting approaches we might consider:

1. **Leverage Polls:** Let's initiate regular polls to directly gauge user behavior and preferences, such as which Autoware Universe features are most commonly used or which updates have been well-received.
2. **Active Engagement in Q&A:** Addressing user queries will serve a dual purpose. It assists our community while also shedding light on common challenges, areas of confusion, or features that might need clearer documentation.
 - Maybe we should analyze the existing questions to figure out the areas we should concentrate our efforts on.
3. **Company Outreach:** With due respect for privacy and community guidelines, we can review the GitHub discussions and create a list of companies that interact frequently or show a keen interest in Autoware Universe.
 - Maybe we can extend an invitation for a focused feedback session or even a casual chat. This direct engagement can offer deeper insights and strengthen our relationship with key users.



xmfcx on Oct 17, 2023 Maintainer

Building on our previous discussions, I'd like to propose an actionable first step towards optimizing Autoware Universe. The idea is to strike a balance between offering comprehensive functionalities and maintaining a clean, efficient architecture.

Initiating with a Package Review:

Before we delve into gathering feedback from our community, we need a starting point—a list of potential areas for optimization. Our internal review can give us a preliminary list.

1. **Compile a Preliminary List:** By diving deep into our codebase, we can identify packages that, upon initial inspection, seem redundant or underutilized. This won't be a final list but rather a starting point for community feedback.
2. **Conduct the Poll:**
 - **The Questionnaire:** Frame the poll to offer clarity. For each package listed, we can provide a brief description and its intended use case. This ensures respondents have the context needed to provide informed feedback.
 - **Seek Specific Feedback:** Instead of a simple 'Keep' or 'Remove' option, perhaps we can have ratings. For example:
 - Essential for my use case.
 - Occasionally use it.
 - Never used it.
 - Didn't know it existed.
 - This granular feedback can provide insights beyond just the utility, revealing areas where we might need better documentation or user onboarding.
3. **Feedback Analysis & Implementation:**
 - Once the poll concludes, the feedback should guide our next steps. Instead of direct removal, perhaps packages that aren't widely used can be made optional or shifted to a separate module, ensuring they're available for those niche use cases without burdening the main codebase.



1



kminoda on Jan 31 Collaborator

FYI: I have created a proposal for deleting several files in perception directory.

<https://github.com/orgs/autowarefoundation/discussions/4136>