

Organizational Matters

MAAV

Contents

1	Communication	2
1.1	Redmine	2
1.2	git	3
1.3	Subversion	3
2	Reimbursement	3
3	Subteam Meetings	4
4	Bugs, Refactoring, Features	5
5	General Outreach Checklist	5
6	Long Outreach Agenda	5

1 Communication

Michigan Autonomous Aerial Vehicles (MAAV) uses multiple tools for communication. In no specific order, they are:

- Email
- Slack
- Redmine
- git
- Subversion

I know what you're thinking; Redmine, git, and Subversion aren't communication tools, they're bug tracking and version control systems. Yes, but they are also important communication mediums. Teamwork requires two things: work, and a *team*. Work is rather straightforward, but the team part requires some effort as well. Being on a team requires putting effort into communication.

1.1 Redmine

A centralized bug tracking system is necessary to work together as a team[3]. Redmine should be the main medium for discussing bugs of any nature, and a supporting medium for discussing features. Most discussion about features is far outside the scope of a bug tracker¹, but Redmine is still useful for tracking the progress of the implementation and general summarizing of direction the feature is taking. Bugs and features as put into Redmine need to be specific, not generic[4], since we also use Redmine for scheduling. The general workflow on Redmine is:

1. Create a bug/feature
2. Assign to a team member
3. Set the due date
4. Toil away, commenting on the bug/feature to keep others updated
5. When ready for review, mark "Resolved" so a manager can review

This workflow should be combined with the workflow in the following git or Subversion sections depending on which version control system (VCS) you are using.

¹ Whiteboards, pizza, passionate discussion, etc. are usually needed.

1.2 git

git relates to communication in that your workflow and commit messages communicate what you're doing. A team that uses a complicated branching model like git-flow[2], just by using that model, communicates that they release infrequently and releasing is a big deal. In the same way, a team that uses GitHub flow[1] communicates that they release all the time, and that releases are no big deal. We use GitHub flow, so the typical workflow is something like this:

1. Create a feature branch off of master
2. Work on feature branch, committing and pushing to your personal remote
3. When ready to merge, mark resolved on Redmine
4. Maintainer reviews changes, giving feedback and/or merging to master

1.3 Subversion

Subversion relates to communication in that your commit messages and the locking/unlocking of files lets everyone know what you're doing. Subversion is simpler but less powerful than git, and MAAV only uses Subversion for repositories that have non-human-mergeable files². The typical Subversion workflow is:

1. Work on some stuff
2. Commit each logical change
3. When ready for review, mark resolved on Redmine
4. Maintainer reviews, gives feedback, marks closed if done

2 Reimbursement

MAAV will gladly reimburse you when you buy stuff for the team. Please note that to be sure you will be reimbursed, you need to check with a lead before you buy something.

To get reimbursed, email the receipts to maav-leads@umich.edu. The receipts must have proof of purchase; in other words, the receipts need to either demonstrate that you paid in cash or with a credit card. If you paid in cash, no further action is necessary. However, if you paid with a credit card, your credit card number *must* be on the receipt. If it is not, you must also attach a credit card statement to the email. Below is an email template for the reimbursement email to send to the leads:

² mechanical/electrical computer aided design (CAD) models, MS office documents, etc.

I spent <amount> on <subject of purchasing>. Please set the check to be <mailed | picked up>.

<your name>
<your address>
<your city>
<your state>
<your zip>

Thanks,
<your name>

It may look like the below (if you would like the check to be mailed):

I spent \$420.69 on plastic and aluminum. Please set the check to be mailed.

Sasawat Prankprakma
666 Broadview Lane
Ann Arbor
MI
48105

Thanks,
Sasawat

3 Subteam Meetings

Subteam meetings are for organization; at the beginning, each member talks for a bit, following this template:

1. What is your deliverable for this week?
2. Regarding that deliverable, what have you accomplished and how did you accomplish it?
3. What are you stuck on?
4. What will you work on for the next week?
5. What is the high-level design you are thinking of for next week's deliverable?

4 Bugs, Refactoring, Features

There are three main categories of engineering work on MAAV:

- Bug: An error in correctness
- Refactoring: Simplifying/improving an already correct design
- Feature: Adding to the current body of work

MAAV (hopefully) follows the “zero defects” or “zero bugs” methodology. Bugs should be fixed first, refactoring after there are no bugs, and new features should be implemented last.

Fixing a bug soon after it is created is mostly a non-issue; fixing a bug a month down the road likely means that other components in the system depend on the buggy behavior. The argument for prioritizing refactoring follows in the same vein.

This methodology applies to every engineering source material: CATIA models, FEA simulations, CFD simulations, electrical schematics, electrical board layouts, and code.

5 General Outreach Checklist

For any outreach that involves standing at a booth and doing demonstrations, follow this checklist:

- Vehicle
- Computer with working competition simulation
- Sample PCBs
- Sample airframe parts

6 Long Outreach Agenda

For outreach events where a group is in the Wilson Center for at least 20 minutes, the following schedule should be approximately followed. Remove or add demos to the schedule as necessary.

- Explain the competition, show a video of the simulation.
- Describe the team. Talk about the subteams and what each subteam does. Use props/spare parts as necessary, especially for structures and circuits.
- Demonstrate the motor test bench. Describe why we test and characterize motors.

- Demonstrate the carbon fiber layup process. If spare parts are needed, use real epoxy and make real parts. If the parts are just for demonstration purposes, use 5 minute epoxy. Narrate the layup process, describing what the carbon fiber and epoxy each do.
- Demonstrate Printed Circuit Board (PCB) population. Pick a spare board and solder on a component or two. Narrate the process.

References

- [1] Scott Chacon. Github flow, 2011. <http://scottchacon.com/2011/08/31/github-flow.html>.
- [2] Vincent Driessen. A successful git branching model, 2010. <http://nvie.com/posts/a-successful-git-branching-model/>.
- [3] Joel Spolsky. The joel test: 12 steps to better code, 2000. <http://www.joelonsoftware.com/articles/fog0000000043.html>.
- [4] Joel Spolsky. Evidence based scheduling, 2007. <http://www.joelonsoftware.com/items/2007/10/26.html>.