

Report on Stage 3

Written 2019 02 17

While our team seems to be becoming far more proficient at accomplishing and planning our workflow as time progresses, our turnaround time on new assignments is still far too slow, as a goal for our team during the next stage I plan to enforce a deadline constrained to a week before the actual submission time, to help our productivity and performance in the earlier stages of our assignments and cause far less of a concentrated workload near the deadline.

Observations of the coding seem to be that currently, even though all our team has attended the git seminars, we've either forgotten what we've learned or never gained proficiency, within Stage 4 I would like all members of our team to be comfortable with using git and not be dependent on others when it comes to commits and changes.

While we did briefly discuss the specificities of our program flow regarding the authenticate use case, we haven't really discussed the overall structure and flow of program in general. This short-sighted design choice will almost certainly lead to a lot of the currently written code to be restructured or rewritten entirely. This is not a large deficit currently due to only one, rather simple, use case being modelled; but within Stage 4 we need to take a far more structured approach to avoid any design choices that will add un-necessary tech debt.

Much of our documentation is well suited for record keeping, member conduct, transparency of operations; but we are currently rather lacking in proper coding standards, member experience within a development environment and the functional development of the program. Stage 4 should include the ironing out and further development of our coding regulations, reviews and documentation.

An issue has also arisen over time where our file store for holding documentation has become increasingly disorganized and cluttered, this has become especially clear during our migration from Google docs to the GitLab platform. Some housekeeping and stricter standards may need to be put in place to avoid this in future, or simply having a temporary file store which is not version controlled for less formal file transfer.

Something else to consider is holding all of our team's resources in a single place, currently our GANTT chart is hosted externally but we have been made aware that GitLab has support for GANTT creation and management, in a future stage I believe having the GANTT accessible without navigating to an external site will make it far more likely for it both to be maintained and adhered to.

Finally, beginning this stage I will be maintaining "Captain's Logs", reports written at the end of a Stage as a retrospective of the performance of a given stage, with notes on what needs accomplishing in future stages including potential improvements to our workflow.

This is captain Marin signing off.

Report on Stage 4

Written 2019 03 03

Group performance for this stage is up to the standard of previous stages, but the lack of my ability to come up with proper planning documentation for our methods and classes causes us to come up with more ad hoc solutions given specific implementations.

This is a problem both because it violates our TTD lifecycle coming up with unstructured solutions, but also because of the continual structuring then restructuring of code due to not having proper guidelines for general structure of the program.

While our meetings and sessions assist a lot in specific implementations, and diagrams drawn during them informs our design decisions, we're limited by our pre-planning in this stage and have come up with naïve solutions to problems as they've arisen.

Another limitation of this stage was the assignment of work, we have currently a documentation team and a coding team, while this would be fine in a planned workspace this isn't what the team members want or our initial goal for the group work. We want everyone to understand and contribute to the codebase and the members currently doing documentation feel this way too, within stage 5 a stronger focus should be held on working on the code as a team as well as planning it out and documenting it better.

Our filesystem while much improved still has junk files in it, mislabeled files, files without a proper place of storage, etc. We still need minor revisions to our file system to make stage 5 easier but to also to make it, so we spend less time micromanaging minor issues from disorganization.

For the next stage we either need to outline our work far more explicitly so our team all knows where we're going and how we're getting there, preventing the creation of more hacky code.

During this stage we also did a trial run of a new system where work is reported 6pm before the day is assigned to heighten accountability, but many people didn't adhere to this improvised rule, we may attempt it again at the beginning of the next stage, but if it turns out to be bust then we'll be forced to settle for our current methods of accountability until a better solution is found.

To supplement the effort above we've also made team review documents, where team members will review other members based off their performance and any particularly low review will be discussed in the first meeting returned.

There isn't much more to talk about in this stage, group performance was good, but organization is simply taking up too much time to be practical. More of the group should also be assigned coding roles throughout the project.

This is captain Marin signing off.

Report on Stage 5

Written 2019.03.25

This stage we had many issues, with organization, program planning, time management. I believe this to be primarily my fault, not setting up a strict schedule for my team to follow, not having a solid vision or idea of program structure to press onward and support my team with, not understanding design patterns well enough to build a functional and cohesive system.

While our documentation could be maintained and was somewhat well preplanned and adjusted as our project progressed the bottleneck from first creating the documentation and then maintaining it and updating it drained morale from our team very quickly and slowed our progress in later stages. Whether it would be more beneficial to have done things the way we did or have a simple set of core principles and adhered to them as we saw necessary is hard to determine. Maybe enthusiasm waned due to the length of the project rather than simply the fatigue caused from the effort espoused in the early stages.

Either way the strong obstacle of getting the documentation complete feels like it was monumental enough to prevent our team from ever getting any good rhythm.

Vision is also something very important for a project, feeling as if you had a goal to get to and knowing how to get there helps productivity and engagement immensely. Much of our planning for the future focused too much of the divergent paths of our progress rather than where it was headed, this led to us spinning our wheels in place a lot and never getting anywhere, despite there actually being many hours of work and sweat involved during our meetings and sessions.

Me and James were confident enough to code any system or subsystem of the program but without the foresight to properly determine scope our program's requirements vary wildly based off how we wish to view it, this also causes a rift with the rest of our team, not understanding what is expected to be done or what has been done so far and why limits their ability to contribute productively to the codebase. "What if I define this method wrong relative to the system", "I want to work on this feature but I don't know how this subsystem works yet", etc.

We had planned on running a TDD cycle to mitigate some of these issues but couldn't properly curate plans for the system, smart planning should be emphasized in the future rather than simply more of it, prioritization of what needs to be done and how and where rather than simply all the ways something can be done without any significance attached to what we want.

Also from our documents fatigue certain naïve policies slightly fell out of date, for example the coding style guide and it's rules regarding null variables and typecasting, during our programming experience we have actually found both useful and rather necessary at times and should have amended that document, but from all of our bureaucratic work in the past changing this seemed like too much of an inconvenience. Although to be fair a vast majority of our policies were in fact maintained to the letter so rather well thought out overall.

This is captain Marin, signing off one last time.