**Action Items**

1. Create a style guide document with basic formatting rules and naming conventions
   1. This will become part of your codebase documentation
   2. The style guide can (and should) evolve as you work through the project (just be sure to go through the code once more at the end to make sure all code is formatted to your guidelines)
2. Agree on a basic process for writing code and conducting code review
   1. Will commits happen before any review occurs or always after some review?
   2. How large will commits be? (e.g. Is each commit a task? Commits of 100 lines or less)
   3. Who will conduct the review for each file? Will there be a set rotation within your team?
   4. Who will make edits and updates, the initial coder or the reviewer?
3. Designate roles for larger and/or recurring tasks
   1. Who is responsible for merges?
   2. Who is primarily responsible for external documentation of the codebase?
4. Write code
   1. Keep your style guide in mind
   2. Keep the code review checklist in mind
   3. Check for bugs/defects before submitting for review
      1. There should not be any warning or error messages
   4. All code submitted for review should have a purpose
5. Conduct code review
   1. Utilize the checklist and make comments
   2. This should be an independent review (done by someone who wasn't there when code was written)
   3. Should not necessarily be a “defect finding activity”. Should be used to:
      1. Improve codebase
      2. Provide constructive feedback to improve individual skill sets
      3. Improve understanding for the client
6. Make edits/updates
   1. Changes should be small
7. Review by whole team (the goal is that this is minimal after individual reviews)
8. Merge
9. Documentation review
   1. Update README file
   2. Include a vignette/codebook
10. Review for client
    1. What additional documents/information do the clients need in order to use code and its outputs?
    2. Think about usability of codebase