Software Development Process (SDP)

[Principles](#_2s13ohyeqgb7)

[Process](#_dv87m3vones8)

[Roles](#_r4ojepvcw4mh)

[Tooling](#_gtpscwd44yls)

[Definition of Done (DoD)](#_gtqo52h8bto2)

[Release Cycle](#_3elqxufgawry)

[Environments](#_nqxqf74b5aov)

# **Principles**

*The goal of our software development process is to maximize the amount of our work that ends up being useful to the end user. This entails rooting all of our activities in the benefits they will provide to the end user, ensuring we can efficiently coordinate the activities of different developers, and maintaining a standard of code quality both in releases and during development.*

* *We answer communications within 24 hours on weekdays through Discord. We have a Discord for our individual team and one shared between all the groups working on the project.*
* *We’ll use the GitHub issue tracker to keep track of what each developer is currently working on.*
* *Each GitHub issue will have a corresponding issue branch where work to complete that issue will take place.*
* *When an issue is complete, a pull request will be made into main.*
* *Pull requests can be merged only after approval from at least one other developer who is familiar with the issue subject.*
* *No breaking changes will be merged into main - “breaking” meaning any changes that could result in errors or misbehavior that other developers may need to fix before they can start work on an unrelated issue.*
* *We will organize issues into weekly sprints with issues assigned at the weekly group meeting, and with the intention that they be completed by the next weekly group meeting.*
* *If any unexpected or pressing problems arise on the development side (e.g. a model cannot be parallelized as well as expected, or a release breaks when run in a new environment), we will bring it up with the project partner in a meeting in order to develop options to proceed.*

# **Process**

* *Project Partner Meeting (1/week)*
  + *Each week we demo progress to the project partner and receive feedback which will be used to inform new creation of issues for new features or bug fixes at the next sprint review meeting.*
* *Team Meeting / Sprint Review & Planning (1/week)*
  + *Weekly team meeting led by the project manager where we review the team Kanban board (Todo, In-Progress, Review, Done) on GitHub.*
  + *Issues that are Done will be closed and archived by the project manager.*
  + *Any issues stuck in review will be assigned a reviewer by the project manager.*
  + *Issues that are still in-progress will either be edited to shrink the scope, or removed and replaced with a smaller issue. This is so no issue will drag on for longer than a week.*
  + *New issues will be created based on feedback from the Project Partner meeting and added to the Todo column.*
  + *The project manager will assign each developer at least one issue from the Todo column each week.*
  + *Each week an issue for updating the process documentation is assigned to one of our developers by the project manager.*
* *TA Meeting*
  + *Weekly check in to review progress and ensure that we are meeting all of our obligations, which is used to inform issue creation and allocation at the sprint review meeting.*

# **Roles**

***Project Management & Infrastructure & Testing - George Connor Hutchinson***

*Responsible for group meeting management and the issue backlog. Responsible for creating and maintaining the CI/CD system on GitHub.*

***Backend Simulation Engineering - Dylan Oldham***

*Responsible for studying models of radiation damage and creating a simulation of the evolution of microstructural defects in materials which are exposed to radiation, and ensuring the simulation can be effectively parallelized in a way that meets the needs of the project partner.*

***Simulation Parallelization Engineering - Ella Riis***

*Responsible for parallelizing the simulation using CUDA. Takes simulations as programmed by the backend simulation engineer and creates an optimized version using the CUDA framework.*

***Design & Frontend UI Engineering - Benjamin Reed***

*Responsible for creating the simulation UI according to the needs of the project partner. Responsible for creating a system for automatic visualization of simulation results.*

# **Tooling**

| **Version Control** | GitHub | GitHub will allow us to maintain current versions of our projects as we work, as well as implement new features as the project evolves. |
| --- | --- | --- |
| **Project Management** | GitHub Issues and Projects | GitHub issues and Projects allow us to keep track of what are the pressing features we need to implement. It also keeps us on track as to who is working on what. Issues allow for us to provide feedback to one another and push changes the whole team agrees with. |
| **Documentation** | GitHub Wiki | GitHub Wiki is a convenient way for us to provide documentation to the user and will be linked with our repository for easy access. |
| **CI/CD** | GitHub Actions | With GitHub Actions we can ensure a given build meets our requirements and passes the tests we define for it. It will also allow us to manage Github artifacts so that our project partner can execute working versions of our code without leaving the Github website. |
| **IDE** | Visual Studio Code | Visual Studio Code is a convenient way to edit code as well as provide built-in Github functionality, which will help speed up the development process. |

# **Definition of Done (DoD)**

* *Acceptance criteria are validated*
* *CI/CD workflows are updated to work with any changes to the code build process.*
* *Documentation is updated, functional changes are documented in the README for the corresponding module.*
* *CI tests are successful*
* *Breaking changes are evaluated/avoided*
* *Changes are merged (to main branch)*
* *Changes are deployed as an executable artifact on Github*

# **Release Cycle**

*Our project’s main branch will be automatically built into a staging executable and deployed as a downloadable executable artifact on Github using CI/CD. Any formal releases will be tagged on Github and built using the same mechanism.*

*Our project partner has defined a* ***Version 0*** *of our simulation which will meet the following objectives:*

* *Simulates cluster dynamics using one of the models presented in the papers he has provided.*
* *Utilizes GPU acceleration to speed up calculation using CUDA or OpenCL.*

*Due to the exploratory nature of the project, it is not appropriate yet to say exactly what kind of features would warrant a major or minor version change. At present our only planned formal release is Version 0 in the first week of December.*

*After Version 0 is finished we may consider setting up a more regular formal release schedule.*

# **Environments**

| **Environment** | **Infrastructure** | **Deployment** | **What is it for?** |
| --- | --- | --- | --- |
| Production | GitHub Artifacts | Release | For access by the project partner |
| Staging (Test) | GitHub Artifacts | PR | For our own DoD, and ensuring that it compiles and executes with our automated tests |
| Dev | Local (Linux and Windows) | Commit | Development and manual testing |