Timeline :

- **July 29 - July 31**
  - Work on new PRs:
    - Pick a shogun class and show that it works with tags-parameter.
    - Modify SG_ADD to use tags-parameter.
  - Get [Plugins PR](#) merged.
- **August 1 - August 7**
  - Get [Tags swig interfaces](#) merged.
  - Get SG_ADD PR merged on feature/tags branch.
  - Get old cookbooks merged:
    - [Euclidean distance](#)
    - [MKL Multiclass](#)
    - [MKL Binary classification](#)
    - [Meta example api tests](#)
- **August 8 - August 14**
  - Write a readme guide for Tags-parameter framework.
  - Write a readme guide for plugins architecture.
  - Start peer review.
- **August 15 - August 24**
  - Complete peer review.
  - Work on feedback from peer review.

Old Timeline :

- ~~**May 23 - June 28: Students work on their project**~~
  - ~~**3 weeks**~~
    - ~~Separate branch for tags/parameters~~
    - ~~Make shogun support tags~~
    - ~~Branch for plugins from tags/parameters branch~~
    - ~~Tests for tags/parameters~~
    - ~~Merge tags/parameters branch in shogun develop in the end for serialization project~~
  - ~~**2 weeks**~~
    - ~~Ensure plugins are complete.~~
    - ~~Tests for plugins~~
    - ~~Choose classes for base shogun and start transitioning.~~

- ~~**June 21 - June 28: Midterm evaluation**~~
  - ~~Shogun supports tags/parameters~~
  - ~~Plugin arch is complete~~

- Class Transition for base shogun is almost done.

- **June 28 - August 16: Students continue coding**
  - **1st week**
    - Merge tags branch into shogun feature branch
    - Start the PR for tags SWIG interface
  - **2nd week**
    - Start working on plugins.
    - Try to get Tags swig PR merged by the end of this week.
    - Update meta example generator to work with tags
  - **6 weeks**
    - Complete class transition for base shogun
    - Make plugins for descendents for base classes from old shogun framework
    - One plugin also includes unit tests, examples(meta), notebooks

- **August 16 - August 30: Code submissions, Evaluation and results**
  - Here we submit most of the code with the new plugin interface with proper unit tests, integration tests, documentation, notebooks and tutorials.