MUSICA Software Development Plan v0.1

MUSICA Lead Scientist: Mary Barth
MUSICA Lead Software Engineer: Matt Dawson

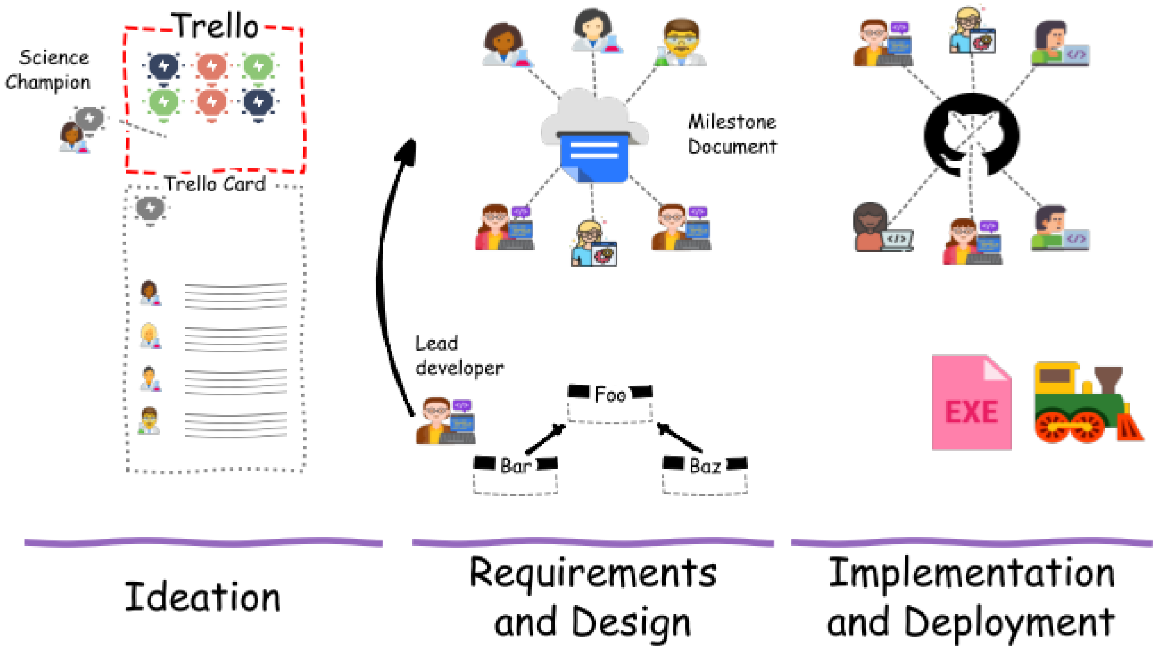

Figure 1[1] An overview of the steps for adding new functionality to MUSICA.

Steps for Development

1. Write down idea on Trello board and discuss idea among scientists
2. Identify Science Champion
3. Create Milestone in Milestone Google document listing requirements
4. Iterate with Design Lead to clarify requirements
5. Create Design and Implementation Plan
6. Schedule Milestone development
7. Execute Implementation Plan in Sprints
8. Perform Retrospective to evaluate product and process
9. Deploy product

[1] Icon attribution

The MUSICA project is intended to support the scientific community in chemical weather and climate models. The use of this document is intended for those who are interested in participating in the advancement of MUSICA software. The general purpose of this document is to provide guidance on the process of developing components for MUSICA.

This *living document* describes the modified Agile process, outlined below, recommended for teams to collaboratively work on the MUSICA project and all of its software components[2]. This document contains recommendations for processes that teams should follow. The goal is to focus software development on user-centric activities and allow both software and science needs to evolve organically, but with intention and purposeful design.

The MUSICA team is welcome to suggest changes to the software development plan, especially if issues arise when following the plan. The MUSICA Lead Scientist and Software Engineer (SE) are responsible for maintaining this plan and can be contacted with any suggested changes.

Progression of MUSICA Development

Here we give a description of the progression of MUSICA Software Development. In the Appendix, a tangible example of the process is given using the implementation of TUV-x in CESM. There is a Glossary of Terms at the end of this document defining the terminology used below.

Requirements

A new science need or model feature

Once a new science need or model feature is identified, the first ‘movement’ of MUSICA Software Development involves documenting the need on the MUSICA Trello Board.[3] It is expected that when initially proposed, the need may not yet be well-defined or fully thought through. *We want to begin documentation at this nebulous first state*. The structure of this document is expected to evolve, and it should primarily be thought of as a scratch-pad for scientists to jot down and discuss new ideas. Cards that are associated with developments that have been completed over the last quarter will be archived to keep the board up-to-date.

To start a new conversation about a science need, create a card on the board in the column most related to the science need (it can always be moved to other columns in the future if needed). There is only one requirement for new additions to the Trello board: the main description should complete the sentence “**As a user of MUSICA, I need to…**”. This helps keep

[2] NCAR/OIS is in the process of developing a Secure Software Development Framework (SSDF) based on NIST standards. The process described in this document will be updated once the SSDF is available.
[3] If you need access to the Trello Board or other resources email musica-support@ucar.edu

the requirements user-focused and avoids straying into implementation details. The final “to” is particularly important in this regard. For example,

> ~~“As a user of MUSICA, I need a database of chemical mechanisms”~~

is implementation-focused and does not actually describe the user need, whereas:

> “As a user of MUSICA, I need ***to*** be able to use chemical mechanisms from any of the most common families of mechanisms for my simulations”

is user-focused and avoids prescribing an implementation.

The conversation among scientists about the new science need can continue in comments on the card and refinement of the description. When one of the scientists is willing to commit to the role of Science Champion, the need can become a Draft Milestone.

Draft Milestone

When a Science Champion has been identified for a new science need, it can become a Draft Milestone, beginning the second movement of MUSICA Software Development. This occurs by the Science Champion creating a new Milestone subsection in one of the Milestone Documents. Milestone Documents are associated with specific MUSICA software packages (e.g., MICM or TUV-x). If the need is not directly related to an existing MUSICA science package, or it is related to several of them, the Science Champion can work with the SEs to determine where to include the Milestone.

Once a Draft Milestone has been created, the Science Champion should contact the SEs to identify a Design Lead. The Design Lead is responsible for communicating uncertainties about the Draft Milestone to the Science Champion through comments on the Milestone Document. *It is important that all clarifying information about a Draft Milestone be documented in the Milestone Document so the whole team has visibility of the science needs.* Note that the Science Champion acts as a point-of-contact for the Design Lead, and will likely bring in other scientists to address questions about the requirements.

Once the Design Lead understands the request sufficiently to begin the design phase, the Draft Milestone moves out of draft status and becomes a Milestone. This completes the Requirements gathering phase, and begins the Design & Planning phase.

Design & Planning

In the third movement of MUSICA Software Development, the Design Lead coordinates the creation of a design to meet the needs of the Milestone. The primary outcome of this phase is a Design Document and an Implementation Plan. Although this phase begins with a clearly defined need in the form of a Milestone, it is expected that additional questions will arise in the course of creating the design. These questions are communicated to the Science Champion through comments on the Milestone Document or the Design Document. The Design Lead will note the technical risks, security concerns, and resources in the Milestone Document as they are identified. The Design Lead is responsible for ensuring these issues are addressed. *Again, it is important that all clarifying information about the science needs be documented in the Milestone Document or the Design Document so the whole team has visibility of the science needs.*

The design process will adhere to the standards for MUSICA software development. An standardized approach to leveraging new hardware and software solutions, once developed, will be referenced here.

Once the Design Document and Implementation Plan are completed, a timeline (in approximate SE-hours) will be provided by the Design Lead to the Science Champion. At this point, the Milestone is ready for implementation, but the next phase will not begin until all Prerequisite Milestones have been completed and the MUSICA Leads allocate developer resources to the implementation of the Milestone.

The goal is that over time a collection of Milestones will advance to the point of having Implementation Plans and timelines. This will allow the MUSICA Leads to prioritize Milestones and schedule development over longer periods of time.

Implementation

Once a Milestone has an Implementation Plan in place, it is ready to be developed. The first step in this fourth movement of MUSICA Software Development is to identify the Milestone Team members. See the section on MUSICA Development Ensembles for more details of the team composition.

MUSICA software packages are maintained in individual GitHub repositories. If the Milestone involves the creation of a new software package, a new GitHub repository is created. Each development task is added to the GitHub repository as an issue attached to the MUSICA Project Board and placed in the Backlog column.[4]

With the Milestone Team selected and the GitHub Issues created, development begins and is scheduled in sequential 2-week sprints until the Milestone objectives are fulfilled. At the beginning of each sprint, the members of an active Milestone Team meet and each team member selects development tasks for the upcoming sprint and recaps the previous sprint by presenting a demo if possible. The team members can use the effort points assigned to each task in the implementation plan to gauge how many tasks to take on for the 2-week period (1 point ≃ 4 hours work). The Milestone Team will meet in regular short standup meetings a minimum of once a week. These meetings should be scheduled for short durations (15 or 20 minutes) and their purpose is strictly to let team members understand how the work is

[4] If you need access to the MUSICA Project Board or other resources email musica-support@ucar.edu

progressing. If a team member is blocked and needs additional time, the Design Lead will identify the people who can best address the issue and schedule a new meeting.

Work will continue in sprints until the Milestone is completed at which point a retrospective with the Milestone Team will take place.

Retrospectives

A retrospective is a reflection done by the Milestone Team that introspects on the quality and correctness of the work and the flow of the work process. A retrospective occurs once all tasks associated with a Milestone have been completed. Retrospectives serve to ensure that scientists can verify the deliverable for scientific correctness and to allow team members to provide feedback about how the work progressed and what is working well or not for the teams.

First, the product is demonstrated to the Milestone Team members. Any software additions, within reason, should be demonstrated live or results displayed if long runtimes are inhibitory. Where applicable, the deliverable for the Milestone should include documentation and instructions on how to use the new feature. Discussions should be recorded in the folder for that milestone.

Science testing can then take place. All manually run tests should be documented along with their results in the Milestone folders. If any issues arise during science testing, additional development issues can be created on the GitHub repository associated with the Milestone by any scientist or developer involved in the retrospective, and scheduled in an upcoming sprint. Once the scientists on the Milestone Team are satisfied that the Milestone objectives have been fulfilled, the Science Champion indicates on the Milestone Document that the development is ready for deployment.

Retrospectives can be as many meetings as required to ensure scientific correctness, but typically only one meeting is necessary. Crucially, any problems that occurred should be examined by the team and any actionable steps that can be taken within reason should be applied to future teams and may reflect as changes in this document.

Deployment

Science components of MUSICA are maintained in individual GitHub repositories and apply semantic versioning (version numbers `<major>.<minor>.<patch>`; e.g., version 2.3.1). The final movement of MUSICA development involves releasing the new or updated software package to the community. A new, unique version number is always associated with the release of the development associated with a Milestone. This new version number is created as soon as the development is ready for deployment and is formed as follows:

- **If the development is in a new GitHub repository…**

    - Version 0.1.0 is applied if the API is expected to change frequently as further development (for different Milestones) continues.
    - Version 1.0.0 is applied if the API is stable and not expected to change frequently.
- **If the development is in an existing GitHub repository in initial development (major version 0; e.g., version 0.3.2)...**
    - If the API is now stable and not expected to change frequently, the development is released as version 1.0.0
    - Otherwise, the minor version number is incremented by 1 (e.g., version 0.3.2 becomes 0.4.0)
- **If the development is in an existing GitHub repository with major version > 0 (e.g., 1.3.2)...**
    - If the development adds new functionality with backwards compatibility of the API, the minor version number is incremented by 1 (e.g., version 2.4.1 becomes version 2.5.0)
    - If the development changes the API such that it is not backwards compatible, the major version is incremented (version 2.4.1 becomes version 3.0.0)

Each release should include in the GitHub release description the features and/or bug fixes addressed in the development.

Individual science components are also included in the MUSICA GitHub repository. This repository includes all individual science components as git submodules, along with build scripts to build a single static MUSICA software library comprising any or all science components (the latest version or any previous version). Users can also choose to build the MUSICA science components as a set of dynamic libraries. These libraries can then be linked to host applications, including 3D models.

The MUSICA collection also applies semantic versioning. The API for the MUSICA collection is considered to be the build script itself. Updates to individual science components increment the minor version number. If the MUSICA build script is modified such that it is not backwards compatible, the major version is incremented.

New MUSICA collection versions are released quarterly (beginning December 2022) and include the latest version of each science component codebase.

MUSICA Development Ensembles

Development Style

The team structure and development style recommended here reaches for a balance between inclusion of NCAR personnel and community partners as well as software development that promotes correctness, efficiency, and clean design.

MUSICA development is a form of modified SCRUM. Development is modified in that each milestone is planned before time, rather than being planned per sprint. However, modifications to the Implementation Plan are expected to occur during development as new issues arise. The Design Lead is encouraged to define a process that works best for the team and not adhere to strict guidelines that limit the team's effectiveness.

Team Composition

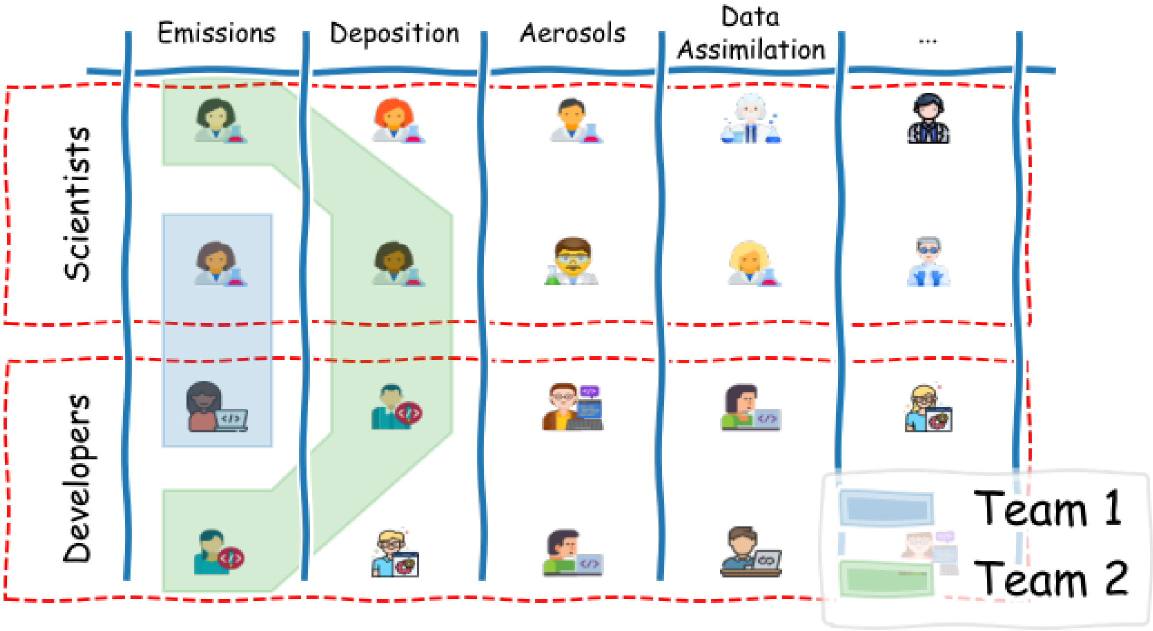

Figure 2[5]. This shows two of any possible number of Milestone Team combinations. A team consists of any number of scientists and SEs, which may be from different labs or even outside of UCAR/NCAR that are together working on one Milestone.

Software development should be a collaborative effort. Development teams will consist of a group of people working together to carry out the work defined in a Milestone. When a Draft Milestone is created, the first two members of the team are selected (the Science Champion and Design Lead).

Once the Milestone is selected for development, additional team members are included. The Science Champion is typically most active during the planning phase for defining requirements and the retrospective phase for verification and deployment. The Design Lead is active at all stages. While teams may be the same from one Milestone to the next, the composition of the team should be determined by the nature of the work, but smaller team sizes should be

[5] Icon attribution

preferred. At minimum, a Milestone Team will consist of the personnel listed as leads for a particular Milestone and may include as many people as are interested and that the Milestone leader(s) deem necessary. The team for each Milestone can be composed of NCAR/ACOM staff, SEs and/or scientists from other NCAR labs, and atmospheric chemistry collaborators from outside NCAR.

Outside Contributions

Each software project will have a contributors document that details how new additions can be submitted and accepted as well as criteria that lead to rejection. A contributors document will define how other entities or organizations can contribute to MUSICA and can be found in the GitHub repository and/or documentation for that software project.

This document can help the MUSICA team to say no to outside contributions that are detrimental to the overall health of the project. Where possible, collaborative contributions start with a conversation between at least one ACOM scientist and SE.

MUSICAlity

MUSICA Software Standards

All software must

1. build as a standalone library or executable
2. achieve a minimum of 80% testing coverage and have documented integration tests
3. be runtime configurable
4. have user and developer documentation for each software project, including contributor guidelines
5. impose a common style within the same software repository

Technical Debt

Each GitHub repository will have a standing collection of tech debt issues. SEs can add tech debt issues as they are identified. Addressing tech debt issues is crucial for the long-term health of MUSICA software packages. During each sprint a fraction of SEs time is allocated to address a small number of tech debt issues.

Reporting Metrics

The outcome of the Design phase is an Implementation Plan which has a total estimate of work hours needed to complete a Milestone. The metrics for tracking progress will at minimum be the start and end dates of the Milestone and the total number of business hours between these two dates.

Advisory Board Reporting

Quarterly reporting must be done to notify MUSICA stakeholders (i.e. MUSICA steering committee, advisory board, WG leaders, and users) of the progress and improvements made within the software. Software releases cannot happen without a plan to report the progress. At a minimum reporting will be:

- A quarterly newsletter that details updates to the software and any scientific contributions carried out with the software
    - A link to the current version of this document will be included in the newsletter

Appendix

Life cycle of an actual milestone

Here we describe the progression of MUSICA software development for deploying TUV-x into CESM CAM-Chem/WACCM.

The science need was identified as “As a MUSICA user, I need to calculate photolysis rates in CESM”. A card for this science need was placed on the Trello board in the Run column: TUV-x in CESM on the Trello board. To achieve implementing TUV-x in CESM, a Science Champion, Doug Kinnison, and a Design Lead, Francis Vitt, were identified.

The Milestone was drafted in the TUV-x Milestone Document by the Science Champion with contributions of other interested scientists. The Trello board card includes a link to the Milestone in the TUV-x Milestones Google document. The milestone has a detailed description expressing the current shortcomings of obtaining photolysis rates in CESM and noting the benefit of implementing TUV-x in CESM.

Once the Milestone was drafted by the scientists, the Draft Milestone became a Milestone ready for software design. As the TUV-x software package already existed, there was already a GitHub TUV-x repository, which was listed at the top of the TUV-x Milestone Document. The Design Lead communicated questions to the Science Champion on the Milestone Document.The clarifying information includes, e.g., a bulleted list of what is and what is not part of the software development. For example, it notes which mechanisms will be used for the implementation and that optimization of the code is not part of the software development. Questions from the software design team to clarify steps for developing TUV-x in CESM are also posed in the Milestone documentation. Lastly, any technical risks or security concerns are given.

Once the software design team understood the Milestone description satisfactorily, the design and implementation plan were created. There is a link to the Milestone’s Design Document and Implementation Plan at the bottom of the Milestone Document. The Implementation Plan lists specific tasks to reach the Milestone. Estimates of time needed to complete each task were given, which then allowed the team to know the total time required to complete the Milestone. A priority for each task was also assigned which aided the software developers in choosing which tasks to perform next.

The tasks identified in the Milestone's implementation plan were transferred to Issues on the MUSICA GitHub Project Board into the Backlog column. At the beginning of each Sprint, the SEs moved tasks from the Backlog column to the Staged column. During the Sprint, the SEs performed the tasks. When completed successfully, the task was submitted as a Pull Request to the TUV-x GitHub repository and moved to the Review. After the Pull Request was reviewed and merged, the task was moved to the Done column.

(Will add description of retrospective and deployment phases once these are completed.)

Glossary of terms

Design Lead: A SE that acts as the point of contact on the design team for a Milestone. This person is responsible for requesting clarifying information from the Science Champion for the Draft Milestone, and leading the design and implementation planning for the Milestone once it moves out of draft status. They are also responsible for providing the status of development upon request once the development has begun.

Design Document: A Google Doc that includes the complete design for a particular Milestone. The design should be detailed enough that an Implementation Plan can be developed for it. The document is located in a subfolder of a MUSICA GitHub project folder (in the MUSICA Software Development Plans folder) labeled "Milestone X".

Draft Milestone: A science need that has a Science Champion and has been added to a Milestone Document, but has not yet been clarified to the point that a design and implementation plan can be developed. Draft Milestone headings in the Milestone Documents are in **red**. Template documents can be found here.

Implementation Plan: A Google Sheets document that includes all the development tasks required to implement the design of a Milestone. The document is located in a subfolder of a MUSICA GitHub project folder (in the MUSICA Software Development Plans folder) labeled "Milestone X". Each line item in the plan corresponds to an issue in Github. Template documents can be found here.

Milestone: A science need that has a Science Champion, and has been clarified to the extent that the SEs understand what the requirements are to address this need. Milestones are documented in Milestone Documents. Milestone headings in the Milestone Documents that are in the Design & Planning Stage are in **yellow**; those that have an Implementation Plan are in **green**.

Milestone Document: A Google Doc in a subfolder of the MUSICA Software Development Plans folder that contains the Milestones for a particular MUSICA GitHub project.

Milestone Team: A small team comprising scientists and SEs that are responsible for achieving the goals of a specific Milestone.

Prerequisite Milestone: A Milestone that, for technical reasons, must be completed before the implementation of another Milestone can begin.

Science Champion: A scientist that acts as the point of contact for questions from SEs related to a Draft Milestone or Milestone. This person will be responsible for having the questions addressed, either by personally answering them, or through communication with other scientists.

Icon attribution

- Developer icons
- Scientist icons
- Github icon
- Lightbulb icon
- Google doc thing
- Executable
- Train

Change Log

v0.1 - Initial draft of the Software Development Plan