

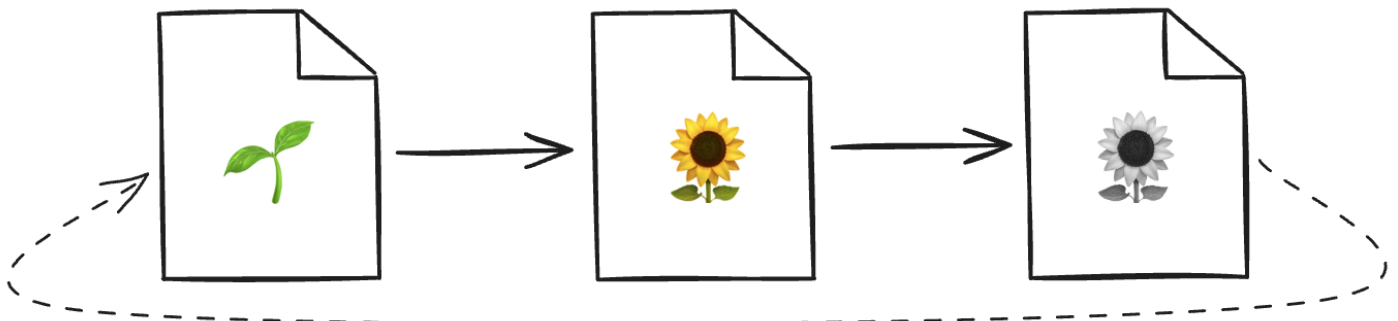
# Welcome

## Contents

- Contributing
- Garden Map

Welcome to the Software Gardening Almanac, an open-source handbook of applied guidance and tools for sustainable software development and maintenance.

## Inspiration



*Fig. 1 Software is created, grows, and decays over time.*

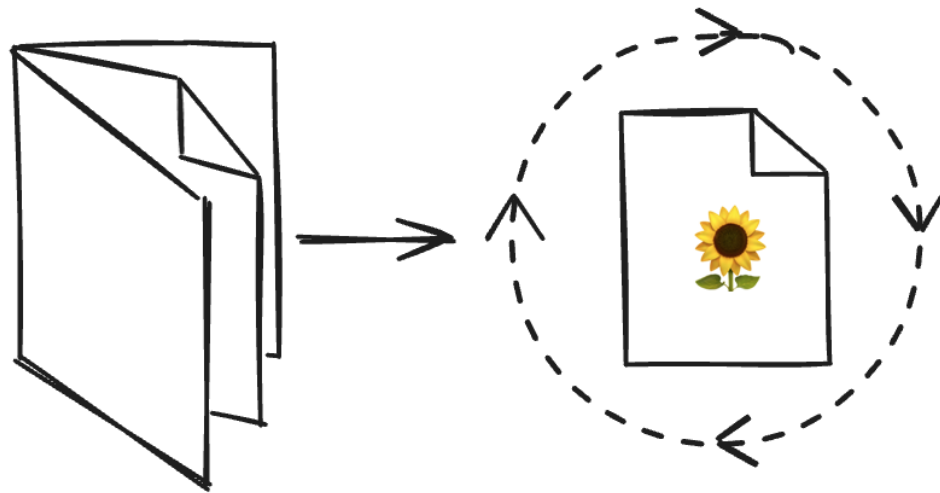
Software experiences development cycles, which accumulate errors over time. However, these cycles are not well understood nor are they explicitly cultivated with the impacts of time in mind. Why does software grow quickly only to decay just as fast? How do software bugs seem to appear in unlikely scenarios?

These cycles follow patterns from life: software is created, grows, decays, and so on (sometimes in surprising or seemingly unpredictable ways). Software is also connected within a complex system of relationships (similar to the complex ecology of a garden). The Software Gardening Almanac posits we can understand these software lifecycle patterns and complex relationships in order to build tools

[Skip to main content](#)

*“The ‘planetary garden’ is a means of considering ecology as the integration of humanity – the gardeners – into its smallest spaces. Its guiding philosophy is based on the principle of the ‘garden in motion’: do the most **for**, the minimum **against**.” - Gilles Clément*

The content within the Software Gardening Almanac is inspired by ecological systems (for example, as in [planetary gardening](#)). We also are galvanized by the [scientific method](#), and [almanacs](#) (or earlier [menologia rustica](#)) in documenting, expecting, and optimizing how we plan for time-based changes in agriculture (among other practices and traditions).



*Fig. 2 Almanacs help us understand and influence the impacts of time on the things we grow.*

The Software Gardening Almanac helps share knowledge on how to cultivate change in order to nurture software (and software practitioners) for long periods of time. We aspire to define, practice, and continually improve a craft of *software gardening* to nourish existing or yet-to-be software projects and embrace a more resilient future.

# Motivation



Fig. 3 We depend on a delicate network of software which changes over time.

(Image source: [‘Dependency’ comic by Randall Munroe, XKCD](#))

Our world is surrounded by systems of software which enable and enhance life. These systems are both invisible and sometimes brittle, breaking in surprising ways (for example, beneath uneven pressures as in the above XKCD comic). At the same time, [loose coupling](#) of these software systems also forms the basis of innovation through multidimensional, [emergent](#) growth patterns. We seek to embrace these software systems which are innately in motion, embracing diversity without destroying it to perpetuate discovery and enhance future life.

“Start where you are. Use what you have. Do what you can.” - Arthur Ashe

We suggest reading or using this content however it best makes sense to a reader. Just as with gardening, sometimes the best thing to do is jump in! We structure each section in a modular fashion, providing insights with cited prerequisites. We provide links and other reference materials for further reading where beneficial. Please let us know how we can improve by [opening GitHub issues](#) or [creating new discussion board posts](#) (see our [contributing.md](#) guide for more information)!

[Skip to main content](#)

# Who's this for?

*"We'll share stories from the heart. All are welcome here."* - Alexandra Penfold

The Software Gardening Almanac is designed for developers of any kind. When we say *developers* here we mean anyone engaging in pursuing their vision towards a goal using software (including scientists, engineers, project leads, managers, etc). The content will engage readers with pragmatic ideas, keeping the barrier to entry low.

## Garden Circle

The garden circle is a section of the Software Garden Almanac associated with contributions, planning, and internal maintenance of the project.

## Contributing

First of all, thank you for contributing to the Software Gardening Almanac! 🎉 100 🌱 We're so grateful for the kindness of your effort applied to grow this project into the best that it can be.

This document contains guidelines on how to most effectively contribute to this project.

If you would like clarifications, please feel free to ask any questions or for help. We suggest asking for help from GitHub where you may need it (for example, in a GitHub issue or a pull request) by "[at \(@\) mentioning](#)" a Software Gardening Almanac maintainer (for example, `@username`).

## Code of conduct

Our [Code of Conduct \(CoC\) policy](#) (located [here](#)) governs this project. By participating in this project, we expect you to uphold this code. Please report unacceptable behavior by following the procedures listed under the [CoC Enforcement section](#).

[Skip to main content](#)

# Security

Please see our [Security policy](#) (located [here](#)) for more information on security practices and recommendations associated with this project.

## Quick links

- Documentation: [🔗 software-gardening/almanac](https://github.com/software-gardening/almanac)
- Issue tracker: [🔗 software-gardening/almanac#issues](https://github.com/software-gardening/almanac#issues)

## Process

### Reporting bugs or suggesting enhancements

We're deeply committed to a smooth and intuitive user experience which helps people benefit from the content found within this project. This commitment requires a good relationship and open communication with our users.

We encourage you to file a [GitHub issue](#) to report bugs or propose enhancements to improve the Software Gardening Almanac.

First, figure out if your idea is already implemented by reading existing issues or pull requests! Check the issues ([🔗 software-gardening/almanac#issues](https://github.com/software-gardening/almanac#issues)) and pull requests ([🔗 software-gardening/almanac](https://github.com/software-gardening/almanac)) to see if someone else has already documented or began implementation of your idea. If you do find your idea in an existing issue, please comment on the existing issue noting that you are also interested in the functionality. If you do not find your idea, please open a new issue and document why it would be helpful for your particular use case.

Please also provide the following specifics:

- The version of the Software Gardening Almanac you're referencing.
- Specific error messages or how the issue exhibits itself.
- Operating system (e.g. MacOS, Windows, etc.)

[Skip to main content](#)

---

- Any examples of similar capabilities

## Your first code contribution

Contributing code for the first time can be a daunting task. However, in our community, we strive to be as welcoming as possible to newcomers, while ensuring sustainable software development practices.

The first thing to figure out is specifically what you're going to contribute! We describe all future work as individual [GitHub issues](#). For first time contributors we have specifically labeled as [good first issue](#).

If you want to contribute code that we haven't already outlined, please start a discussion in a new issue before writing any code. A discussion will clarify the new code and reduce merge time. Plus, it's possible your contribution belongs in a different code base, and we do not want to waste your time (or ours)!

## Pull requests

After you've decided to contribute code and have written it up, please file a pull request. We specifically follow a [forked pull request model](#). Please create a fork of Software Gardening Almanac repository, clone the fork, and then create a new, feature-specific branch. Once you make the necessary changes on this branch, you should file a pull request to incorporate your changes into the main The Software Gardening Almanac repository.

The content and description of your pull request are directly related to the speed at which we are able to review, approve, and merge your contribution into The Software Gardening Almanac. To ensure an efficient review process please perform the following steps:

1. Follow all instructions in the [pull request template](#)
2. Triple check that your pull request is adding *one* specific feature. Small, bite-sized pull requests move so much faster than large pull requests.
3. After submitting your pull request, ensure that your contribution passes all status checks (e.g. passes all tests)

We require pull request review and approval by at least one project maintainer in order to merge. We will do our best to review the code additions in as soon as we can. Ensuring that you follow all steps above will increase our speed and ability to review. We will check for accuracy, style, code coverage.

[Skip to main content](#)

## Git commit messages

For all commit messages, please use a short phrase that describes the specific change. For example, “Add feature to check normalization method string” is much preferred to “change code”. When appropriate, reference issues (via `#` plus number) .

## Development

### Overview

We write and develop the Software Gardening Almanac in [Python](#) through [Jupyter Book](#) with related environments managed by Python [Poetry](#). We use [Node](#) and [NPM](#) dependencies to assist development activities (such as performing linting checks or other capabilities). We use [GitHub actions](#) for [CI/CD](#) procedures such as automated tests.

### Getting started

To enable local development, perform the following steps.

1. [Install Python](#) (suggested: use `pyenv` for managing Python versions)
2. [Install Poetry](#)
3. [Install Poetry environment](#): `poetry install`
4. [Install Node](#) (suggested: use `nvm` for managing Node versions)
5. [Install Node environment](#): `npm install`
6. [Install Vale dependencies](#): `poetry run vale sync`

### Development Tasks

We use [Poe the Poet](#) to define common development tasks, which simplifies repeated commands. We include Poe the Poet as a Python Poetry `dev` group dependency, which users access through the Poetry environment. Please see the `pyproject.toml` file’s `[tool.poe.tasks]` table for a list of available tasks.

[Skip to main content](#)

For example:

```
# example of how Poe the Poet commands may be used.
poetry run poe <task_name>

# example of a task which runs jupyter-book build for the project
poetry run poe build-book
```

## Linting

[pre-commit](#) automatically checks all work added to this repository. We implement these checks using [GitHub Actions](#). Pre-commit can work alongside your local [git with git-hooks](#). After [installing pre-commit](#) within your development environment, the following command performs the same checks:

```
% pre-commit run --all-files
```

## Automated CI/CD with GitHub Actions

We use [GitHub Actions](#) to help perform automated [CI/CD](#) as part of this project. GitHub Actions involves defining [workflows](#) through [YAML files](#). These workflows include one or more [jobs](#) which are collections of individual processes (or steps) which run as part of a job. We define GitHub Actions work under the `.github` directory. We suggest the use of `act` to help test GitHub Actions work during development.

## Attribution

We sourced portions of this contribution guide from `pyctyominer`. Big thanks go to the developers and contributors of that repository.

## Garden Map

Our garden map is where you can learn about what features we're working on, what stage they're in, and when we expect to bring them to you. The content here follows that of a [technical roadmap](#) (providing strategic visibility on implementation details and timelines)

[Skip to main content](#)



# Ethos

We intend for the garden map to provide useful tools for us to efficiently create and maintain the Software Gardening Almanac. We practice keeping the garden map as “petal-light” as possible through built-in tools, automation, and an avoidance of [analysis paralysis](#) where possible.

## We’d love your input

We welcome your input on the garden map and elements found within it! Please see the [contributing](#) guide for more information on how to participate in this process.

## Active garden map(s)

The following are active garden maps for the Software Gardening Almanac. Please see below for a further description of these projects.

- [2024 BSSw Fellowship Milestones](#): this project communicates activities associated with the 2024 [BSSw Fellowship](#) associated with the Software Gardening Almanac.

## Sections

You can explore the garden map with different views:

- **“Canopy-view” development visibility**: We track new, existing, and previous work at a high-level using organization-wide [GitHub Projects](#).
- **“Ground-level” development visibility**: We track repository-specific work using [GitHub issues](#) and a GitHub Project field called “status”, which we associate with each issue (indicated as “Todo”, “In progress”, “Paused”, or “Done”).
- **Major milestones**: We track major milestones using common [GitHub issues labels](#) across all repositories through their associated issues.

[Skip to main content](#)

## Visualizations

It can be helpful to visualize the garden map using data plots or other images. We use [GitHub Projects insights](#) to assist with automated creation and updates to garden map activity. Please see this [project insights page](#) for an example of these (using the insights button within any GitHub Projects for viewing others).

## Acknowledgments

We drew inspiration for this content from the [GitHub public roadmap](#). Big thanks to the original works found there.