Windsurf + Spec Kit

My notes on using spec kit on a Greenfield project

Amit; Nov 26th, 2025

About Spec Kit

What is spec-kit and how is it used?

- All AI Tools, even code assistants, need prompts
- to arrive at repeatable prompts that are time-tested/ refined, we develop prompts and capture them into "Prompt Libraries" for our AI Tools to use.
- in software development, this concept has evolved further into Specification-Driven-Development. there are a few key-players:: OpenSpec and SpecKit
- GitHub has launched Spec Kit; its open source; gives very good guidelines which we all can use and extend
- Spec Kit comes with python helper scripts which you can use to add these templates into your project. Spec Kit supports several AI Code Assistants, (personally i am using Windsurf)
- one way of setting up your project, is to use UV package manager and Speckits command line interface
- alternate way is to download the zip-file of templates/scripts and unzip into your project
- once you have speckit activated in the project, you can use slash-commands with Coding Assistants
-

References

Handy links to documentation, explanation

- From the creators:
    - https://github.com/github/spec-kit
- Youtube videos:
    - https://www.youtube.com/watch?v=a9eR1xsfvHg. ; https://www.youtube.com/watch?v=SGHIQTsPzuY. ; https://www.youtube.com/watch?v=8jtIXRyGMQU
- Windsurf links:
    - https://docs.windsurf.com/windsurf/cascade/workflows
- Similar alternative
    - https://github.com/Fission-AI/OpenSpec

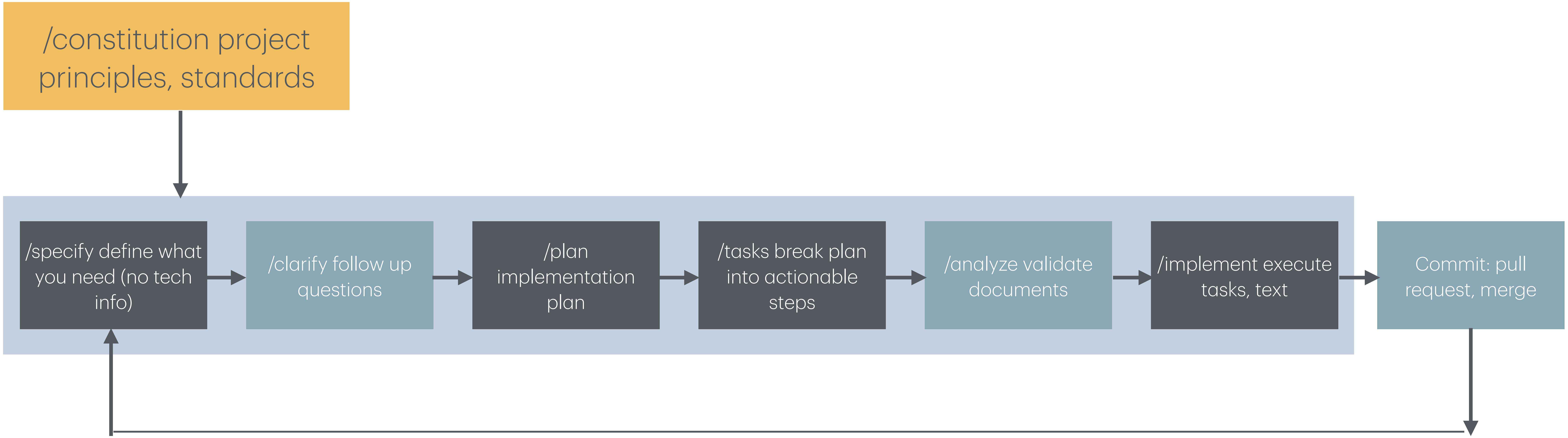
Comparing with OpenSpec

Two similar toolkits

Aspect	Speckit	OpenSpec
Purpose and Focus	Framework for writing and managing specifications	Open-source collaboration for specification-driven development
Integration with AI	Direct integrations with AI coding assistants	Similar integrations
Usability	Polished user interface for enhanced productivity	May require more technical knowledge; customizable
Community and Support	Typically backed by commercial support	Relies on community support; responsiveness varies
Flexibility and Extension	Structured extensibility for easy build-on	Encourages flexibility; open-source allows custom tweaks
Documentation, Learning Curve	Comprehensive documentation lowers learning curve	Depends on community contributions; may vary in quality

Command Workflow

- Basic: constitution —> specify —> plan —> tasks —> implement
- Alternate: constitution -> specify -> clarify -> plan -> tasks -> analyze -> implement



/constitution project principles, standards

/specify define what you need (no tech info)

/clarify follow up questions

/plan implementation plan

/tasks break plan into actionable steps

/analyze validate documents

/implement execute tasks, text

Commit: pull request, merge

Sample project setup

A Weather app.. exploratory data analysis

- Pre-requisites:
  - Python ^3.12; UV from Astral; Speckit (optional or its zip folder)
- Project, speckit setup:
  - Either use speckit init command or unzip the templates into project-root folder.
- Optional: use WeatherData CSV file (kaggle or NOAA)

Step 1: speckit.constitution

Give this slash-command followed by the prompt below

- a single page web application showing exploratory analysis of weather data from its local folder. application should adhere to SOLID principles (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion). application should also use at least 10 best practices for using simple Next.JS framework. it should have a test coverage of 90%

Step 2: speckit.specify

Give this slash-command followed by the prompt below

- the application provides services to get average temperature and humidity per month over the year from the weather data in resources folder of the application. it also provides a service which will for a given month show the trend of temperature and humidity and most common weather condition in that month. the application on the webpage shows two charts side by side for average temperature and humidity over the year. it them provides user option to select a month to see the weather trends for that month

Step 3: speckit.clarify

- for this step, just run the slash command
- when Windsurf asks questions for clarification, answer with appropriate options

Step 4: speckit.plan

Give this slash-command followed by the prompt below

- use python 3.12 and UV package manager from Astral which is installed / available on this project's root folder. use FastAPI or a better library to build the backend services. WeatherData.csv available in src/main/resources folder, is the data to be used for this application. use panda library to read CSV data, use matplotlib and seaborn to create charts, use Next.JS to create a responsive web application with requested features

Step 5: speckit.tasks

Give this slash-command followed by the prompt below

- this project has speckit templates. the spec.md and plan.md for the application are located at ./specs/001-weather-analysis folder. use these and create tasks for this plan. In phase 1, have tasks for creating backend services for user story 1, then user story 2, then sequence the tasks for creating the frontend application. have task for implementing unit test cases at the right sequence. i would like to build one feature at a time.

Step 7: speckit.implement

Give this slash-command followed by the prompts below

- Ask AI assistant to implement only certain tasks at time
- Or you can ask it to implement phase by phase, where phases are a collection of tasks in a defined sequence.