Implementing CI/CD Philosophy for CTSM

The tale of b4b-dev

Erik Kluzek,

CGD/TSS/CSEG SE

June 11th, 2024

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

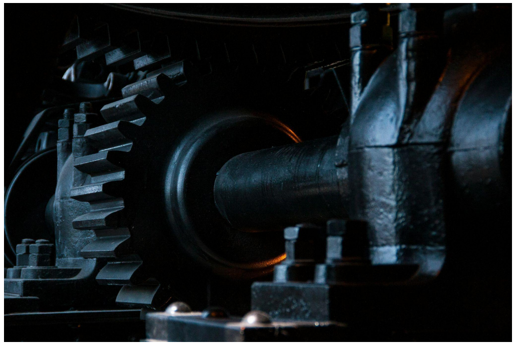
Outline

- Automation with CI/CD is awesome!
- Automating testing with CI/CD is something we should all do
- I'm NOT talking about Automated testing here though!
- CI/CD originally was a concept about how to build software
- I'm going to talk about that and how we adopted some practices that help move us towards the philosophy of CI/CD
- The main one is our ***new b4b-dev branch*** using a git-flow workflow
- Which was a team effort
- I'll go through some history of that effort
- Another practice is our ***"Near term Priorities"*** Project Board
- These have both been positive
- The process in and of itself was a team building exercise
- It also brought our team to have ownership of our process
- And allowed us to tweak the process as needed

Automation With CI/CD Is Awesome!
Something we need/will do more of…

Automated testing means:

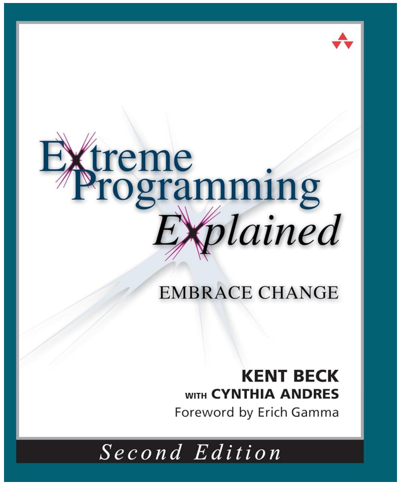
- Adopting SE industry best practices
- We leverage the work we do with tools
- Increase our output
- Do things faster
- Catch problems sooner
- Find integration problems rapidly
- Standardizes our software process



I’m Not Talking About Automated Testing Here Though

History:

- 1984 C/I term coined by Grady Booch
- 1987 Kent Beck X-programming
- 2006 Martin Fowler article
- Now – ***commit to same branch sub-daily*** which invokes automated testing



Extreme Programming Explained

EMBRACE CHANGE

KENT BECK
WITH CYNTHIA ANDRES
Foreword by Erich Gamma

Second Edition

C/I Philosophy Goals for CTSM:

- Smaller PR’s/tags
- Commits more often
- Fix little problems quicker
- Integrating together on the shared branch more often

John Stevens Definition of C/I

Continuous integration (CI) is a software engineering practice where members of a team integrate their work with increasing frequency. In keeping with CI practice, teams ***strive to integrate at least daily and even hourly****, approaching integration that occurs "continuous-ly."*

https://www.synopsys.com/blogs/software-security/agile-cicd-devops-difference.html

The Main One Is Our New b4b-dev Branch Using A Git-Flow Workflow

Current:

- I work on tags for about a month
- Tags come into the group about weekly
- I add miscellaneous things to tags I’m working on

Pain points:

- Larger tags make it hard to review
- “Adding things” can be problematic
- Bigger tags mean more integration updates/problems/conflict with other work
- Tag queuing slows down development
- Tags require ChangeLog/baseline creation which disincentives adding something small to a new tag

New:

- “Big” or answer changing things still go to master
- “Miscellaneous things” and quick fixes can go to b4b-dev branch
- b4b-dev branch merged to master every two weeks
- Easier to review smaller changes on b4b-dev
- Small PR’s go in without new baselines and ChangeLog
- b4b-dev is updated for the dev group more often so closer to C/I
- Still testing using our test lists

Eliminate Tagging Bottleneck for Simple Changes

Team Effort

Adrianna Foster

Ryan Knox
FATES LBL

Keith Oleson

Sam Levis

Greg Lemieux
FATES LBL

Sam Rabin

Matvey Debolskiy
NorESM

Will Wieder
LMWG Chair

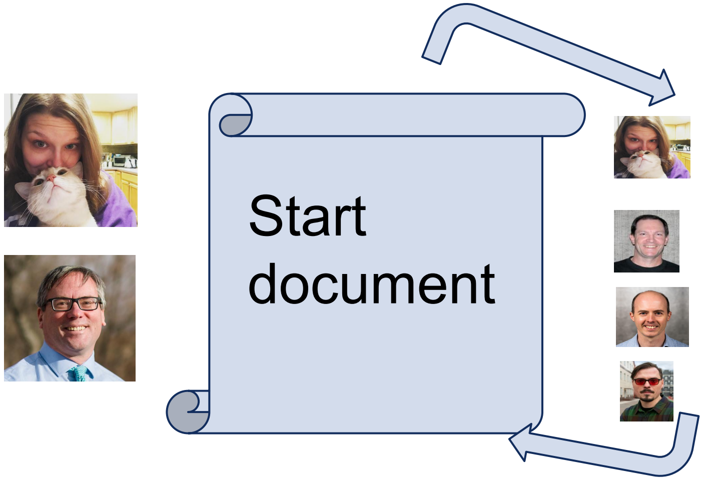
History

We should look into the git flow workflow

X Infinity…

We should start listening to Adrianna..

October…



Start document

Iterate on the document as a team. Over a few months…

Results we've Seen

- Fix for "run_neon" went in a matter of hours vs weeks
- 24 PR's came in under 6 tags on master
- Process is easy – "turn the crank"
- Take turns
- These PR's easier to code review

Important Takeaways

- Working on ***software process*** is critically important
- When people start, educate them in what is currently done – later listen to them on how to improve
- ***Listen to everyone on your team*** – especially when you think you disagree
- Did I not listen to Adrianna because of gender/age/experience bias? Keep thinking about that
- Everyone contributes to the process:
  - Builds ***ownership, buy in,*** and ***personal commitment*** to the process
  - Builds in a way to ***continuously improve***
  - Gets everyone thinking about pain points and how to reduce them
  - Gets everyone working the same way
  - Helps spinning up new people in the process

Listen to the Team

Near Term Priorities Github Board

Problems!

- ***Making good SE time estimates is hard!***
- Making bad estimates is demoralizing
- We don't do it enough
- We haven't tracked our accuracy
- ***Continually need to make estimates to plan and coordinate work***
- Pressure to "get things done" adds to stress, bad estimates, and bad process
- ***Prioritization*** is difficult – making sure we are working on the most important things
- Working on too many things at once – means everything is too slow and not efficient
- Long term planning is too difficult to have a handle on

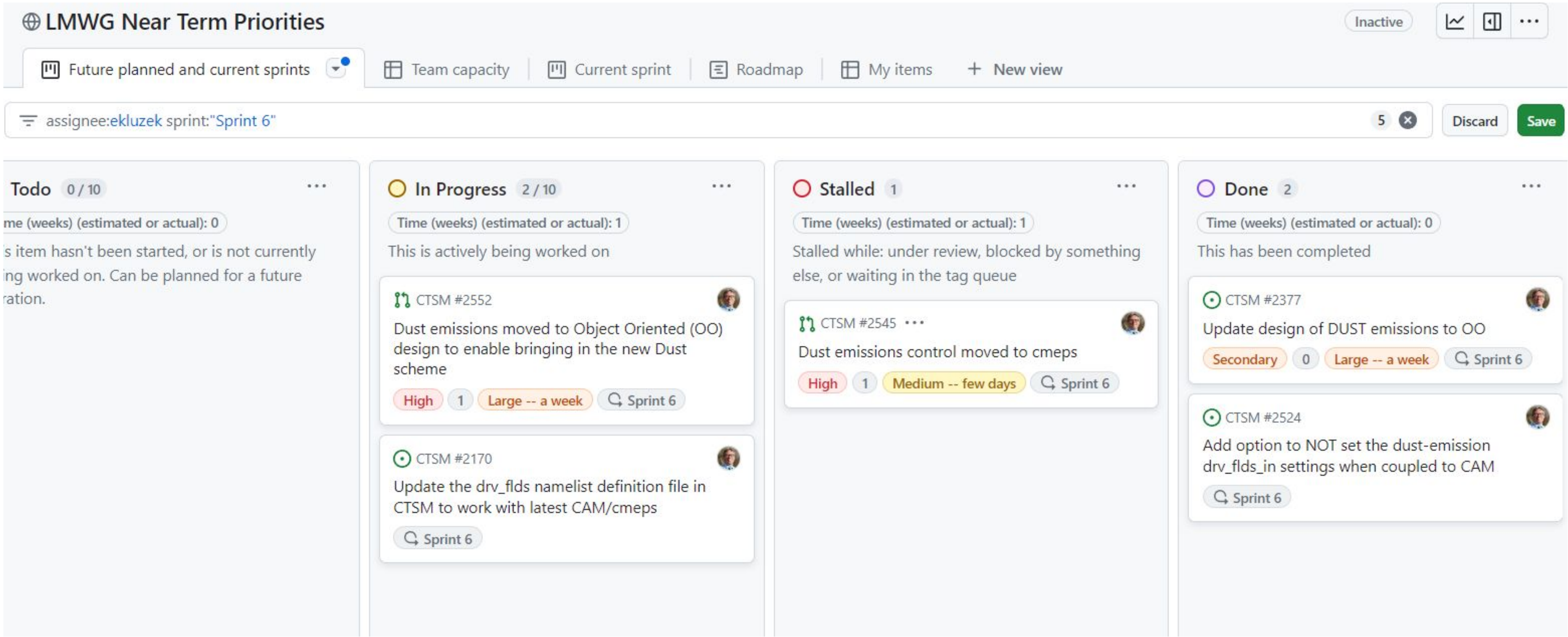
Solutions?:

- Let's use estimation methods from the SE industry
- Recognize estimates vary with actual – track minimum, mean, and maximum guesses (a good estimate is just within the range)
- Average of estimates over time being near reality means estimates are useful
- Let's concentrate on ***near-term goals*** – rather than long term
- ***Constantly assess our prioritization*** is correct
- Practice making estimates for near-term goals, track and then assess afterwards to improve over time
- ***Github "Team Planning Template"?***

Near Term – 3 week Sprints

- Continually assess priorities
- Plan next cycle:
  - Requirements
  - Design
  - Implementation
  - Testing
  - Tag release
- Evaluate previous cycle
- See how the estimates turned out

What does it look like? (current planning)



LMWG Near Term Priorities

assignee:ekluzek sprint:"Sprint 6"

Todo 0 / 10

me (weeks) (estimated or actual): 0

s item hasn't been started, or is not currently ing worked on. Can be planned for a future ration.

In Progress 2 / 10

Time (weeks) (estimated or actual): 1

This is actively being worked on

- CTSM #2552
  Dust emissions moved to Object Oriented (OO) design to enable bringing in the new Dust scheme
  High | 1 | Large -- a week | Sprint 6
- CTSM #2170
  Update the drv_flds namelist definition file in CTSM to work with latest CAM/cmeps
  Sprint 6

Stalled 1

Time (weeks) (estimated or actual): 1

Stalled while: under review, blocked by something else, or waiting in the tag queue

- CTSM #2545 ···
  Dust emissions control moved to cmeps
  High | 1 | Medium -- few days | Sprint 6

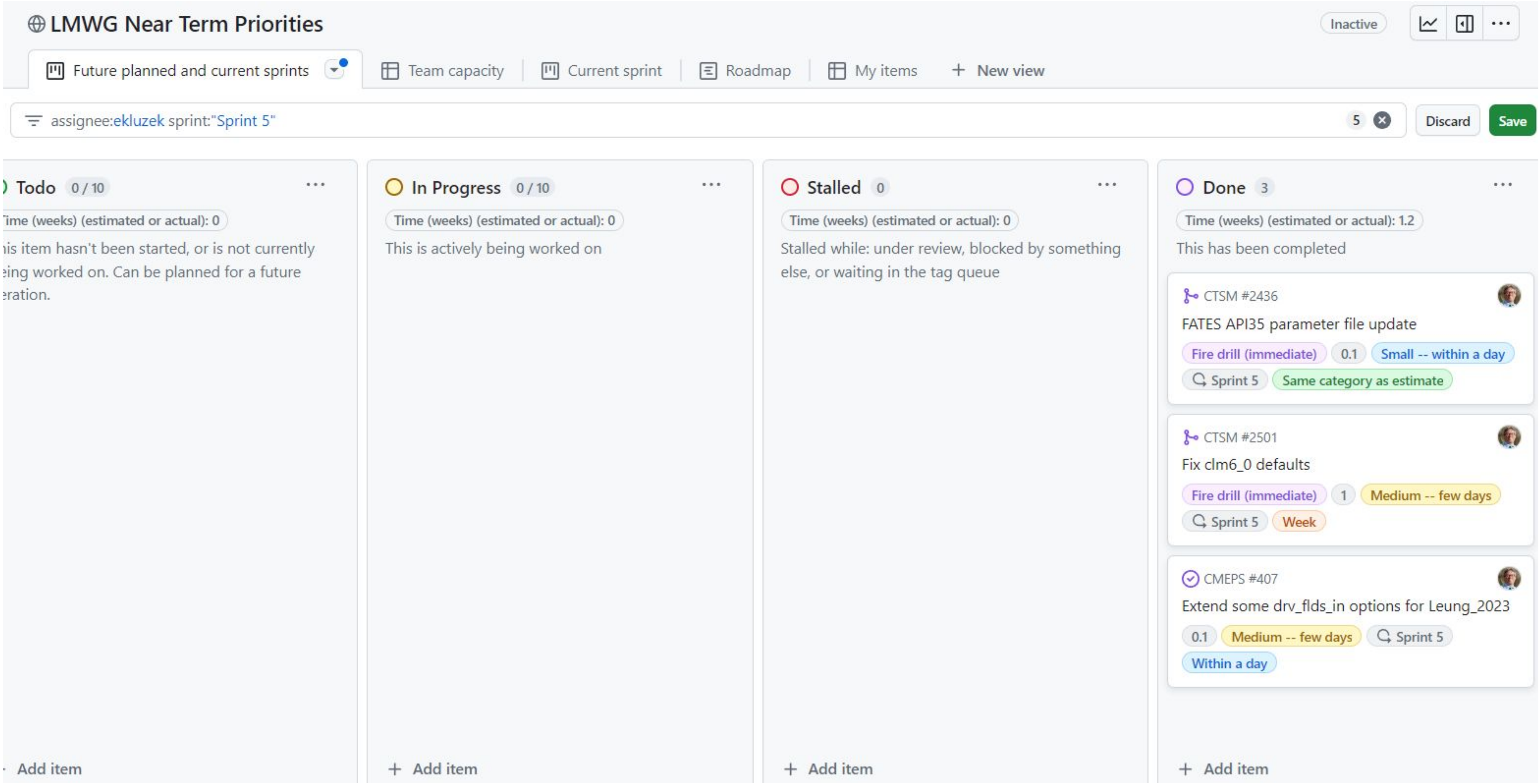
Done 2

Time (weeks) (estimated or actual): 0

This has been completed

- CTSM #2377
  Update design of DUST emissions to OO
  Secondary | 0 | Large -- a week | Sprint 6
- CTSM #2524
  Add option to NOT set the dust-emission drv_flds_in settings when coupled to CAM
  Sprint 6

Previous Sprint (Review)



assignee:ekluzek sprint:"Sprint 5" 5

Todo 0 / 10

ime (weeks) (estimated or actual): 0

is item hasn't been started, or is not currently eing worked on. Can be planned for a future eration.

Add item

In Progress 0 / 10

Time (weeks) (estimated or actual): 0

This is actively being worked on

Add item

Stalled 0

Time (weeks) (estimated or actual): 0

Stalled while: under review, blocked by something else, or waiting in the tag queue

Add item

Done 3

Time (weeks) (estimated or actual): 1.2

This has been completed

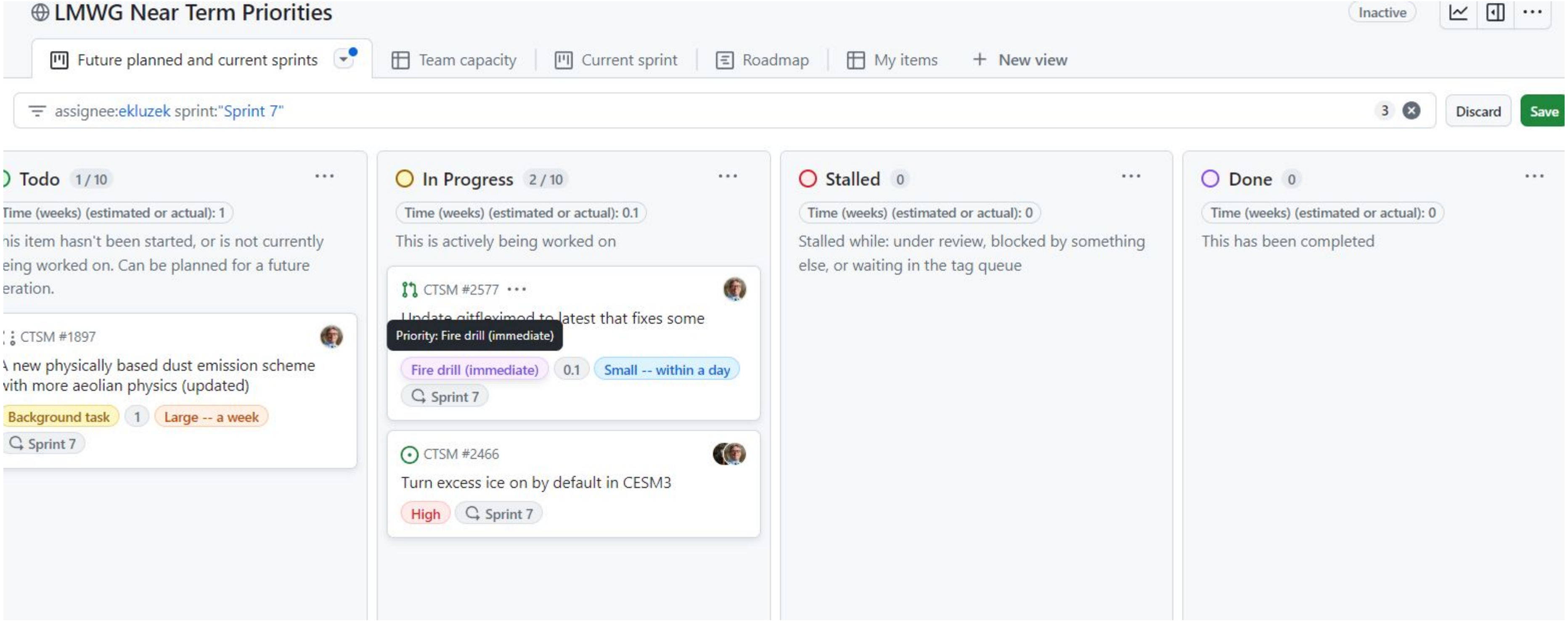
CTSM #2436
FATES API35 parameter file update
Fire drill (immediate) | 0.1 | Small -- within a day | Sprint 5 | Same category as estimate

CTSM #2501
Fix clm6_0 defaults
Fire drill (immediate) | 1 | Medium -- few days | Sprint 5 | Week

CMEPS #407
Extend some drv_flds_in options for Leung_2023
0.1 | Medium -- few days | Sprint 5 | Within a day

Add item

Future Sprint (Planning)



LMWG Near Term Priorities

assignee:ekluzek sprint:"Sprint 7" 3

Todo 1 / 10

Time (weeks) (estimated or actual): 1

his item hasn't been started, or is not currently eing worked on. Can be planned for a future eration.

CTSM #1897

\ new physically based dust emission scheme vith more aeolian physics (updated)

Background task | 1 | Large -- a week | Sprint 7

In Progress 2 / 10

Time (weeks) (estimated or actual): 0.1

This is actively being worked on

CTSM #2577

Update gitfleximod to latest that fixes some

Priority: Fire drill (immediate)

Fire drill (immediate) | 0.1 | Small -- within a day | Sprint 7

CTSM #2466

Turn excess ice on by default in CESM3

High | Sprint 7

Stalled 0

Time (weeks) (estimated or actual): 0

Stalled while: under review, blocked by something else, or waiting in the tag queue

Done 0

Time (weeks) (estimated or actual): 0

This has been completed

Takeaways for All

- Think about our SE processes in your teams!
- Work together to develop and improve your process
- Listen careful to ALL voices
- Try new practices and iterate on them
- Develop practices that address these:
  - Prioritization!
  - Planning for short cycles
  - Working on current cycle
  - Analysis of the previous cycle
  - Keep redoing above on short cycles

References

Steven, J. (2018, Mar 18) *What's the difference between agile, CI/CD, and DevOps?*, Synopsys https://www.synopsys.com/blogs/software-security/agile-cicd-devops-difference.html

CTSM SE Team (2024): *CTSM b4b-dev Dev Branch*, CTSM Drive https://docs.google.com/document/d/17quJfi5trKxoNtY-QOkxwfG-iLcKRROy8HOI80SIpJ0

Porto, P. (2018, Feb 26) *4 branching workflows for Git*, Medium

medium.com/@patrickporto/4-branching-workflows-for-git-30d0aaee7bf

Beck, K. & Andres C. *Extreme Programming Explained: Embrace Change (The XP Series)* (2nd Edition). Addison-Wesley

https://www.amazon.com/Extreme-Programming-Explained-Embrace-Change/dp/0321278658

Fowler, M. (2024, Jan 18, update from 2001) *Continuous Integration*, MartinFowler

https://martinfowler.com/articles/continuousIntegration.html