SAND2022-7000C

Sandia National Laboratories

Exceptional service in the national interest

AGILE PROCESS IMPROVEMENTS INITIATIVE FOR ASC SIERRA CODE DEVELOPMENT TEAMS

25 MAY 2022

PRESENTED BY

RYAN SHAW

AGILE SME, SANDIA NATIONAL LABORATORIES

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND20XX-XXXX P

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

AGENDA

Acknowledgements

History

Challenges

Proposal

Outcomes

Conclusion

ACKNOWLEDGMENTS

Product Owner Leadership Team

- Martin Heinstein
- Charis Church
- Mike Glass
- Nate Crane

Change Agent

- Salome Thorson

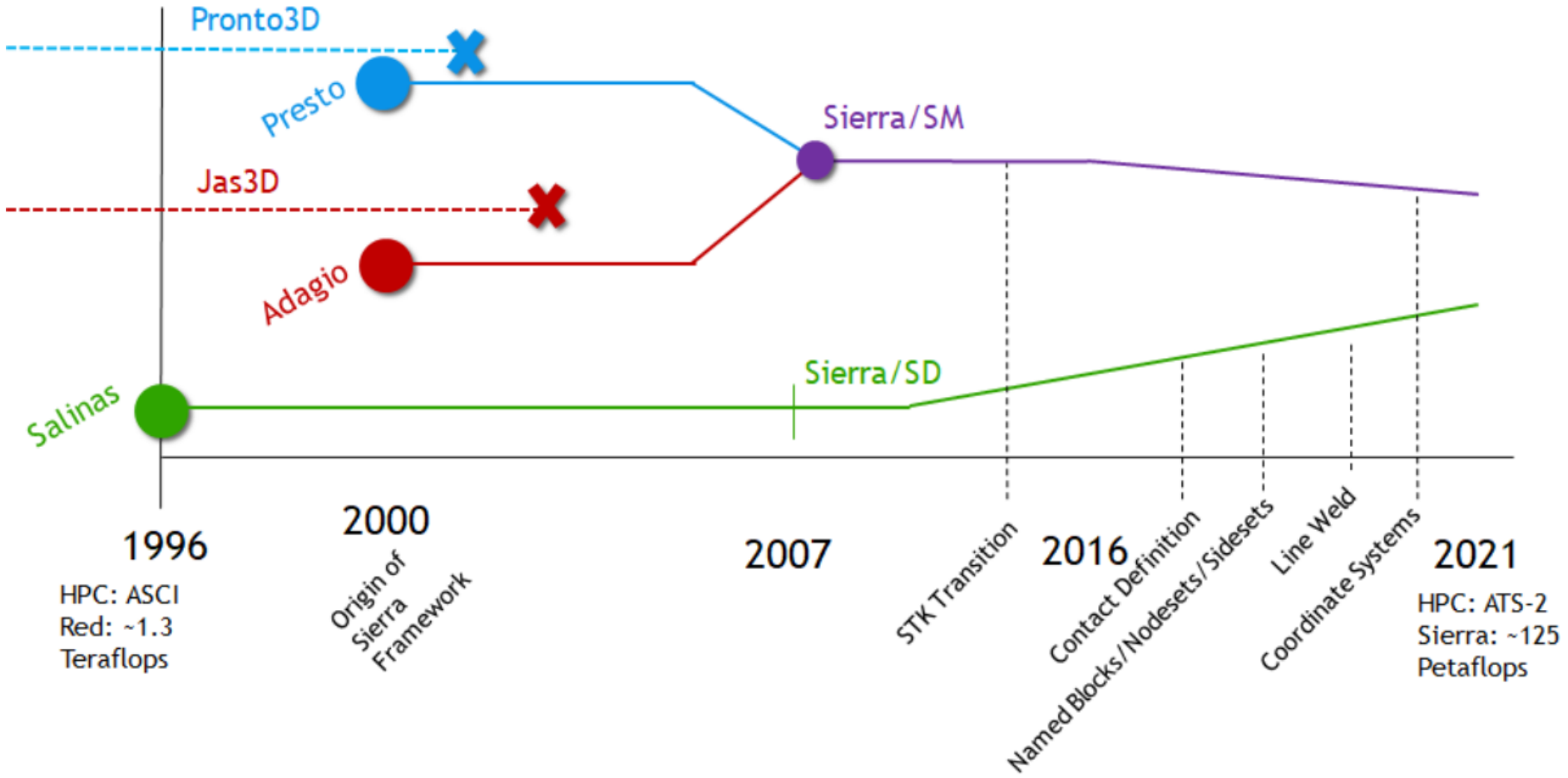
Developer Practices Working Group

- Mark Merewether
- Frank Beckwith
- Payton Lindsay
- Neil Matula
- Matt Bopp
- Tolu Okusanya
- Jon Sykora
- Clint Stimpson
- Jonathan Russ
- Riley Wilson
- Andrew Kurzawski

Product Owner Practices Working Group

- Charis Church
- Todd Coffey
- Jesse Thomas
- Nate Crane
- Alan Williams
- Tim Walsh
- Jesse Thomas

HISTORY OF SIERRA



THE CHALLENGES TO SUCCESS

Agility and responsiveness in the weapon design and development phase.

Commitment to deliver capability on a schedule consistent with upcoming and future ND programs

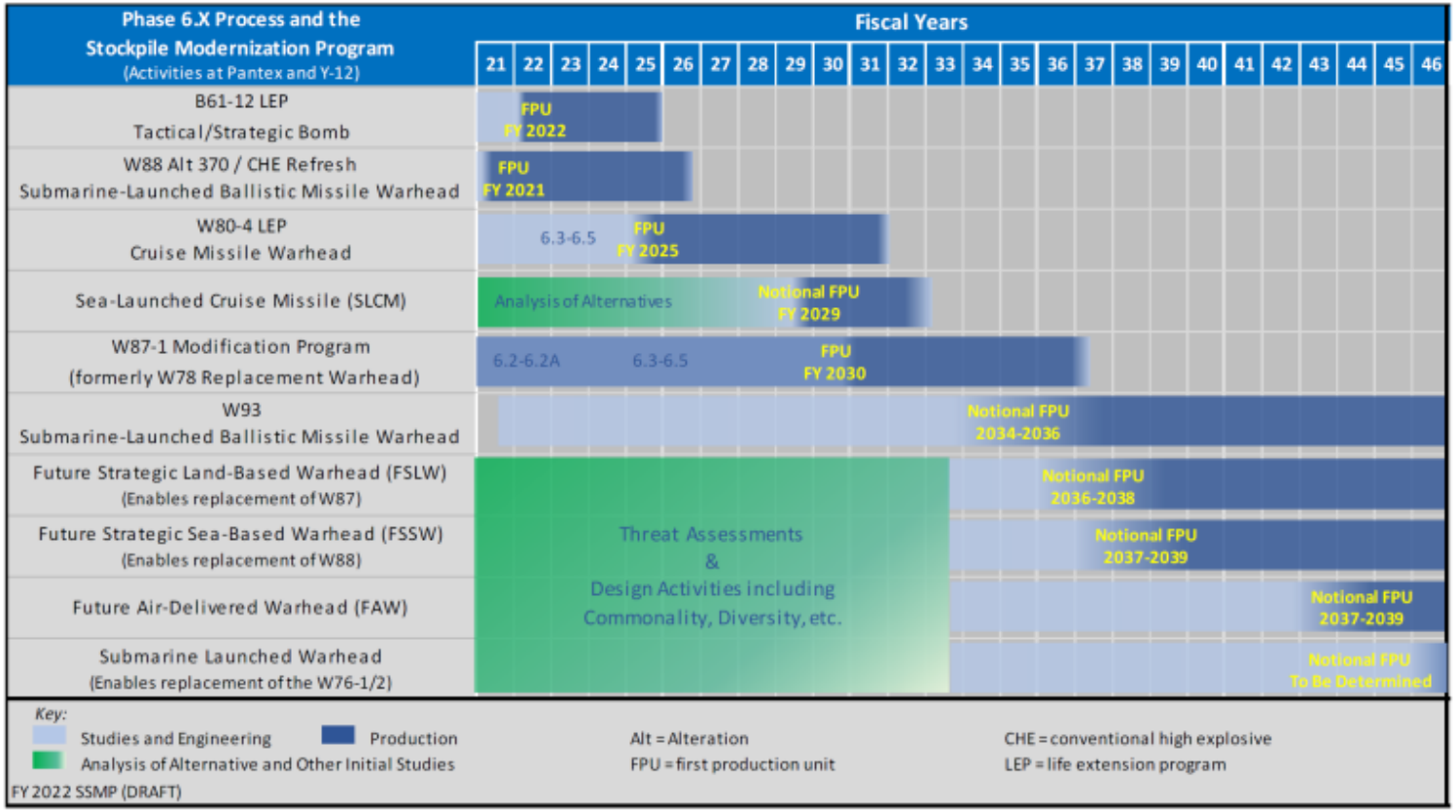


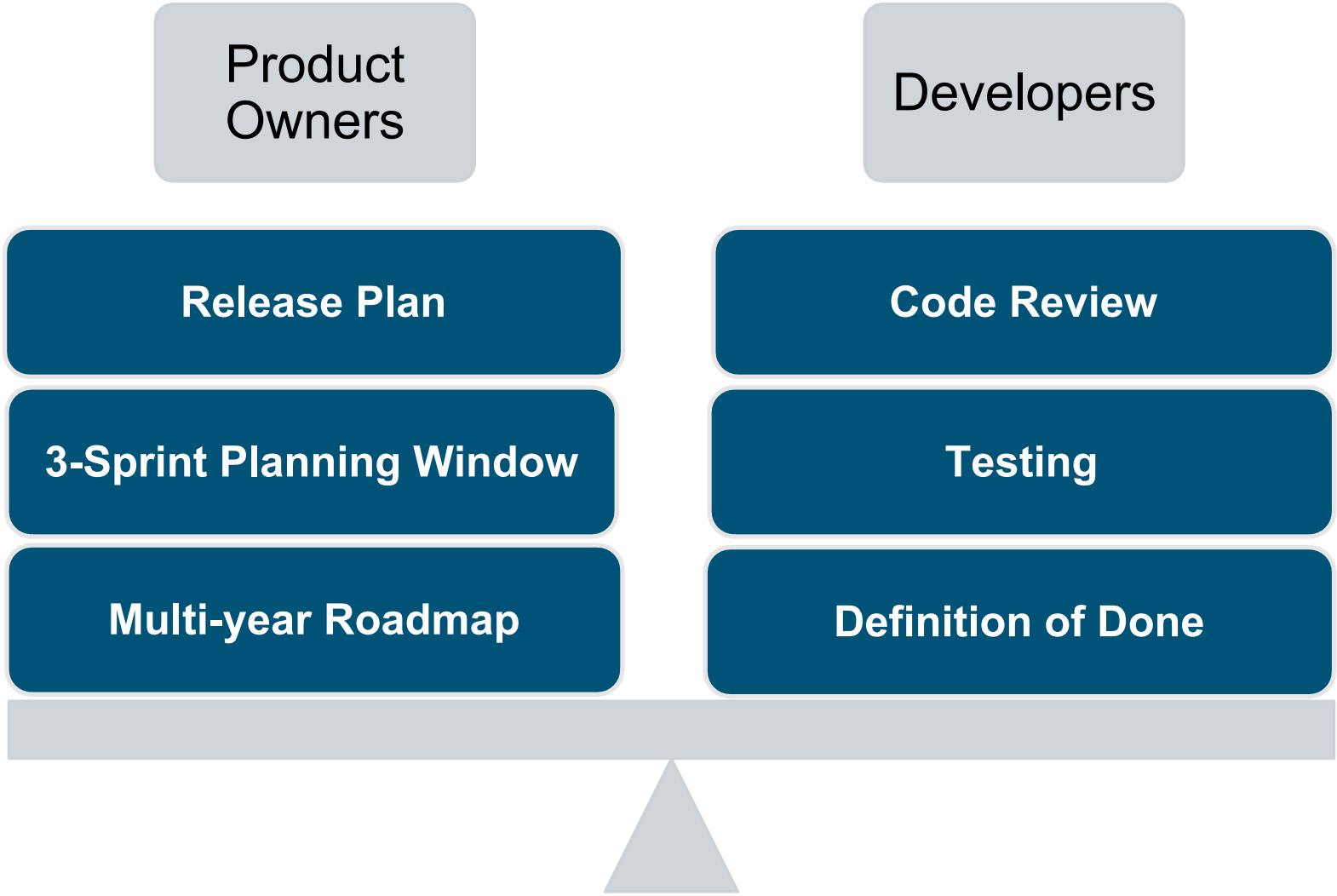
Figure 2–3. DOE/NNSA warhead activities

https://www.energy.gov/sites/default/files/2022-03/FY%202022%20SSMP%20March%202022.pdf

ADOPTION STAGES: A PROCESS, NOT AN EVENT

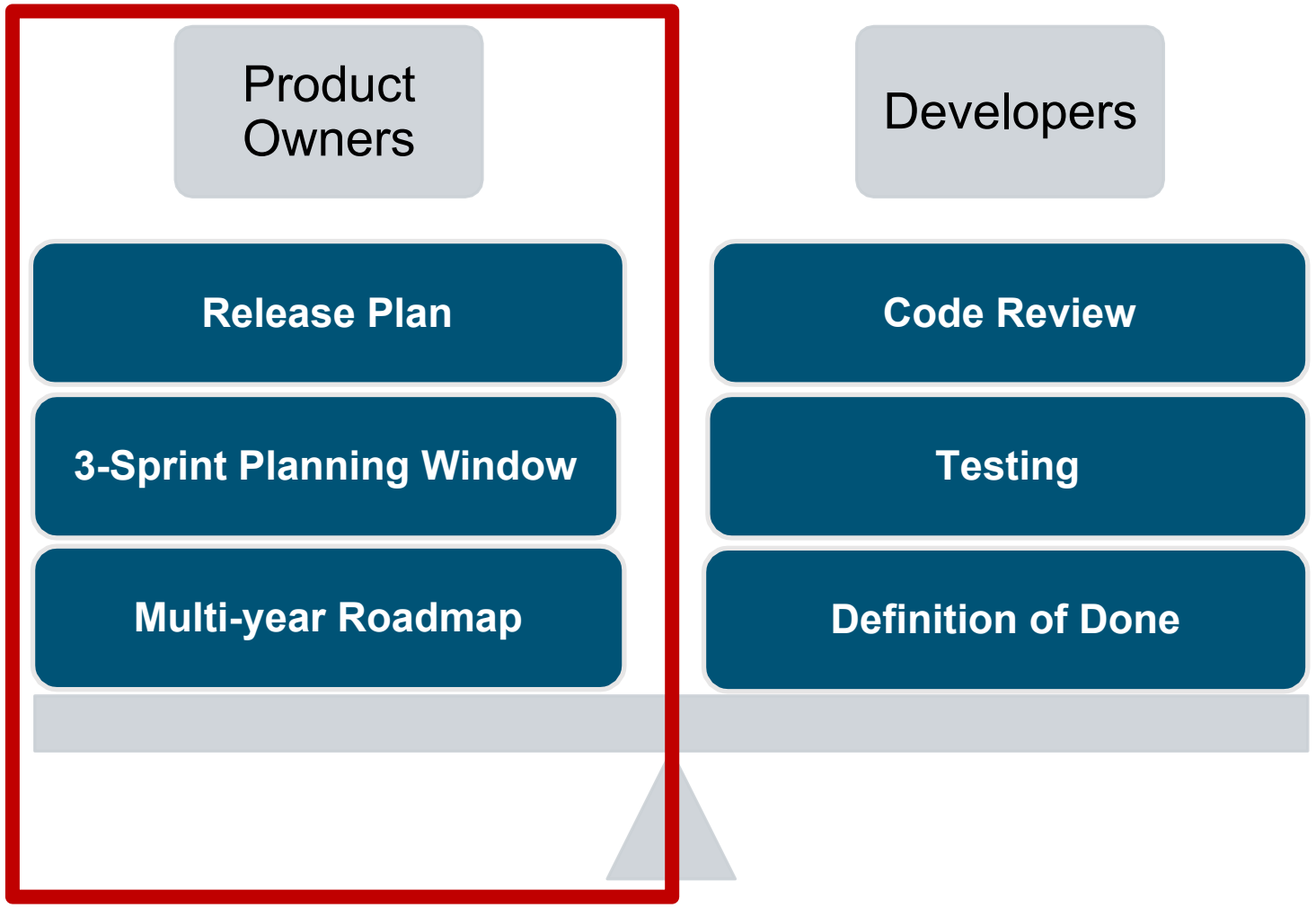
Adapted from: Conner, Daryl R. (1993). Managing at the Speed of Change. (Random House)

IMPROVEMENTS TO PRODUCT OWNER AND DEVELOPER PRACTICES



OUTCOME OF IMPROVED PRODUCT OWNER PRACTICES

Predictable capability delivery enables purposeful conversations and regular outreach with stakeholders. It enables cross-team coordination to deliver capability and encourages a responsive strategy for stewardship.

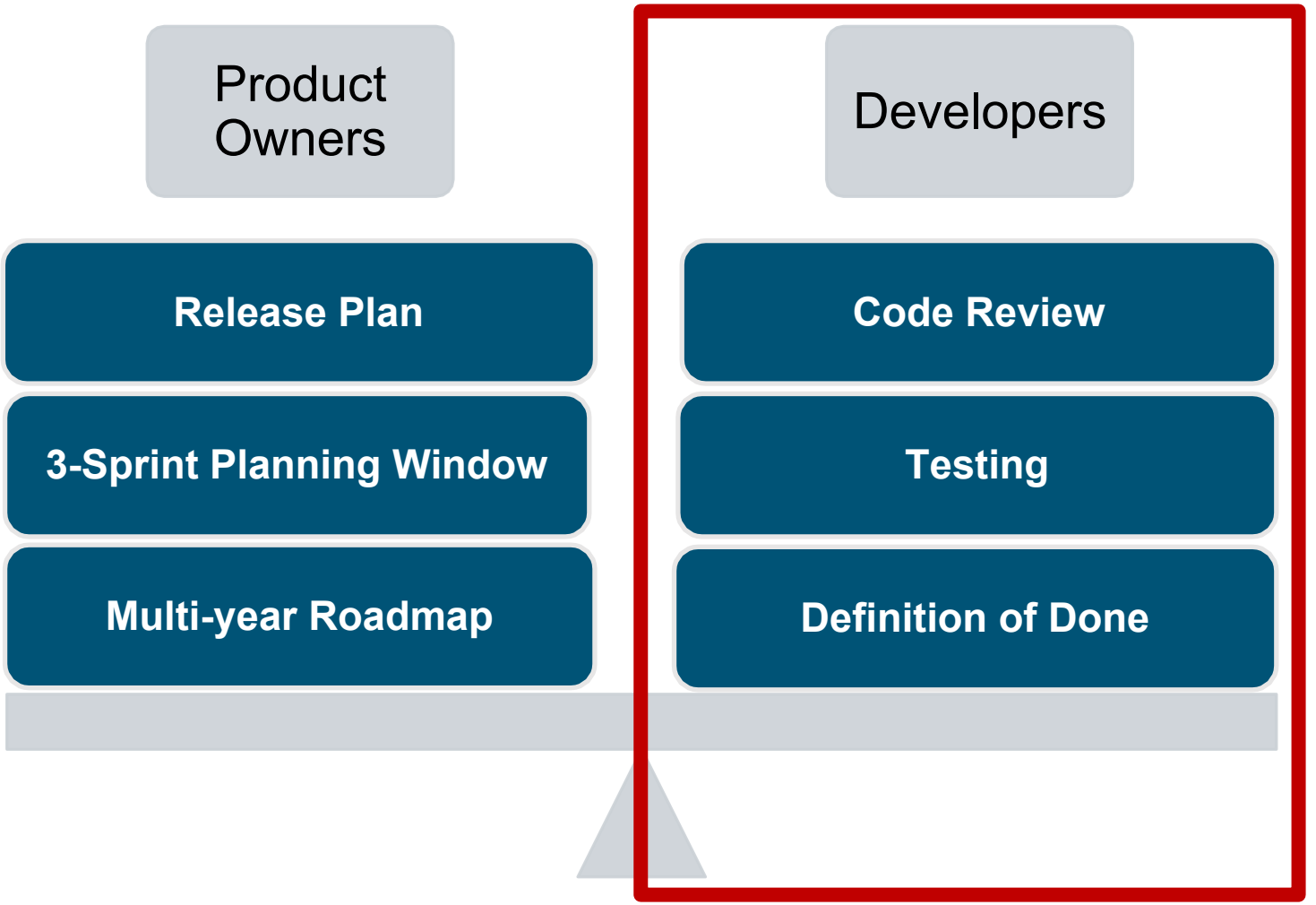


Desired Outcome:

Predictability that enables Commitment

OUTCOME OF IMPROVED DEVELOPER PRACTICES

Sustainable Code is readable, understandable by multiple developers, and can be effectively created or modified. Developers are confident they are not introducing defects in production code; when defects are introduced, they can isolate and fix the problem quickly.



Desired Outcome:

Sustainability that enables Agility

TIMELINE OF EVENTS

2020-09
POLT begins discussions

2020-09 – 2020-12
Created proposal to share with leadership and teams

2020-12 – 2021-02
Presented proposal to leadership and teams

2021-03 – 2021-06
Working groups created plans

2021-07
Working groups plans presented and agreed by leadership

OUTCOMES OF WORKING GROUPS

1540-LEVEL CODE DEVELOPMENT DEFINITION OF DONE

1. Delivered capability meets or exceeds user's defined requirements
2. Modified code builds and passes tests on "production" platforms
3. More than one person has reviewed new or modified code (including the author)
4. New code is accompanied by a test that exercises that code

Individual teams expand and add to these based on their unique needs and circumstances

TESTING

Legacy code:

- When a regression test fails or a user reports a problem, find out why and, **if at all possible,** write a unit test to capture the issue

New code Development:

- **Meets:**
  - New code is accompanied by a test that exercises that code
- **Exceeds:**
  - New code is developed through Test Driven Development (TDD)
  - Any code modified during development is covered by a unit test harness

CODE REVIEW

What code review means for 1540:

Level 1 (Meets):

- The author and at least one other person have reviewed the code
- When you are working in unfamiliar code, one of the reviewers must be on the owning team

Level 2 (Exceeds):

- Code walk-through is performed

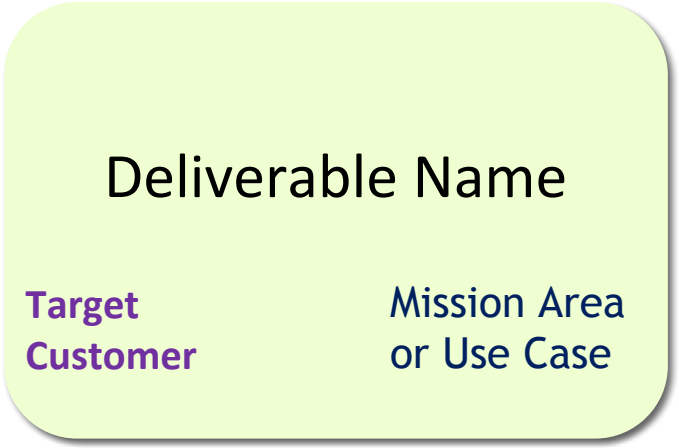
Level 3 (Exceeds):

- Code is developed through pair or group/mob programming

Level 4 (Exceeds):

- Code is checked out locally and a separate developer attempts to break the new code or capability through additional unit testing

TECHNOLOGY ROADMAP LEGEND

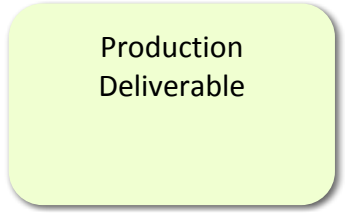


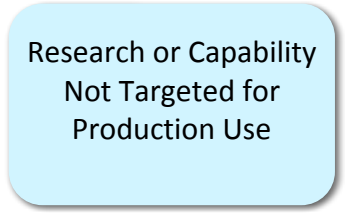
Deliverable: The roadmap lists work at roughly the same granularity of L2 Milestones, L3 Milestones, and Deliverables defined in ASC project plans. Additional tactical, operations and maintenance, or SPP work may also be listed on roadmaps to give a full accounting of work a team is doing to support the ND mission.

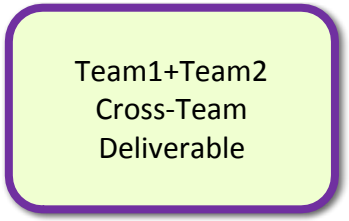
Target Customer: This is the customer, user, or primary stakeholder for which a capability is being implemented. If left blank this indicates a foundational capability supporting many programs.

Mission Area or Use Case: This is the ND relevant simulation use case, initiative, or foundational capability that the deliverable supports.

Deliverable Types







TEAMS DEFINITION OF DONE

Sierra Toolkit (STK)

Exploratory:
- New story(s) is created or team has agreed not to create a new story
- The entire team is aware of the path forward

User Support (Debugging):
- STK unit tests have been put into place to cover bug
- The issue has been resolved
- STK tests passing
- No new failures in app tests
- The code is pushed
- If this was in found in Trilinos, it is pushed to Trilinos
- The user has been notified that the issue is resolved

Optimization:
- The optimization was implemented
- The performance improvement was evaluated
- STK tests passing
- No new failures in app tests
- If the performance benefit is good enough, the code is pushed

New Feature:
- STK unit tests cover all new code
- STK tests passing
- No new failures in app tests
- Code is pushed
- If this came from an external user, it is pushed to Trilinos

Technical Debt:
- STK tests passing
- No new failures in app tests
- The team agrees that overall technical debt is reduced
- Code is pushed

Plato

Proof-of-Concept
- Code is tested
- Tests each run less than 10 seconds

Beta
- Regression/integration tests have been added
- Support added to XML Generator
- Input reference documentation is added

Production
- V&V tests have been added
- One or more real user problem works with the capability
- Support added to Plato/SAW integration
- Relevant demo added to /projects/plato
- Tutorial video has been added

Thermal/Fluid

Has the acceptance criteria been met?

Have appropriate unit tests been added? A unit test or small regression test should cover every new feature.

Have the documentation and release notes been updated? Have the appropriate equations been added to the documentation?

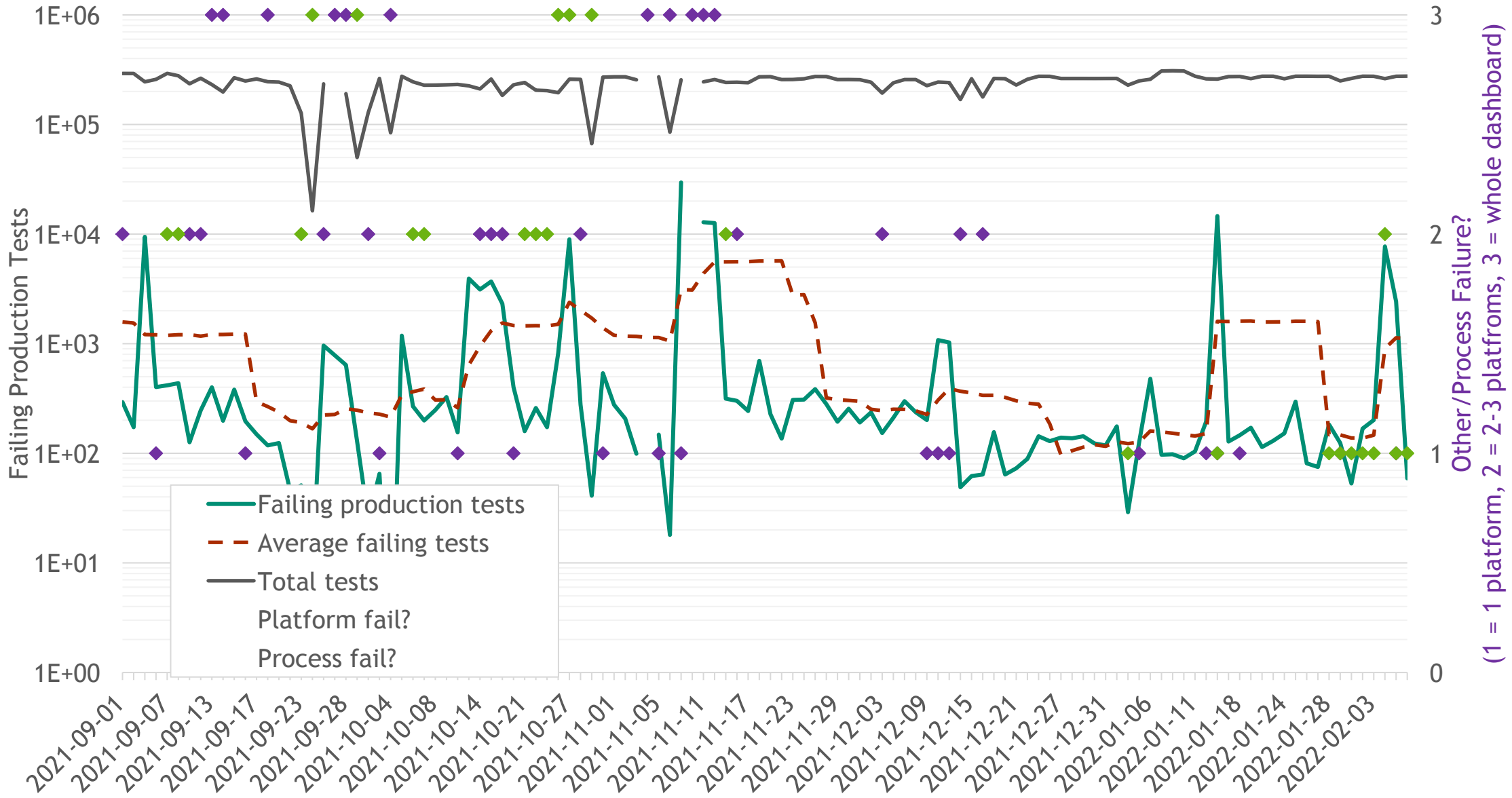
Has the modified code/test been reviewed by a member of the development team?

Does modified code build and pass tests on "production" platforms? Will all lines on the dashboard remain "green" or will dashboard lines not have any new failures?

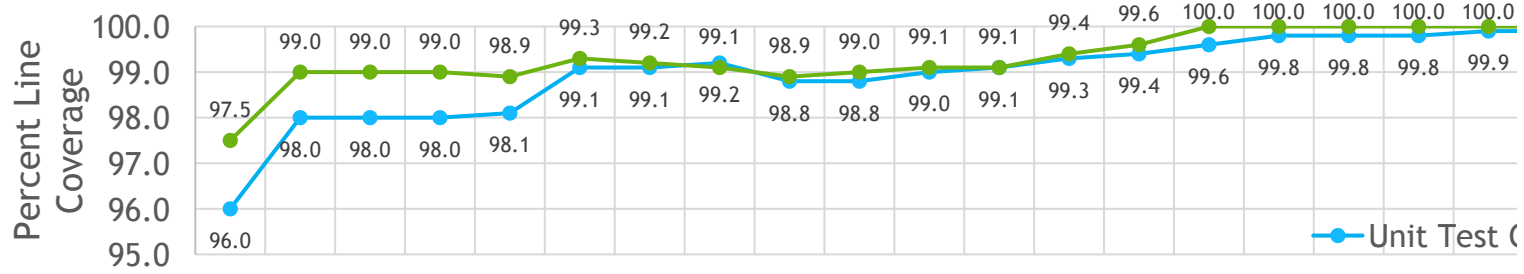
Do we have done slides for the sprint review?

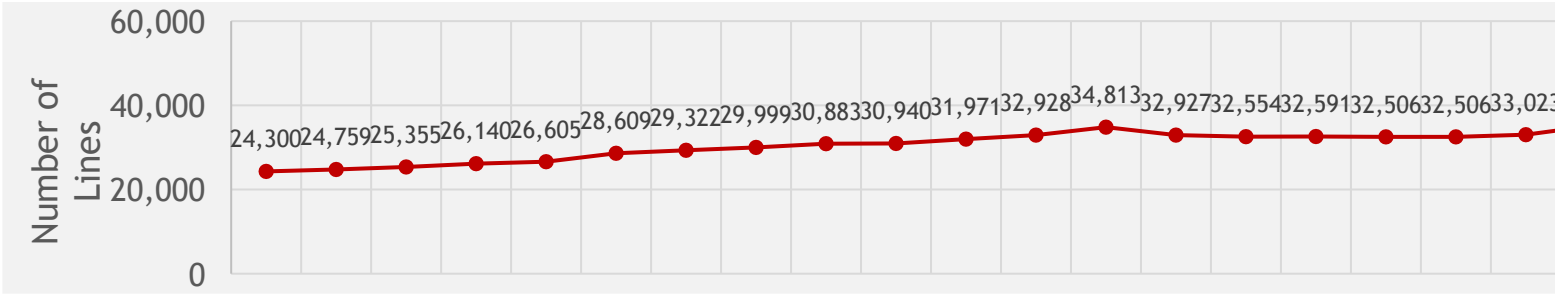
Does the story/epic have a delivered capability and a clearly defined customer?
- If so, ask the user for verbal assurance that the capability is working as expected. Ask the user for a test of this capability and work with the user to add it the nightly testing process.

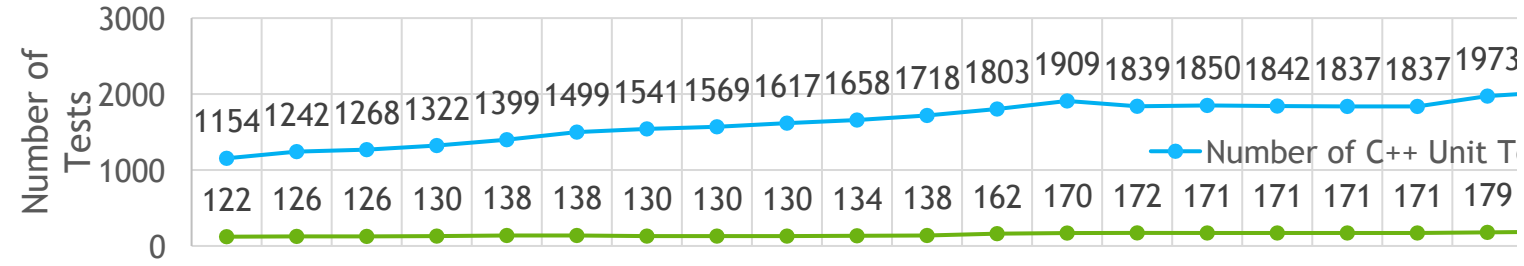
NIGHTLY DASHBOARD TEST FAILURES

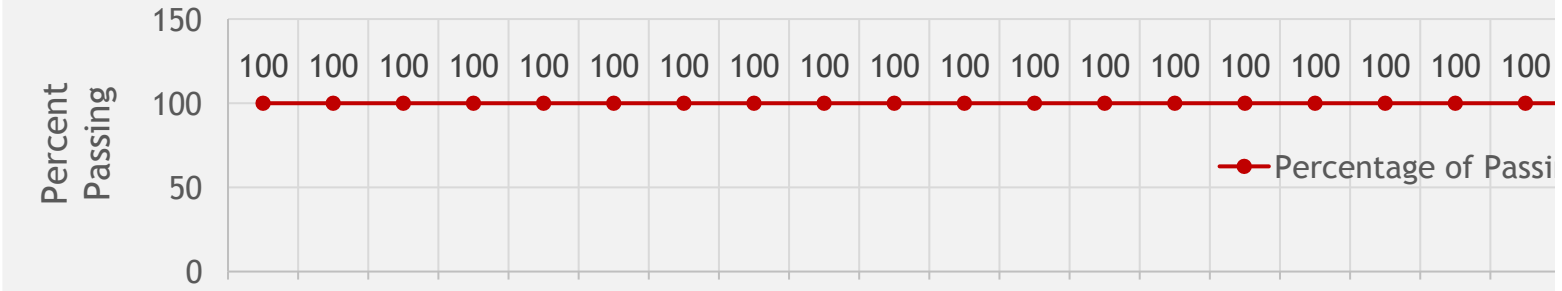


TEST COVERAGE









unit test suite coverage

Fluid/Thermal Mechanics Applications

aero	30.3%
aria	38.7%
chaparral	56.6%
fuego	43.4%
krino	58.9%
nalu	17.0%
particles	53.6%
tftk	87.5%

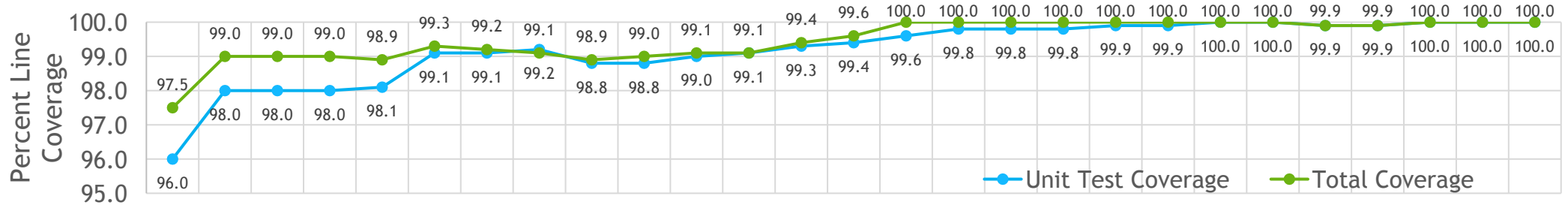
Solid Mechanics/Dynamics Applications

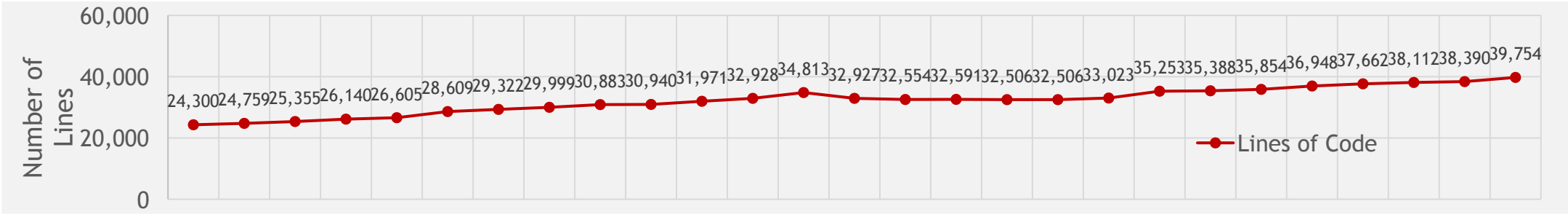
FETI-DP	99.9%
kestrel_interface	0.0%
MATHLIB	48.6%
Salinas	54.9%
adagio	29.7%
apublic	40.3%
contact	37.7%
eagle	38.3%
geometry_toolkit	94.1%
its2sierra	82.5%
lame	34.9%
math_toolkit	99.6%
mesh_toolkit	92.0%
nemoUtils	95.1%
plato	10.2%
smtk	33.8%
zapotec	82.9%

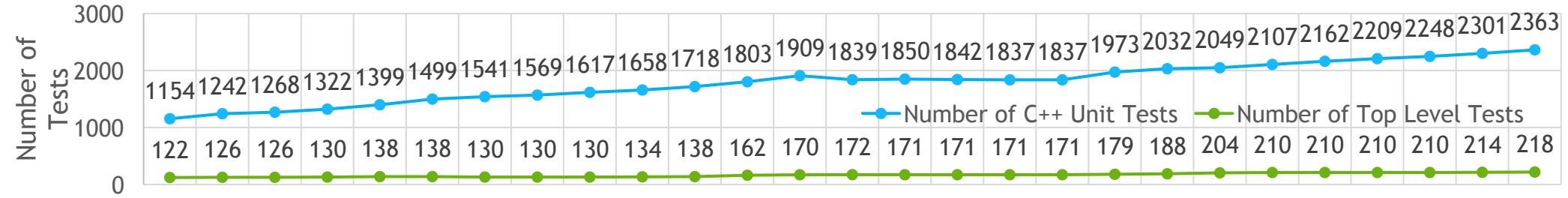
Verification and Validation Tools

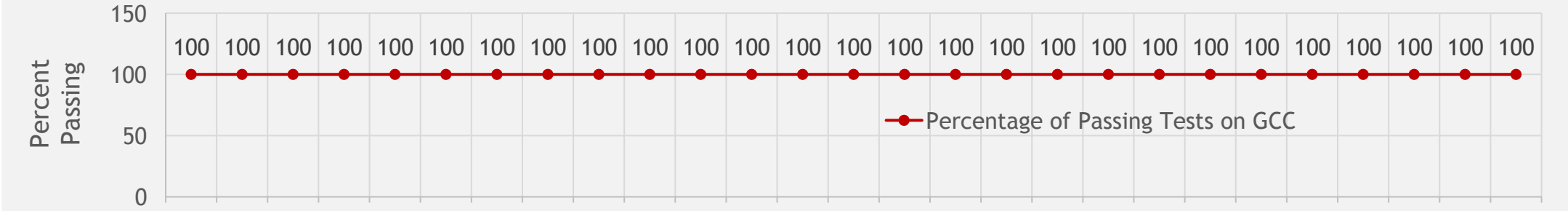
cubit	10.1%
encore	16.7%
mesh_scale	66.4%
percept	61.3%

NGS SQE METRICS (REPORTED BY SIERRA DEVOPS)



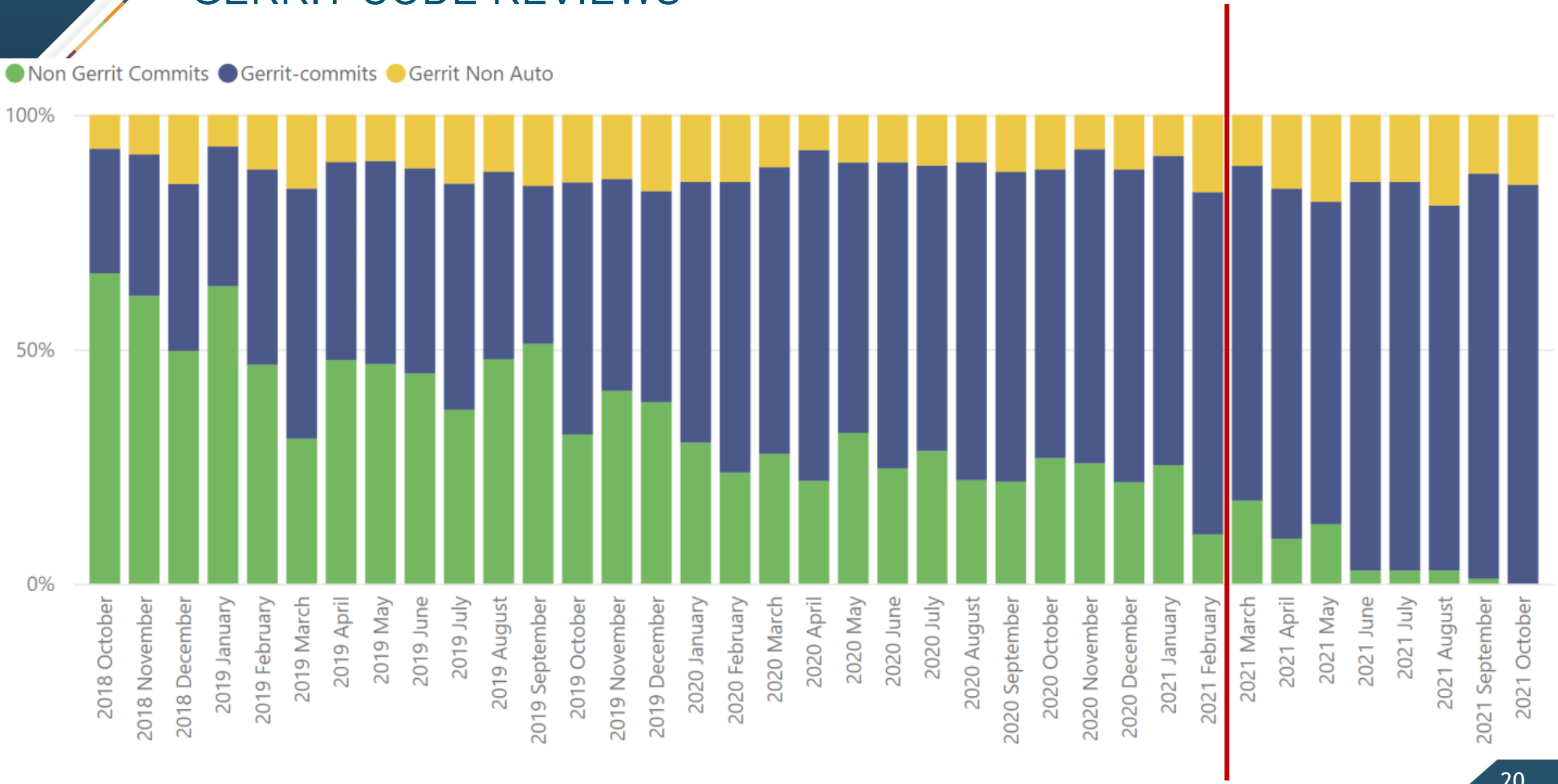






COMPSIM
NEXT GEN SIMULATION

GERRIT CODE REVIEWS



ROADMAPS

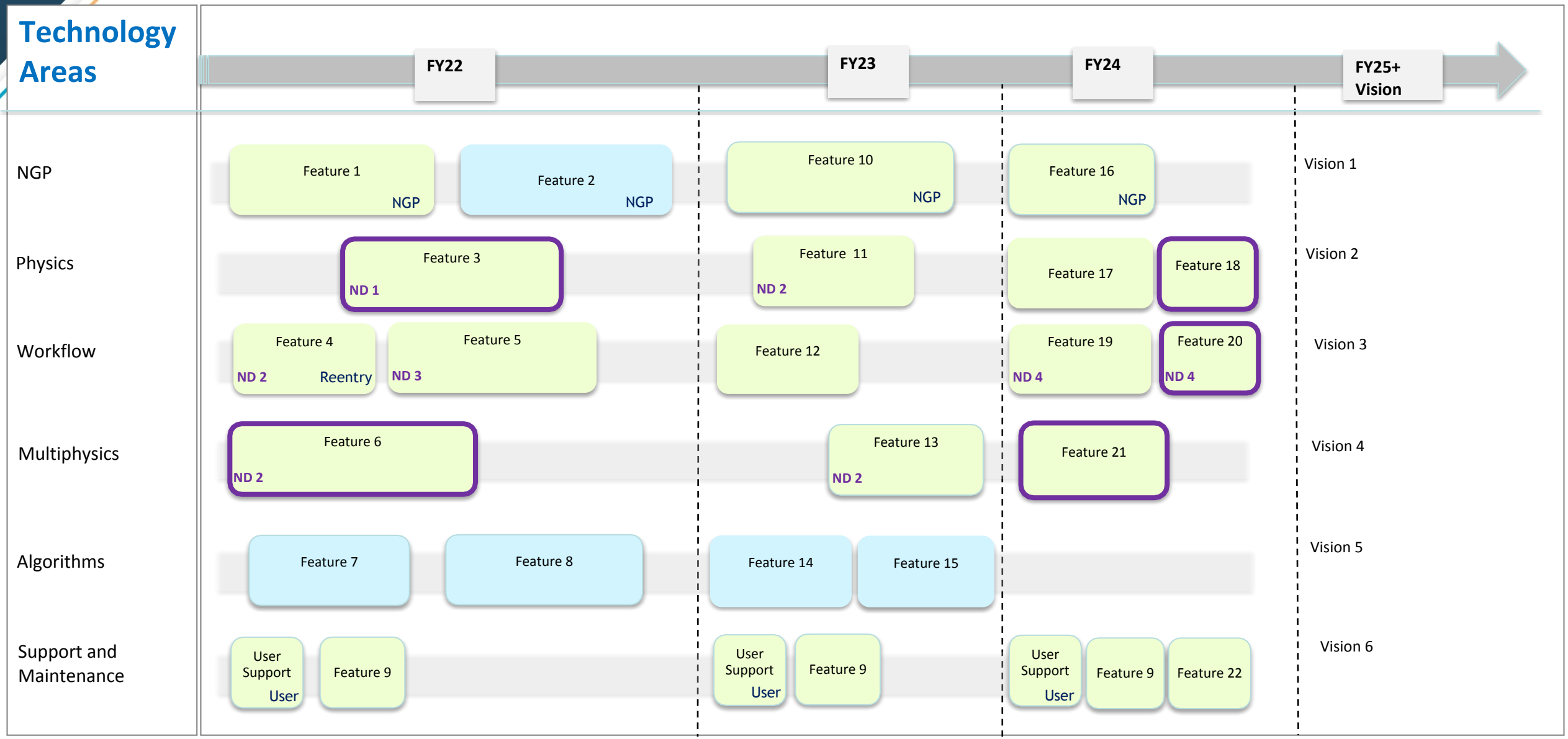
One place

Access for stakeholders

Common look and feel

TECHNOLOGY ROADMAP

POC: PO_NAME
Updated MMM YYYY



Related ASC Project Plans: “Plan 1” “Plan 2”

CONCLUSION

Change is hard

Change is necessary even after a decade of good practices

THANK YOU

ADDITIONAL MATERIAL

ABSTRACT

The software development teams that provide the ASC Sierra software suite have operated in an Agile manner for over a decade. Despite their maturity in this domain, product leadership felt the teams needed to make changes to increase transparency, responsiveness, and value delivery. Through a series of presentations to leadership, developer teams, and special working groups, a series of improvements were proposed and enacted on development teams. In this talk, we will explore these process improvements which fall under two main themes: predictable product planning and sustainable code development.

To find improvements to predictable product planning, a working group of Product Owners formed and discussed methods to better enable purposeful discussions with stakeholders, coordinate cross-team work, and encourage a responsive strategy for stewardship through near-term sprint plans, release plans, and multi-year roadmaps. They then proposed standardized templates for these artifacts to achieve these goals.

Similarly, a working group of developers formed to discuss methods for increasing the sustainability of the software. They proposed improvements in the following three areas: code review, testing, and an organization-wide Definition of Done for software delivery.

We will explore the proposals that were made by both working groups and the impact these recommendations had on the organization. We will close by exploring the need for continual inspection and adaptation to answer the challenges software teams face as they deliver value to their customers and stakeholders.

1540-LEVEL CODE DEVELOPMENT DEFINITION OF DONE

1. Delivered capability meets or exceeds user's defined requirements
   - **Meets:** Customer gives verbal assurance capability is working as expected
   - **Exceeds:** Customer provides acceptance test(s) that must pass upon completion
     - "User acceptance test (UAT) criteria (in agile software development) are usually created by business customers and expressed in a business domain language. These are high-level tests to verify the completeness of a user story or stories 'played' during any sprint/iteration."
2. Modified code builds and passes tests on "production" platforms
   - **Meets:** Each team determines what qualifies as "passing tests"
   - **Exceeds:** All lines on the dashboard are "green"
3. More than one person has reviewed new or modified code (including the author)
   - **Meets:** Code is reviewed after development by a member of the product team
   - **Exceeds:** Code was developed through mob or pair programming
4. New code is accompanied by a test that exercises that code
   - **Meets:** All new code is covered through testing
   - **Exceeds:** All new code is 100% covered by unit tests

Individual teams expand and add to these based on their unique needs and circumstances

TIMELINE OF EVENTS

Create a timeline and make it less wordy

2020-09: Begin discussing with Martin Heinstein the need for practice improvements for Product Owners and developers
2020-09-22: Begin meeting with Martin Heinstein and Salomé Thorson to discuss the format of the initiative
2020-09 - 2020-12: Met many times with Martin, Salomé, Charis, and Mike to refine the approach, proposal, and presentation
2020-12-14: Met with managers and presented the proposal to them to obtain their support for moving forward
2021-01-07: presented proposal to product owners and scrum masters and got feedback from them and modified the approach
2021-01: worked with several people to modify the approach and path forward
2021-02-09 - 2021-02-22: met with each product team and presented the proposal in concert with Martin Heinstein and asked for volunteers for the developer practice improvements working group as well as a special retrospective for each team on the proposed practices
2021-03-04: helped to kick off the PO practice improvements working group. I did not attend most of their working group meetings but did support them in their exploration of roadmapping tools.
2021-03-15: facilitated the SPARC retrospective on proposed practices
2021-03-22: worked with all scrum masters to analyze data from the team retrospective and designed a survey to gather data from individuals as well
2021-03-25: facilitated the developer practice improvements working group.
2021-03-29 - 2021-06-25: set up and facilitated developer practice improvements working group meeting and helped them create a proposal to share with all teams
2021-07-08: supported PO and Developer practice improvements working groups as they presented their proposals with the entire group
2021-07-13: worked with scrum masters to influence teams to enact proposals from the working groups

2021-07-15: included proposals in FY22 planning to help teams decide what to do moving forward