SAND2022-6997C

Sandia National Laboratories

Exceptional service in the national interest

MOVING TO QUARTERLY RELEASES FOR THE ASC SIERRA CODE SUITE

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

Agile Improvements Working Group

ACKNOWLEDGMENTS

1540 Agile Improvements Working Group

- Paul Crozier
- Todd Coffey
- Ryan Shaw
- Riley Wilson
- Mario LoPrinzi
- Julia Plews
- Johnathan Vo
- George Orient
- Rich Drake
- Ryan Viertel
- Corey Ernst
- Tricia Schmitt

CompSim Product Owner Leadership Team

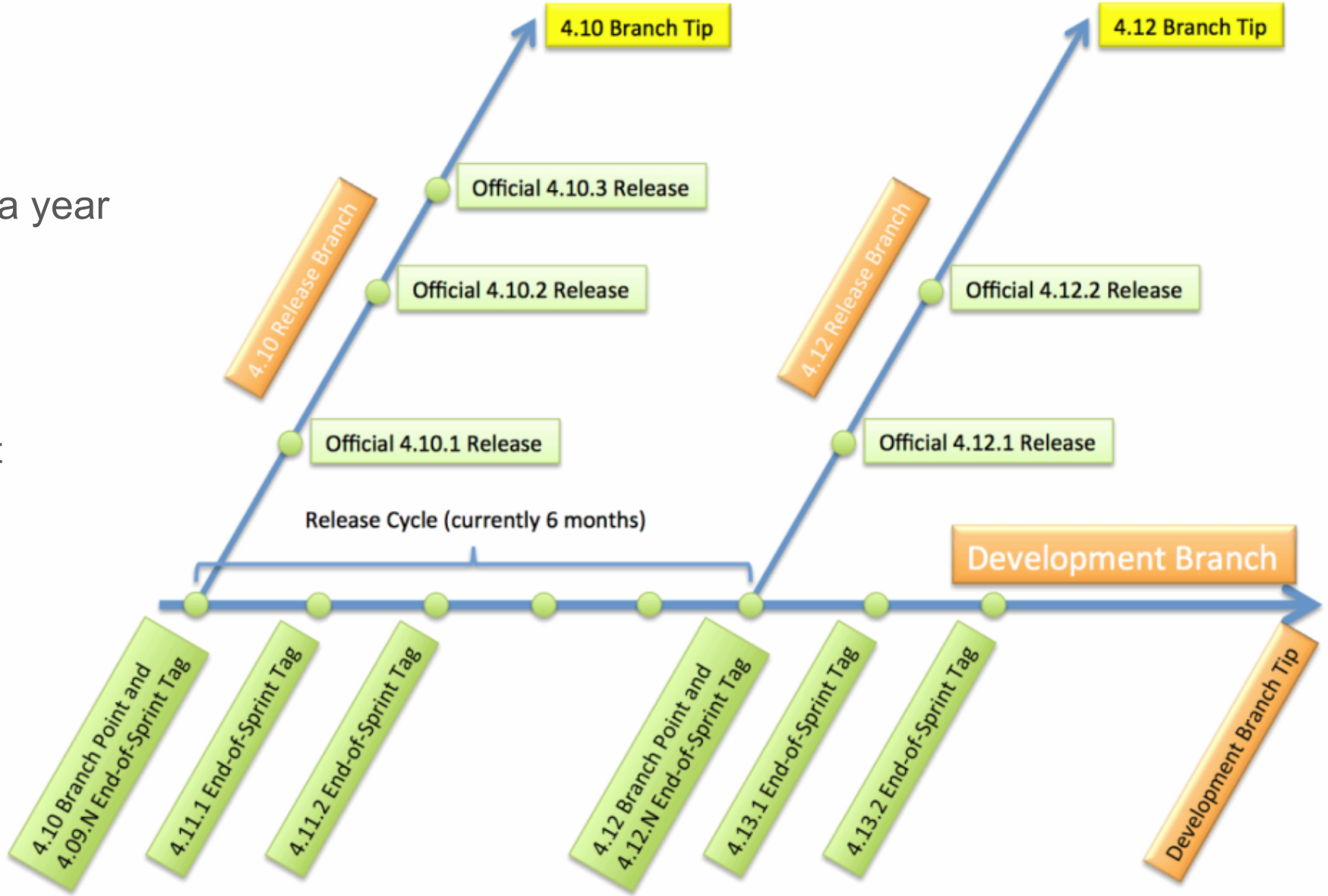
- Martin Heinstein
- Nate Crane
- Mike Glass
- Charis Church

SIERRA RELEASE HISTORY

2007 – 2020: Release twice a year

Support of four releases

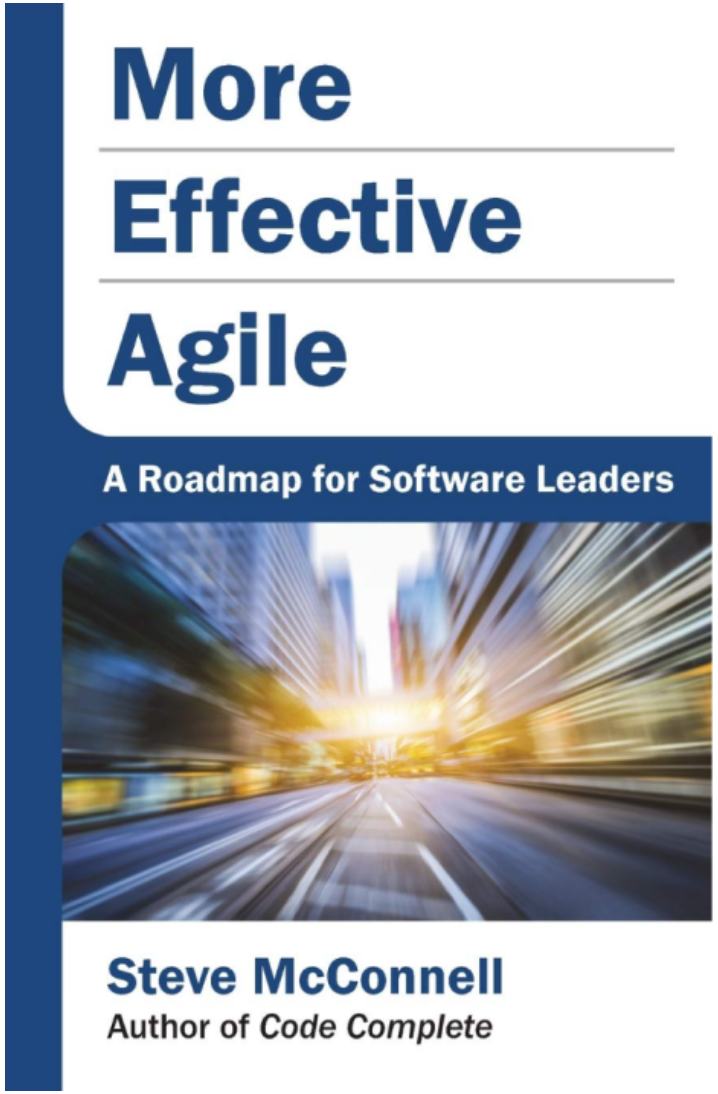
- Two with critical fixes
- Two on maintenance support



1540 AGILE IMPROVEMENTS WORKING GROUP

Principles from the book:

- Tighten feedback loops
- Maintain a releasable level of quality



"If it hurts, do it more frequently, and bring the pain forward."

-- Jez Humble[1]

1. Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation

PROPOSED EXPERIMENT

The Agile Improvements Working Group (AIWG) proposes increasing CompSim's release cadence from 2 times per year to 4 times per year.

This will require releasing Sierra every 3 months for 1 year.

Proposed Benefits:

1. Identify release processes that should be improved
2. Motivate teams to consistently maintain a releasable level of quality
3. More frequent releases to customers
4. Identify developer workflows and tools that should be improved

STAKEHOLDER INTERVIEWS

15 analysts

8 programs (ND & DoD)

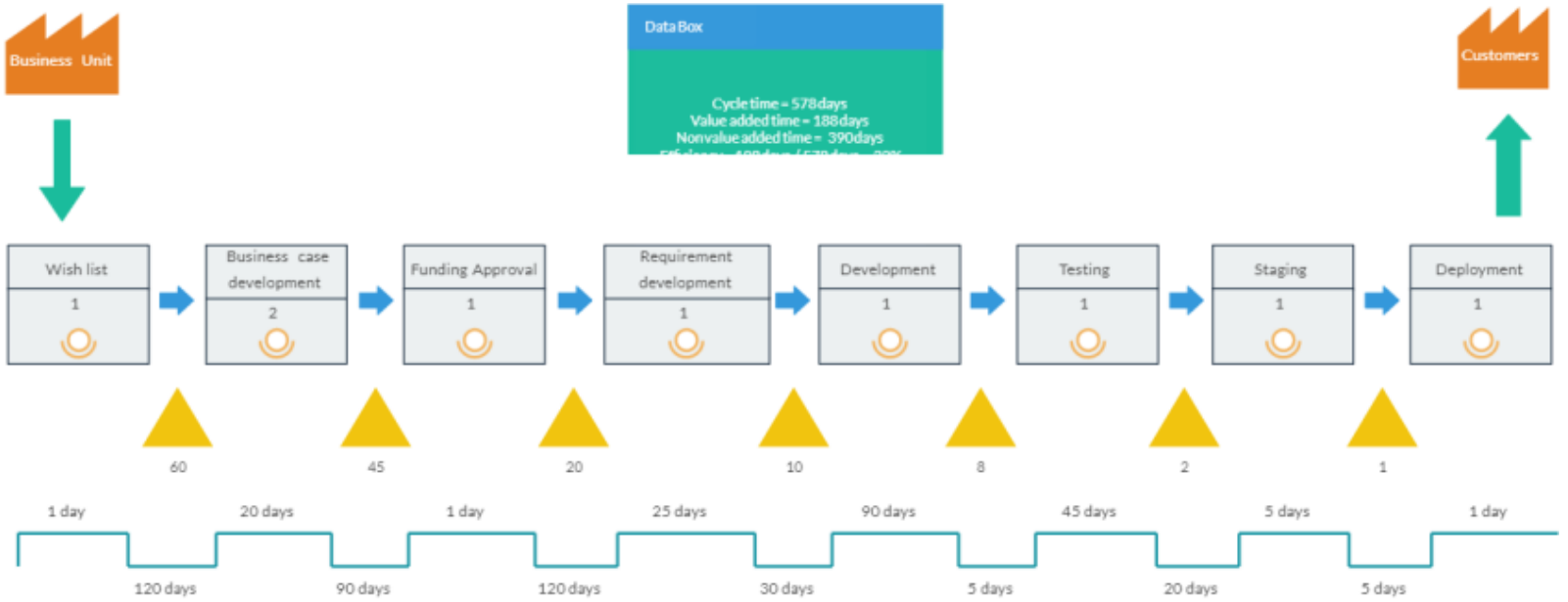
Highlights

- More opportunities to use new features in a pedigreed version
- More frequent acceptance testing → higher quality
- Impacts on long-term analyses

VALUE STREAM MAPPING

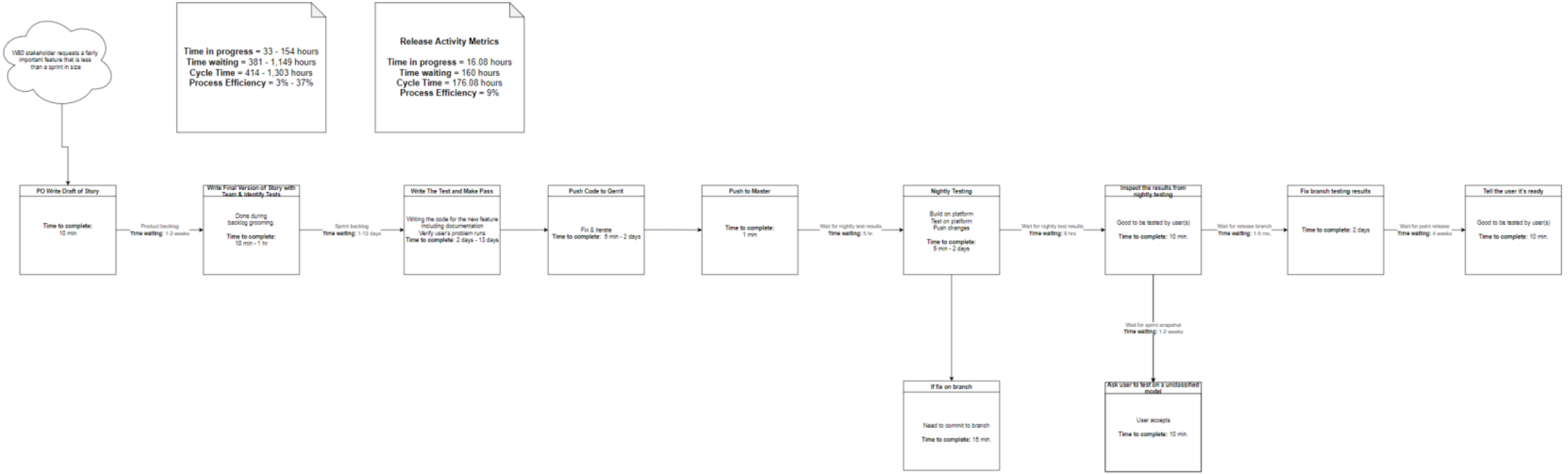
"Value stream mapping is a Lean management method that allows you to visualize, analyze and improve all the steps in a product delivery process." 1

Hypothesis: determine issues with release preparation so CompSim could address them and decrease the amount of time it took to release

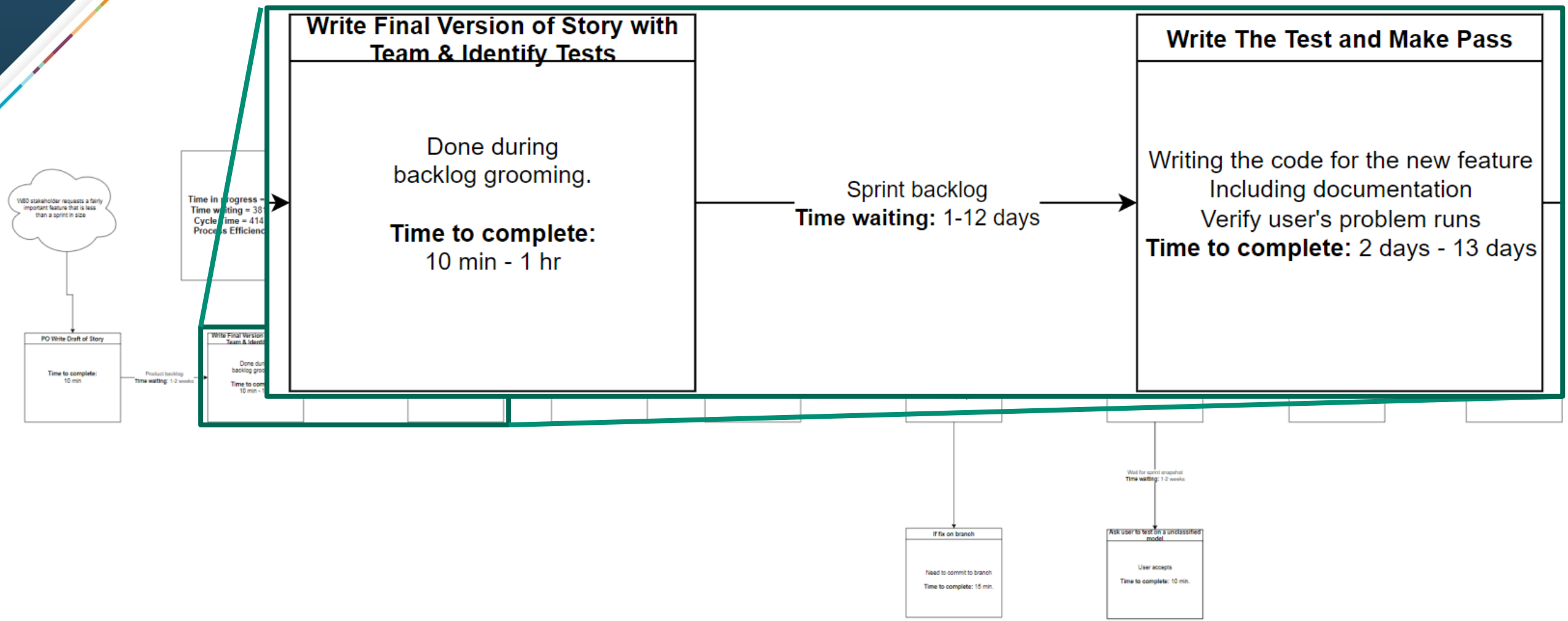


1 https://kanbanize.com/lean-management/value-waste/value-stream-mapping

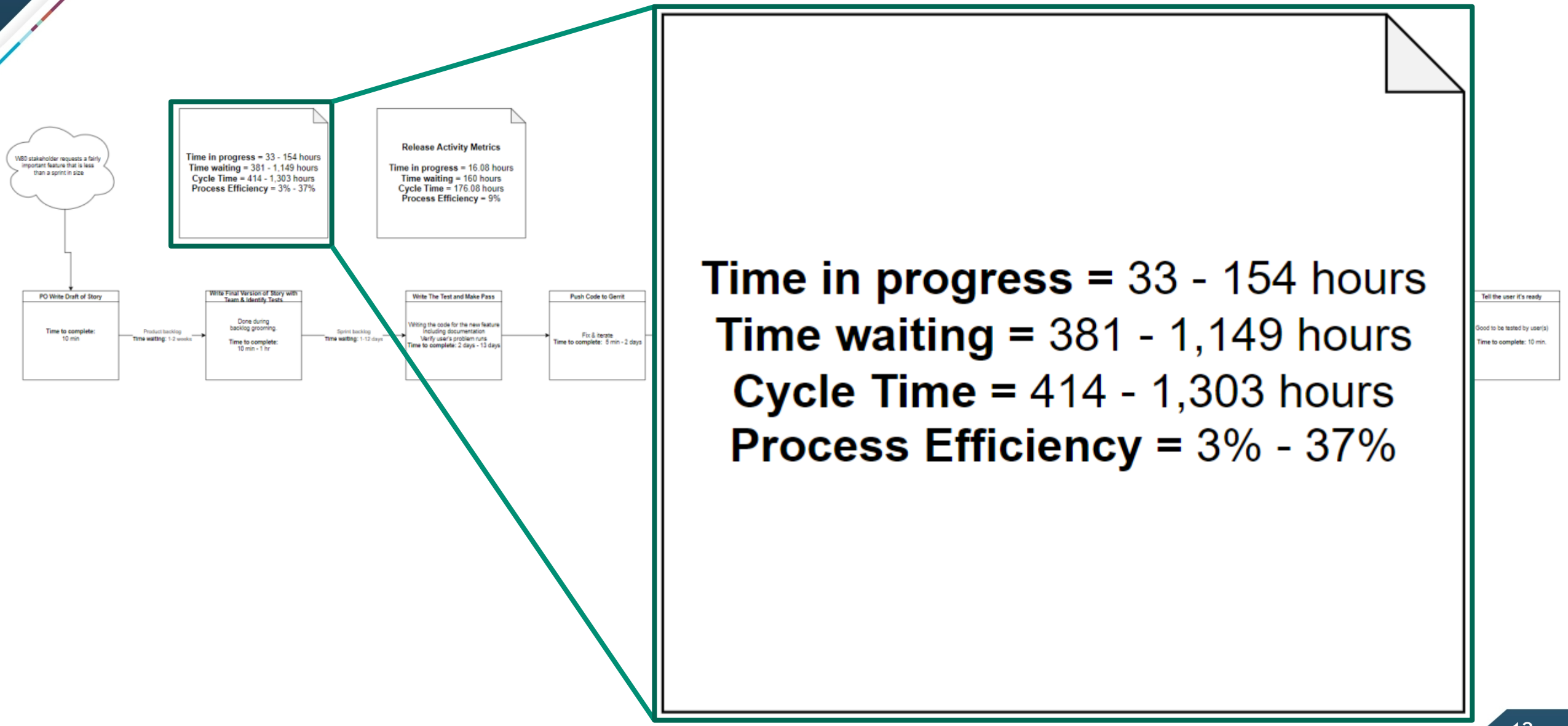
VALUE STREAM MAP



VALUE STREAM MAP



VALUE STREAM MAP



Time in progress = 33 - 154 hours
Time waiting = 381 - 1,149 hours
Cycle Time = 414 - 1,303 hours
Process Efficiency = 3% - 37%

COMPSIM VALUE STREAM MAPS

Team	Time in progress (days)	Total Time (days)	Process Efficiency	Release Activities: Time in Progress (days)	Release Activities: Total Time (days)	Release: Process Efficiency
DevOps	6	71 - 171	3 - 8%	0.5	15.6	3%
G&M	14	72 - 152	10 - 19%	4.5	21.8	21%
Inverse	20 - 25	64 - 199	10 - 38%	4.5	21.8	21%
NGS	9	45 - 165	5 - 17%	4.4	16	22%
Plato	11	57 - 157	7 - 20%	6.5	31.3	21%
SD	25 - 45	30 - 220	17 - 88%	10	37.5	27%
SM	19	72 - 162	12 - 26%	10.4	46.4	22%
SPARC	7	25 - 49	14 - 27%	2.1	2.6	80%
STK	4 - 19	52 - 163	3 - 37%	2	22	9%
T/F	15	58 - 195	7 - 26%	0.1	26.1	0.50%
Average	**13 - 30**	**55 - 163**	**9 - 31%**	**4.5**	**24**	**23%**

OUTCOMES

BRINGING THE PAIN FORWARD

Release processes

Documentation

Definition of Done

NEW VALUE STREAM MAP

Real data from the 5.6 release

Time in progress = 97 hrs
Time waiting = 160 - 560 hrs
Cycle Time = 257 - 657 hrs
Process Efficiency = 15% - 38%

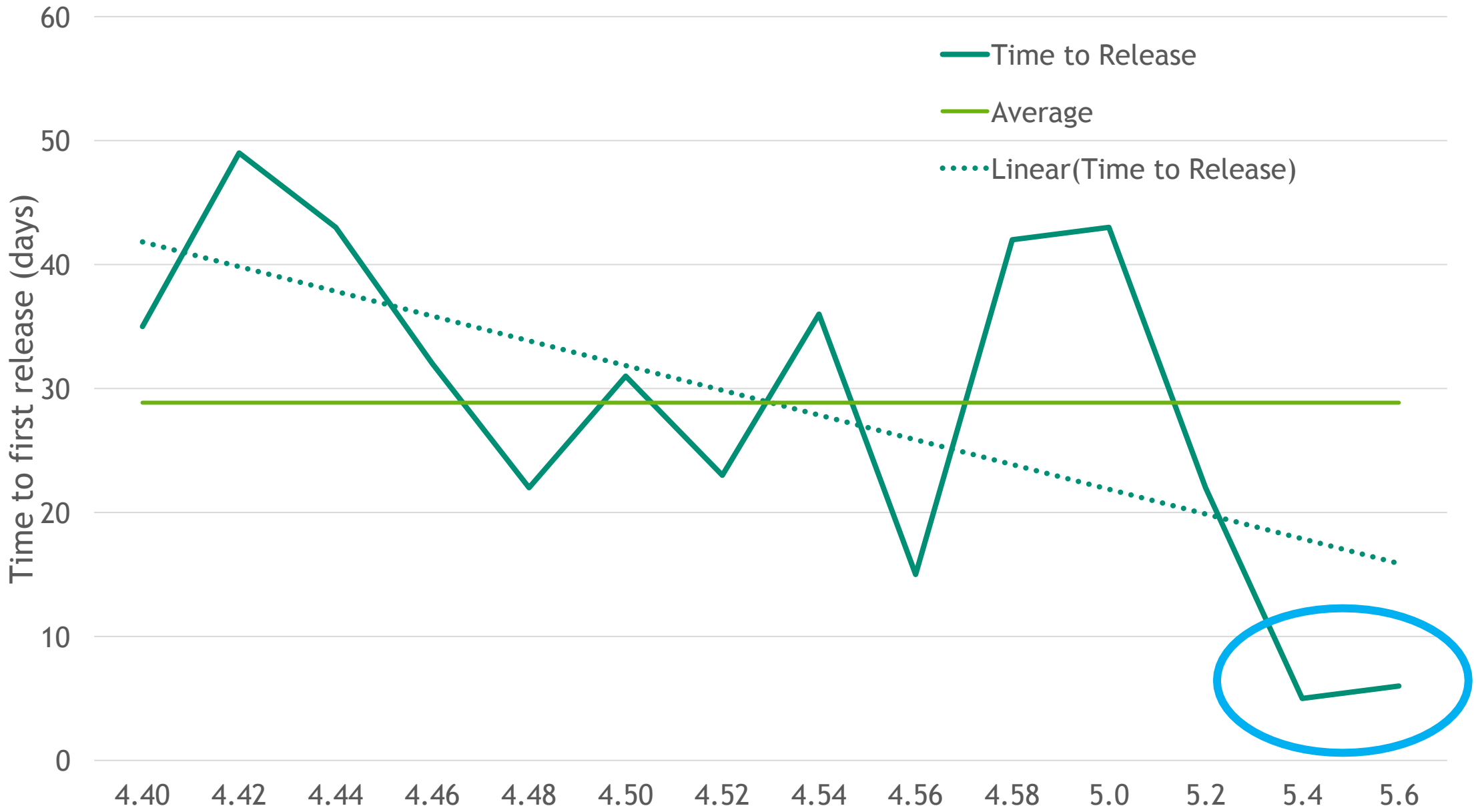
Old: 9% - 31%

Release Activity Metrics

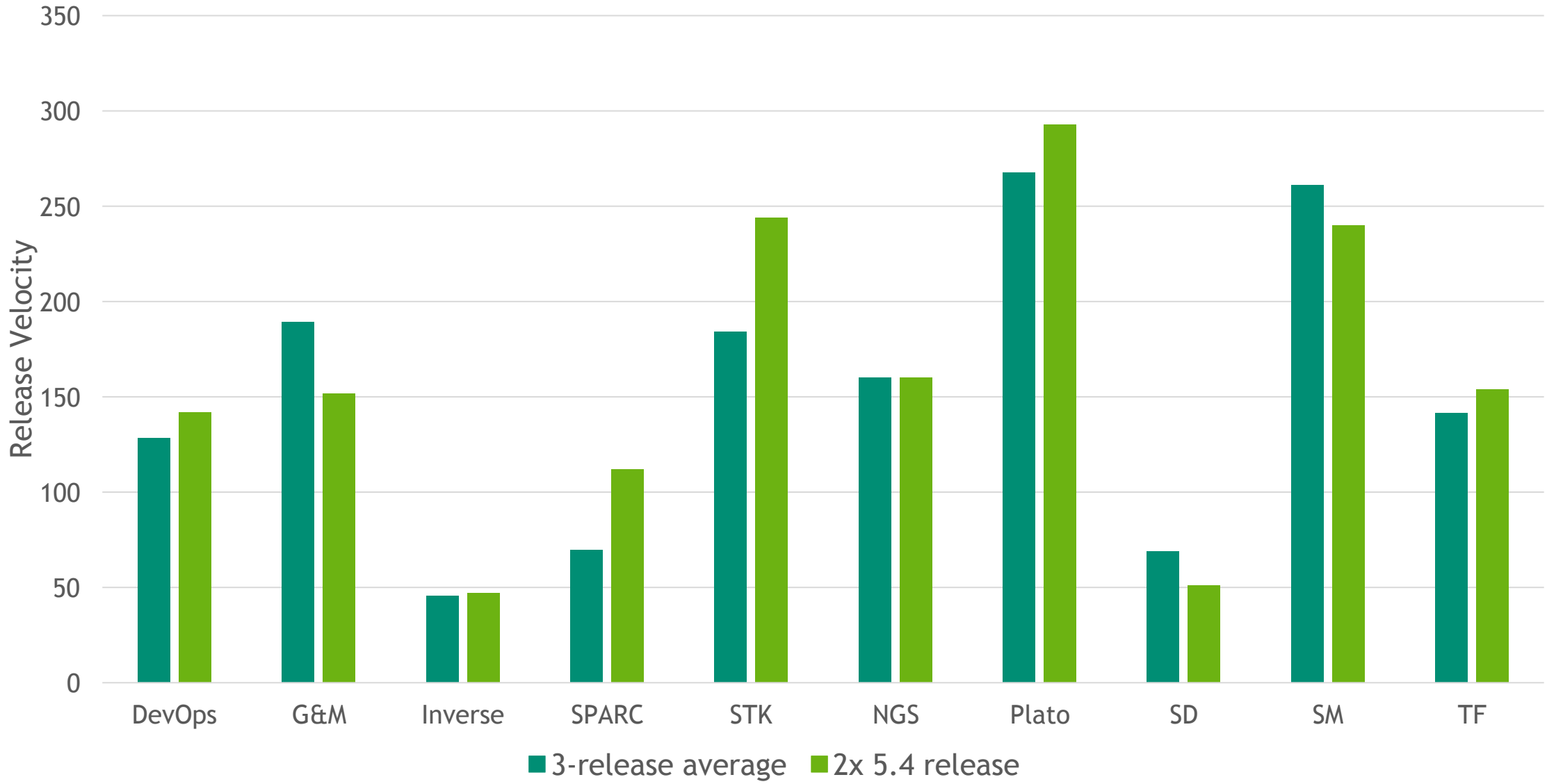
Time in progress = 82.6 hours
Time waiting = 184 hours
Cycle Time = 266.6 hours
Process Efficiency = 31%

Old: 23%

TIME TO RELEASE



RELEASE VELOCITY PER TEAM



LESSONS LEARNED

Documentation

Security policies

Large acceptance testing

3 week sprints

Moving to 3 releases per year

CONCLUSION

Bring forward the pain to spur improvements

QUESTIONS?

ADDITIONAL MATERIAL

ABSTRACT

For over a decade, the ASC software suite Sierra has provided official releases to their customers once every six months. In this talk, we will explore what it took for the Sierra development teams to move from biannual to quarterly releases.

The push to quarterly releases began when the Agile Improvements Working Group (AIWG) was created in August 2020. After evaluating several possible group-wide improvements, the AIWG decided to propose a move to quarterly releases. They presented this proposal to management and product leadership with the realization that the cost to release Sierra was too high for quarterly releases.

To understand the cost of releasing Sierra, the Product Owner Leadership Team (POLT) led a value stream mapping exercise with each Sierra development team. We will discuss how this exercise was accomplished and the results of the value stream maps. It was determined that the largest delay in value delivery stemmed from the biannual release cycle.

Key customers and stakeholders were interviewed to understand their reactions to doubling the frequency of release. Feedback was overall positive, and we will discuss some of the customers' positive and negative responses.

During FY22 planning, this proposal was accepted as an experiment. We will describe the changes teams made to decrease the cost of release, including improved dashboard accountability and automated release deployment. We will also show data from newer value stream maps and Sierra's three-fold decrease in time to release.

PROPOSED EXPERIMENT

The Agile Improvements Working Group (AIWG) proposes increasing CompSim's release cadence from 2 times per year to 4 times per year.

This will require releasing Sierra every 3 months for 1 year.

POTENTIAL BENEFITS

1. **Identify release processes that should be improved:** any pain-points identified by doing more frequent releases are good candidates for focusing on process improvement. A release should not require a lot of manual effort, and any manual effort is easier to accept when releases are less frequent.

2. **Motivate teams to consistently maintain a releasable level of quality**: software quality should be an ongoing practice, not a separate effort in preparation for a release. Any efforts a team can make to be ready to release at any time is a good investment.

3. **More frequent releases to customers**: delivering new capabilities to customers more frequently makes the software more useful to customers and allows for tighter feedback loops with customers who use only released software. One good example is the push for shorter LEP lifecycles, where analysts often need new capabilities in their hands quickly, yet they are required to use only officially released software. In a two-year LEP, a 6-month delay spent waiting for new CompSim capability to be released would be simply unacceptable to analysts.

4. **Identify developer workflows and tools that should be improved:** inherently tied to developer/team processes are developer tools necessary for delivering releasable products at a more frequent interval, such as unit, regression, and performance testing, access to computing platforms, and code review/CI automation tools. If any developer tools are identified as “pain points” on a faster release schedule, these tools present an opportunity to improve, simplify, and streamline developer workflows.

WHAT'S THIS GOING TO COST?

Amortized Costs

- Updating documentation, for example.
- The cost of documenting 3 months of capabilities every 3 months should be approximately the same as documenting 6 months of capabilities every 6 months.

Repeated Costs

- Activities where the same manual effort is required no matter how frequent the release
- These cost more in the long run and should be automated as much as possible

REALLY, WHAT'S THIS GOING TO COST?

DevOps Team:

- Some manual release processes that have had incremental improvements over time
  - Until full automation is achieved, these processes will be a repeated cost for every release
  - Added cost during this experiment, potentially delaying other DevOps priorities

Other teams:

- Updating documentation is a common amortized cost
  - Encourage on-the-fly documentation of features/changes
- Running large acceptance tests is a repeated cost
  - Can we get the same answers from these tests more efficiently?
  - Can we afford to wait 6 months between runs to get this feedback?
- Performance, regression, and unit test results must be reporting regularly & reliably across testing platforms
  - Will more frequent releases lead to a need for team-integrated DevOps?

EVALUATION

At the end of the year (after 4th release), the experiment will be evaluated by **POs, SMs, team members, select analyst partners, and management**. Using this input, the **Senior Manager** will make the final decision on release frequency.

Specific attributes that should be evaluated are:

1. The costs to each team at each of the 4 releases.
2. Any extra effort required to release more often should be evaluated for ways to alleviate the cost or incorporate the cost as part of ongoing efforts instead of dedicated efforts at release. (Improving processes is the primary value of this experiment)
3. Any additional benefits of releasing more frequently that are identified by any stakeholder
4. Are there additional costs to analysts in managing more release versions?
5. What's it going to cost if we don't improve our processes to be more agile?