



“Our journey into the DevOps DOJO world” - Story by Sam’s Club

DEPLOY MORE , SLEEP BETTER

By Sam’s Club Team , Sept ,2020






**You can be brilliant and still go out of
business if you're too inefficient.**

- Sam Walton





Agenda

1. Sam's Adoption towards DevOps
2. How did we start
3. Success Stories
4. DevOps DOJO - Pilot

Adopting DevOps Practices

Adopting DevOps practices can be bottom up or top down or some combination of both. However unless there is some buy in from management and alignment with the strategic goals of the business, the development and operations team will continue to work in a disjointed fashion.

This session describes a number of strategies and approaches of implementing DevOps practices aligned to Sam's business goals.

Why DevOps is Important for Sam's ?

- Agile software development is increasing
- We are migrating to the Azure cloud
- More reliable releases
- Improved productivity and efficiency
- Shorter Time to Market
- Better product quality
- Support DDR model

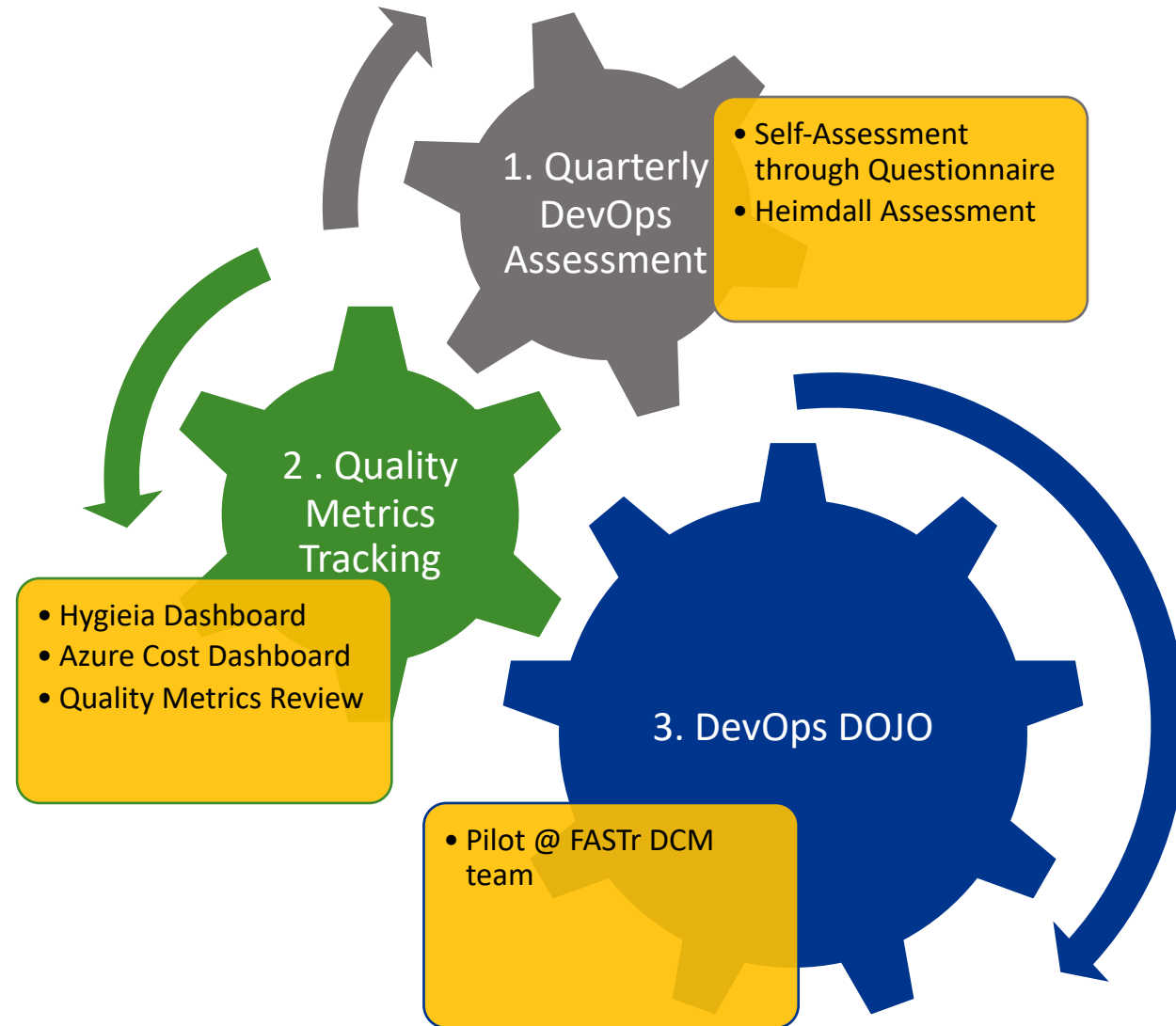


Sam's DevOps Goals

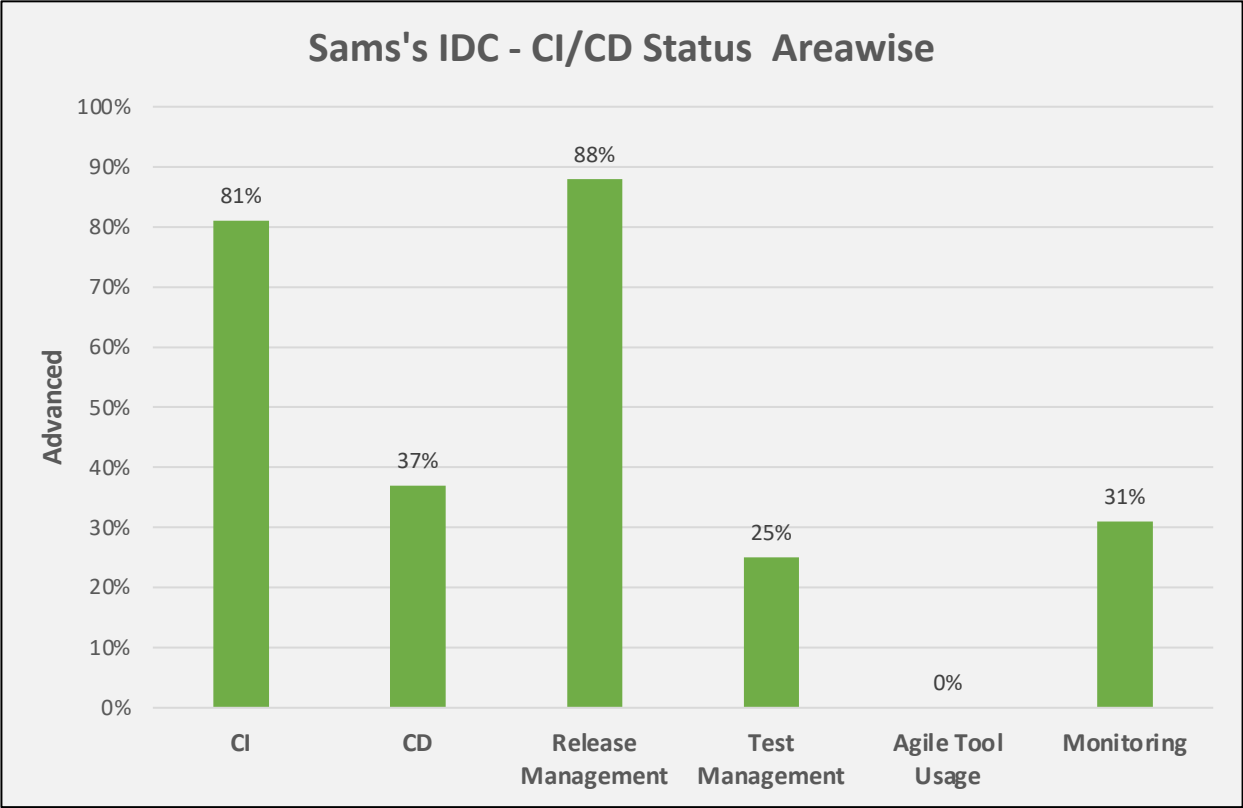
- Engineering alignment and business responsiveness
- Faster, smaller, more frequent releases at the end of the sprint
- Improved – Time to market
- Quality of code, products and services
- Productivity ,Customer satisfaction , Employee satisfaction
- Less waste and fewer defects
- Lower long-term costs



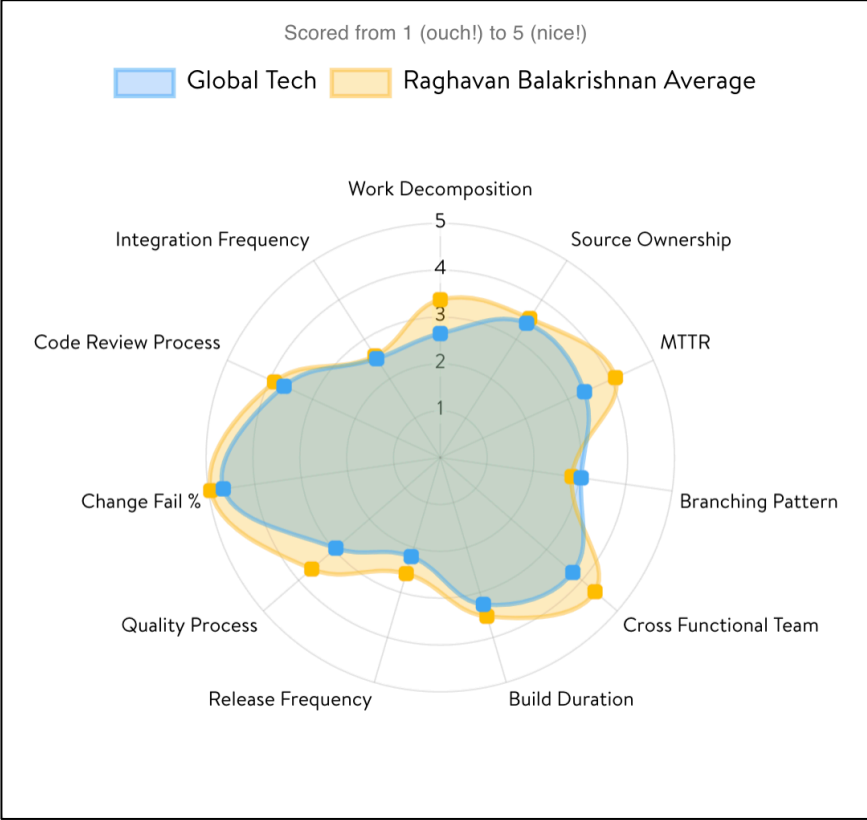
How did We start our journey?



Quarterly DevOps Assessment



Q1 Assessment Report



Q2 Assessment Report in Heimdall



Quality Metrics



Metrics	Definition	Ideal Trend	Target Value
Time to Build (TTB)	Average time it takes for new requests to be ready to go to production	Decreasing	Less than the Sprint duration
Change Fail Rate	Percent of change resulting in rollback	Decreasing	0%
Code Quality	Static code analysis metrics from SonarQube	Decreasing	Less Critical and Blocker issues
Automated Code Coverage	Unit Test Coverage	Increasing	100%
Cloud Spend vs Budget	Monthly budget vs expenditure	Decreasing	Under budget
Cloud Efficiency	Buffer cost on Azure Cosmos, App Services and AKS	Decreasing	Optimize the wastage



DevOps DOJO Pilot Overview

What is DevOps DOJO



- Safer, faster, and more resilient delivery for every product
- Quality over speed with the knowledge that delayed delivery reduces quality
- Rapid detection of quality issues over inspecting completed work
- Automating everything possible to reduce damage from manual processes

Duration



- 8 sprints , each sprint was for 1 week

The Team



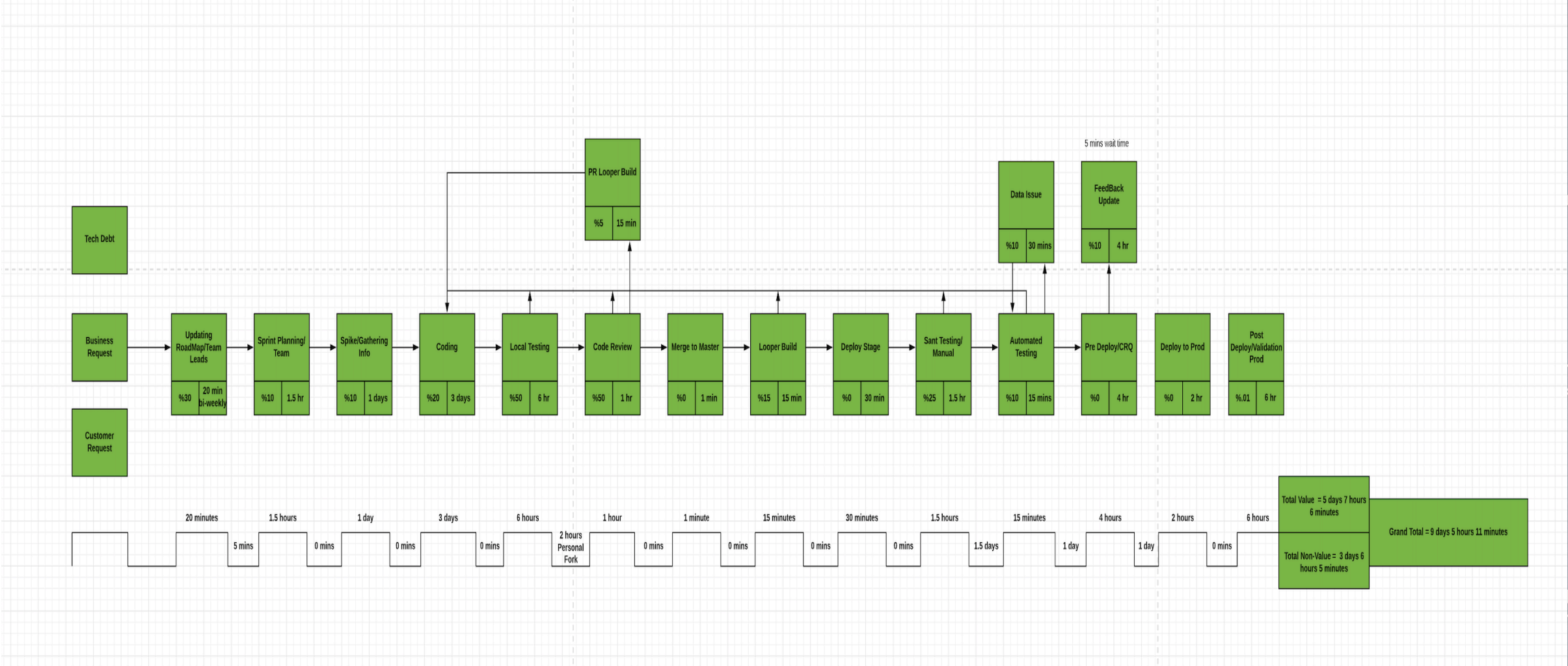
- FASTr DCM engineering team (Ashwin, Abhishek, Praveen, Sakthi)
- Program Management (Subha)
- DevOps DOJO (Bryan , Preston, Javier)
- IDC Agile Coach (Padma)

Contact: <http://devops.walmart.com>

You can't just keep doing what works one time, everything around you is changing. To succeed, stay out in front of change.

-- Sam Walton

VSM Workshop – First Point to enter DOJO





1. Knowing What to Build

- Do we understand how to measure the success of this feature?
- Do we know everything we need to know about how it's expected to behave?
- Do we understand and have a plan for all dependencies, met or not?
- Is this the right thing to build into our product?
- Have we finalized the exact behaviours to prevent gold plating?
- Do we know how to verify the behaviours with tests?

2. Definition Of Ready

- Description of the value the work provides (Why do we want to do this?)
- Testable Acceptance Criteria. (When do we know we've done what we need to?)
- The team has reviewed and agreed the work is ready (Has the team seen it?)

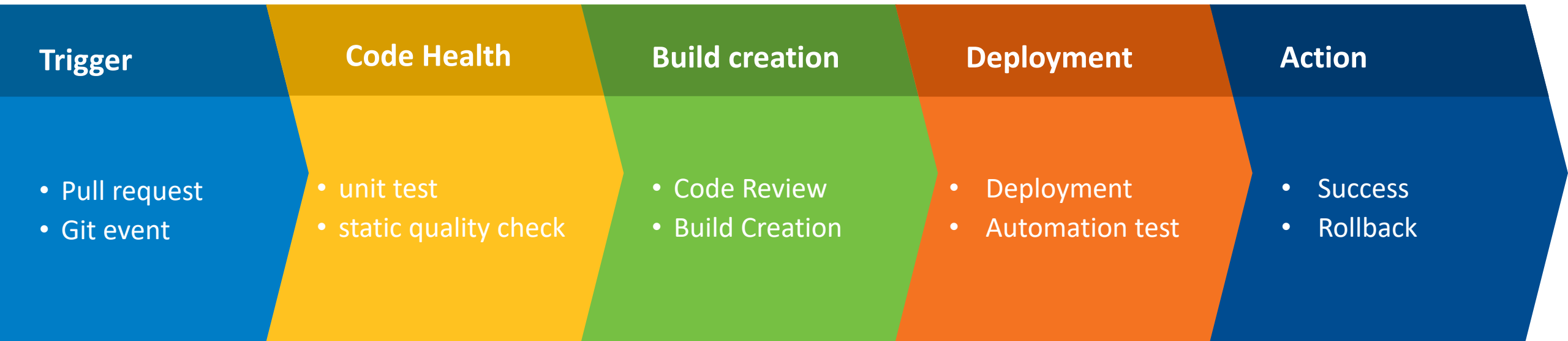
3. Definition Of Done

- All tests passed.
- All acceptance criteria have been met
- Code reviewed by team member and merged to trunk
- Demoed to team/stakeholders as close to prod as possible
- All code associated with the acceptance criteria deployed to production



CI CD With Concord

SUCCESS Automatic FLOW from merge to deployment in few Minutes



Automatic Fail Fast at every step for early detection of any issue

Computers perform repetitive tasks, people solve problems – People should focus on creative work, not testing, compliance, or rule verification.

One-Week Sprint



01



Better planning

With small sprint size it will be easy to Estimate work. Resulting in better planning and on time deliveries.

02



Negligible Ad-hoc Tasks

With small sprint size (one week) its easy to avoid any ad-hoc task and reprioritize upcoming user stories more frequently.

03



Spike & Implementation Rule

Always have spike-story and implementation story in separate sprints, this will help in predicting efforts more accurately.

06



Minimum dependency

With small sprint size its easy to detect any external dependency and very helpful in avoiding any blockers saving a lot of time of development team.

05



Weekly Product demo

With small sprint size by giving the deliverable demo to product every week we can quickly accommodate user inputs and enhancements if any required without wasting developer efforts.

04



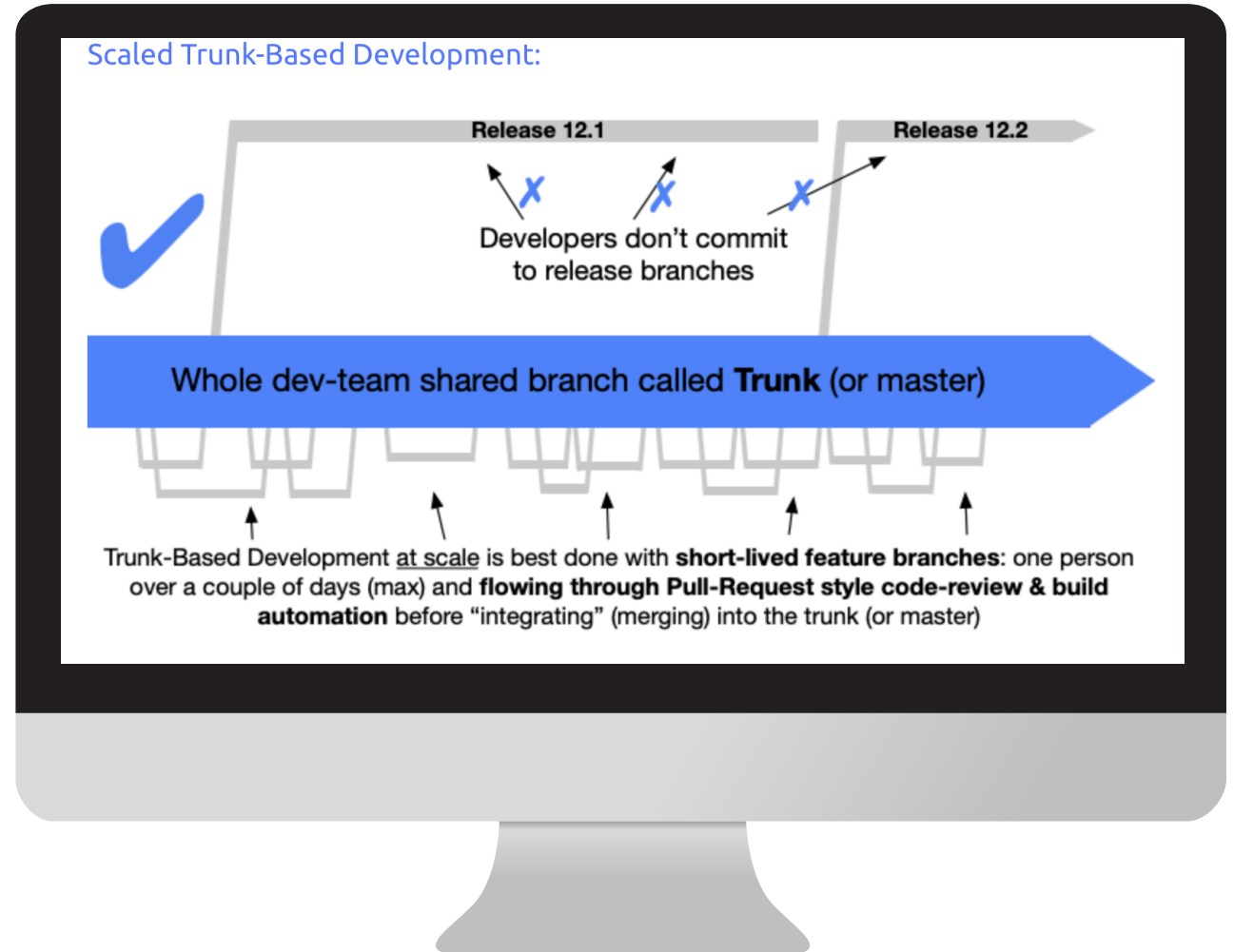
Fail Fast

With small sprint size it will be easy to validate the assumption and implementation differences quickly.

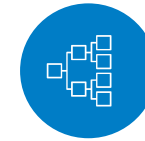
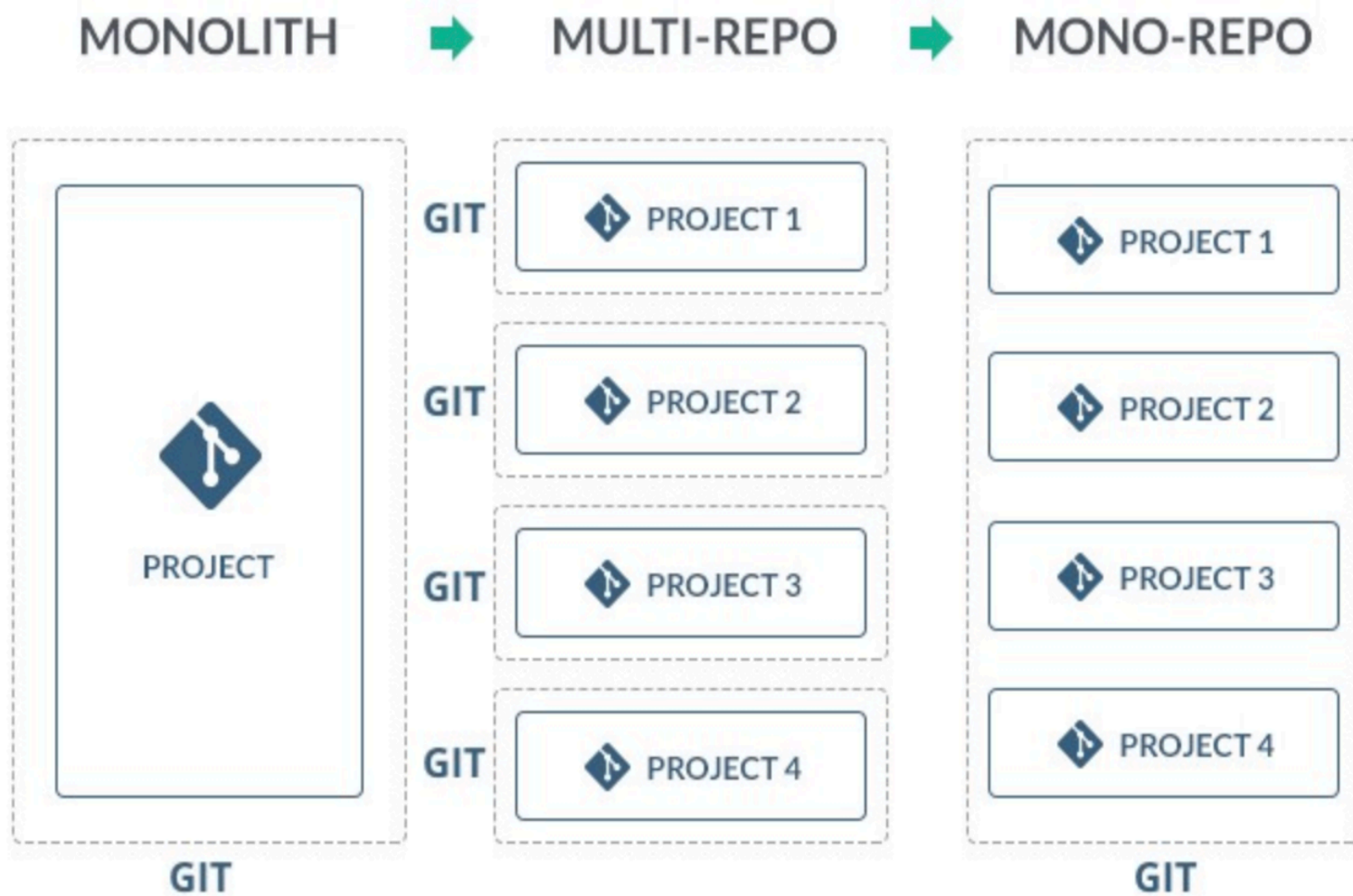


Code Health & Trunk Based development

- ✓ **Short lived Feature branch**
In this approach there will be a short-lived feature branch used by developer to be merged to master approx. every alternate day.
- ✓ **All releases from trunk**
All releases will be created from shared Trunk(master branch) and all feature branches will be deleted after PR merge.
- ✓ **Avoid using Forks**
Forks should be used by other teams doing any code change to your code instead own team. Resulting in long lived branches difficult to track
- ✓ **Code Health**
Always have code health tools like Sonar and Junit coverage to provide code health as code gate before merging.



Multi Repo over Mono Repo



Easy to maintain code



Easy to add and remove



Very useful in parallel development



Easy to break complex changes into simpler ones

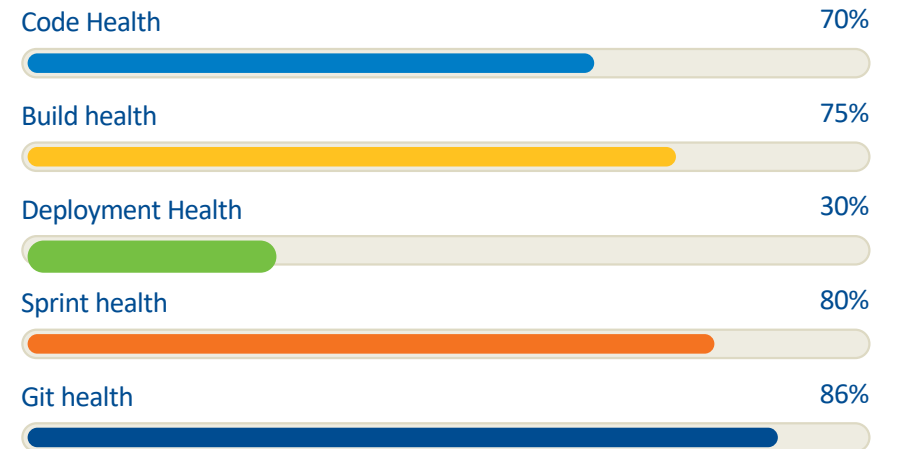


Overall score

40%

Overall system works as a machine , if one part fail whole system will fail .





- Use overall health metrics for better quality.
- Use tools like Hygieia & Heimdall to track your progress .



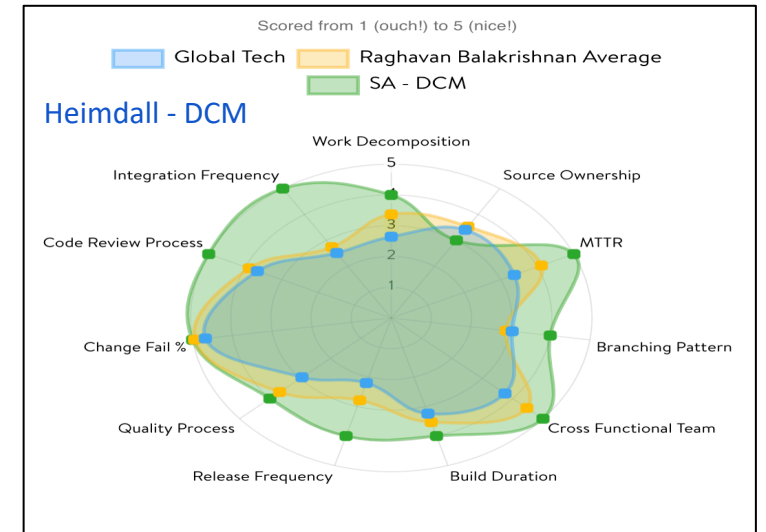
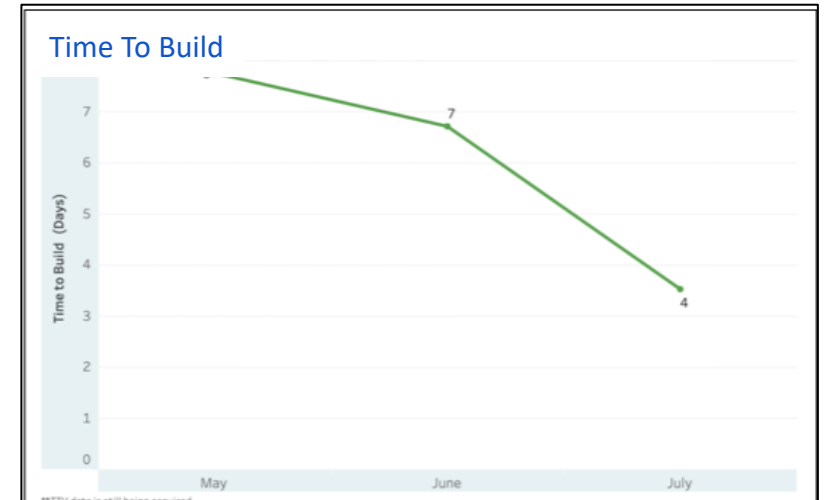
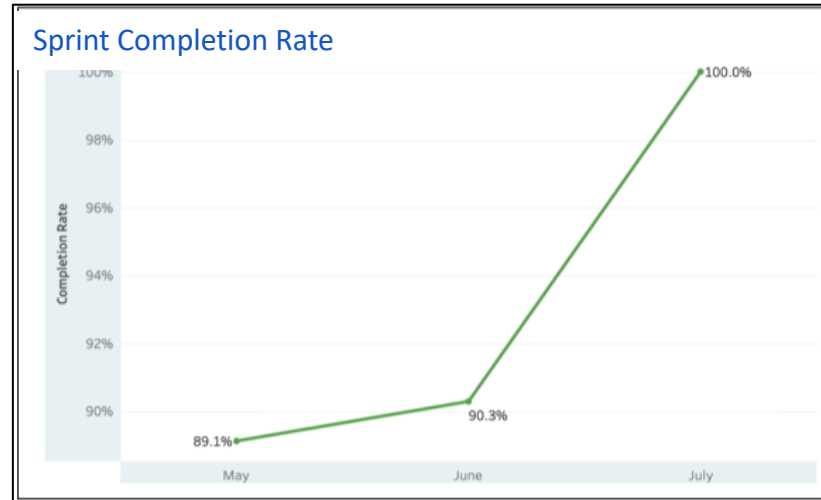
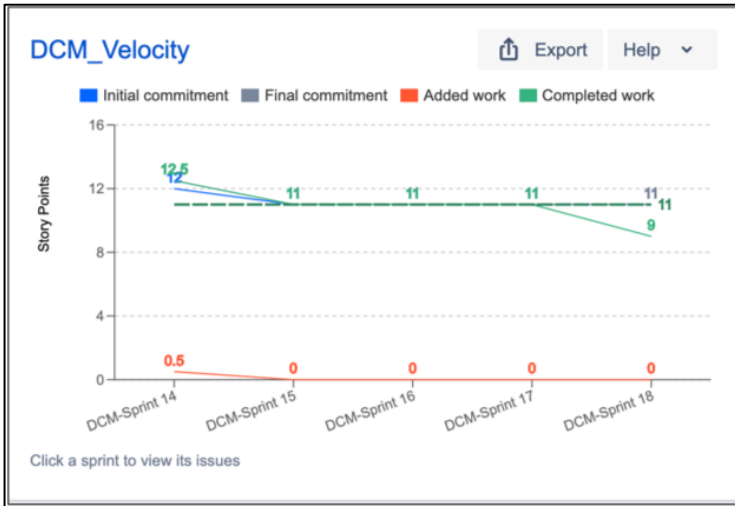
Adhere to Agile Best Practices



Pilot Outcomes

Improvement Metrics	Before (May)	After(July)
Time To Build	8 Days	4 Days
Sprint Completion	75.5%	100%
Unplanned Work	76%	0%
Sprint Duration	2 Weeks	1 Week
Velocity Prediction	-	11 Story Points
Backlog Maturity	0 (Used to do in the same sprint)	5 Sprints (Introduced N-1 sprint Backlog Refinement)
Team Health		  
Quality		0 Blockers Sonar issues

Metrics





THANK YOU

