PERFORCE

Break Through Bottlenecks in Your CI Pipeline

Robert Cowham,
Perforce Software

Who We Are

Robert Cowham
rcowham@perforce.com

Principal Consultant, Professional Services
Perforce Software

1 The CI Process

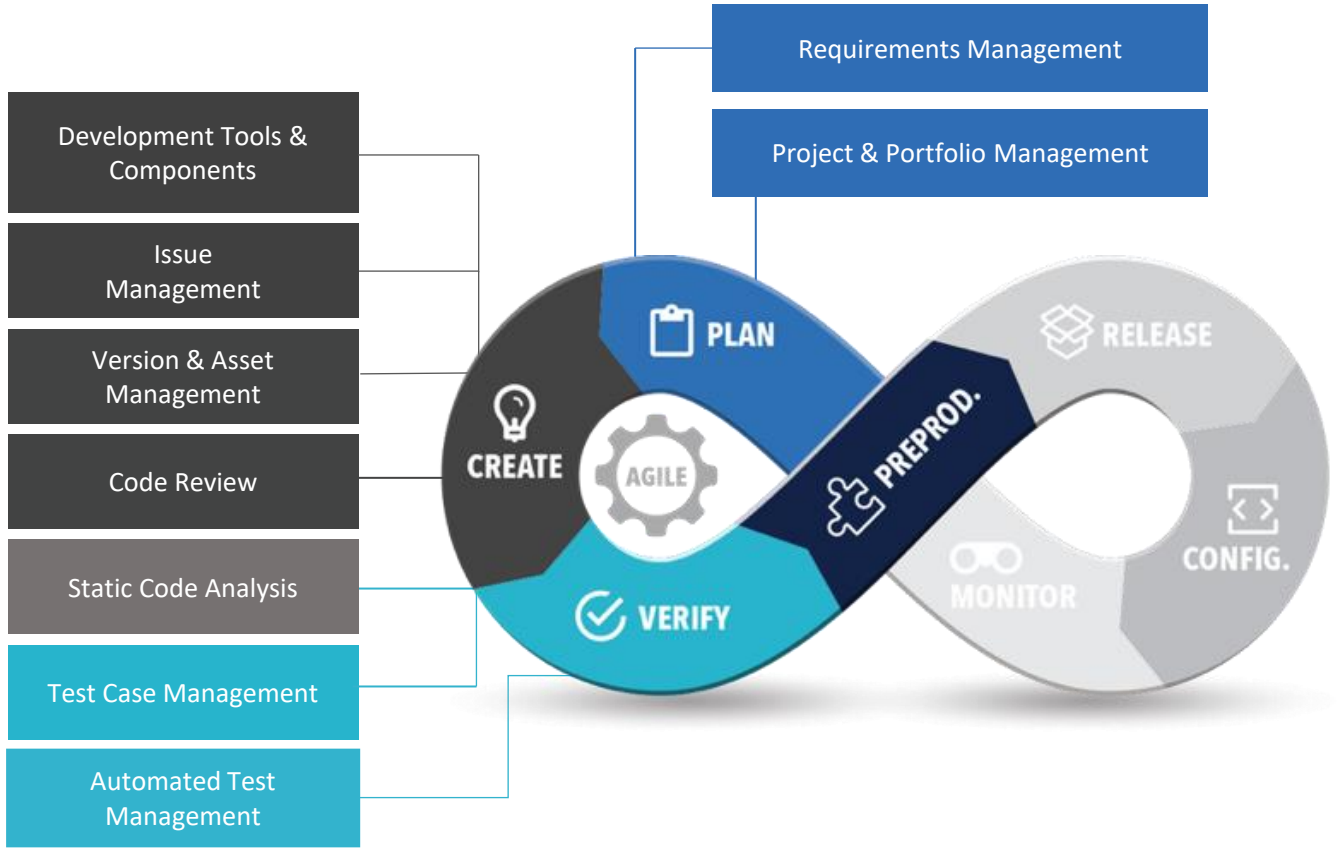
2 The Improvement Approach

3 Steps in the CI process

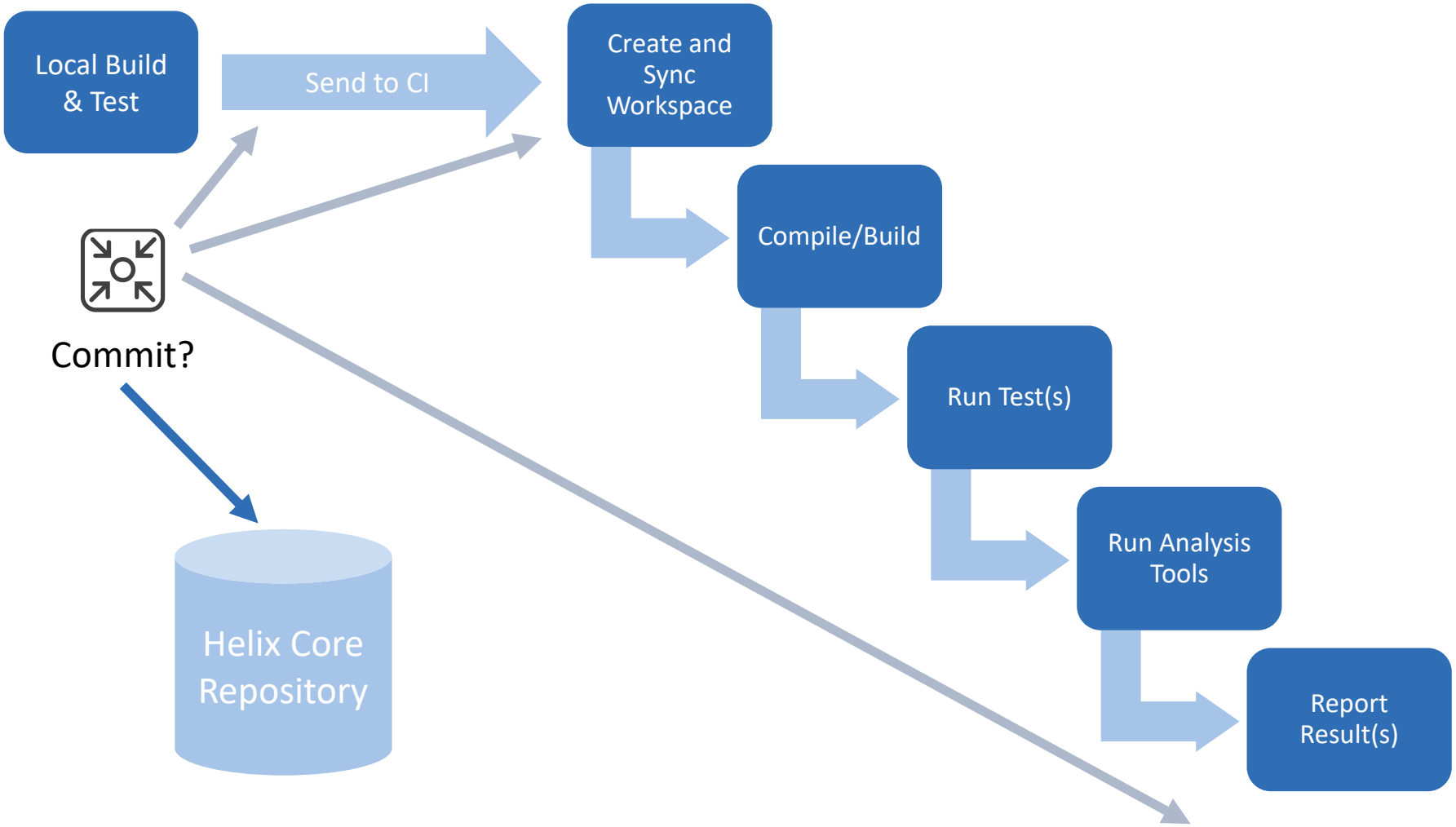
4 Summary

The CI Process

Where Does CI Fit in the Overall Lifecycle?



Typical CI Job



Improvement Approach

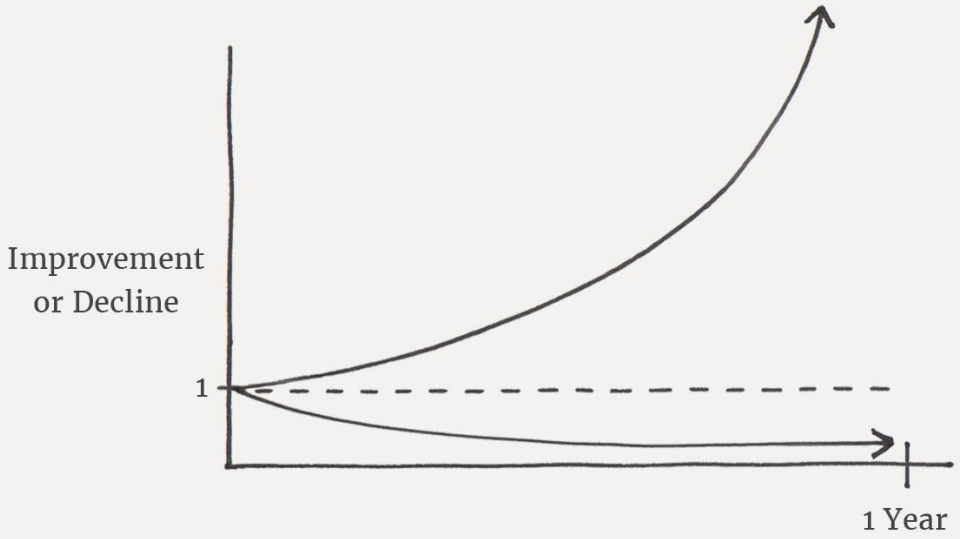
Continually Balancing Speed vs Quality

Achieving the
“Goldilocks Point”

The Power of Tiny Gains

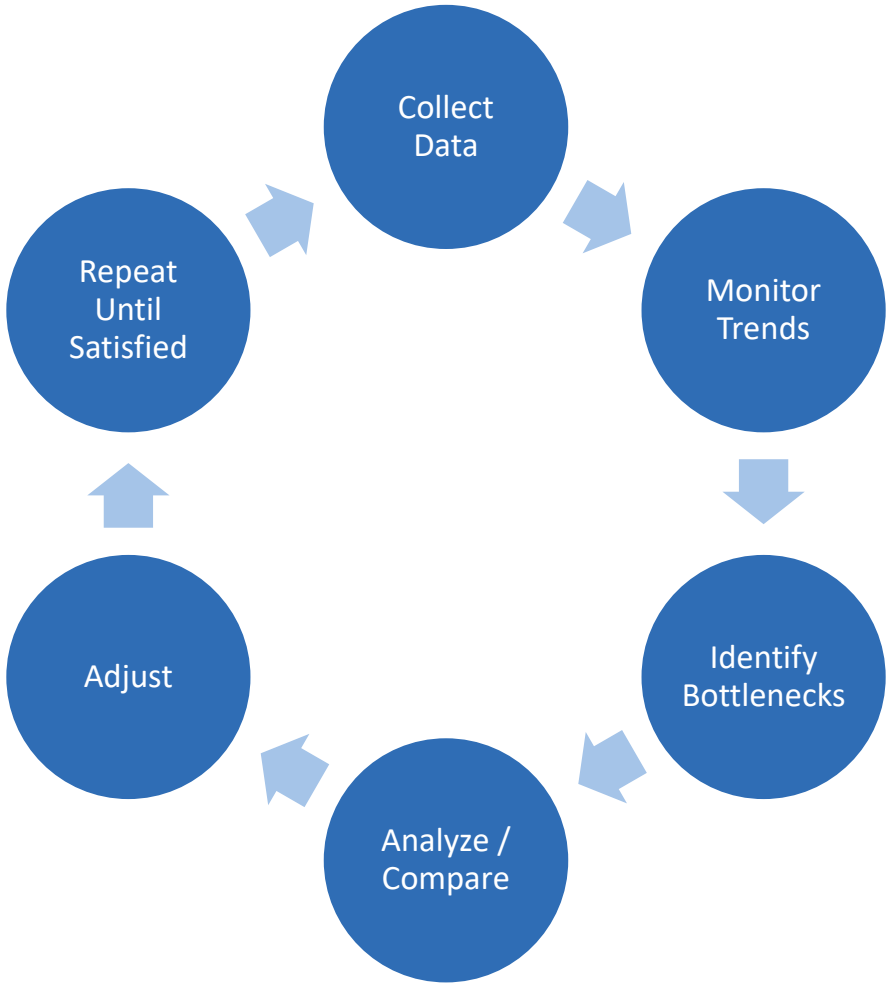
1% better every day $1.01^{365} = 37.78$

1% worse every day $0.99^{365} = 0.03$



JamesClear.com

Monitoring and Analysis



Improving Steps in the CI Process

Basic Tool Considerations

Compilation/Build Speed

Incremental vs Full
Parallelizable?

IDE Integration vs. CI Server

Cross Compilation?

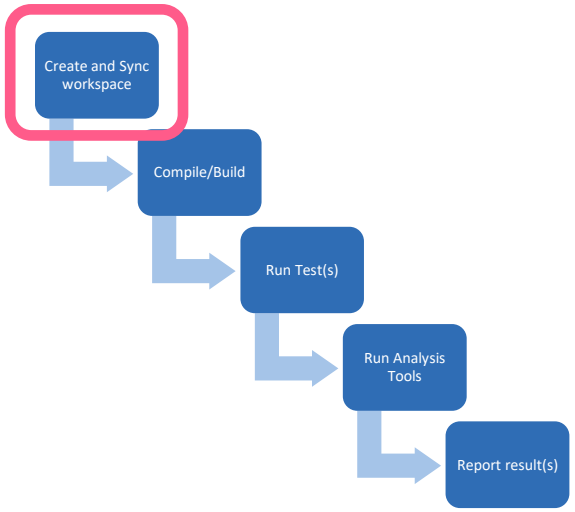
Version Compatibility/ Toolchain Dependencies

Embedded versions of tools can be behind state-of-the-art.

Tool Support Timeframes

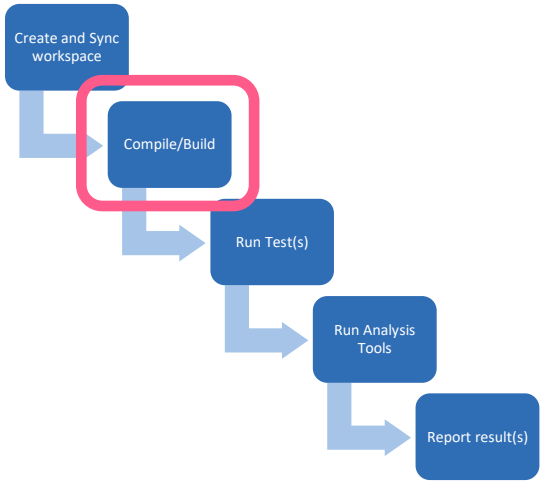
Syncing Workspaces

- New workspace per job vs. re-use workspace?
  - Incremental vs. full sync.
- Full clone vs. partial clone.
- Local caching of repository.
- How easy is it to set up build farms?
  - Installation of build tools?
  - Any difficult dependencies for installation?



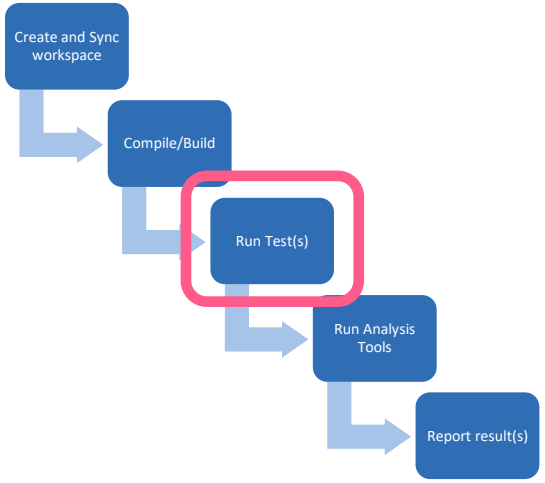
CI Compile & Build

- More likely full rather than an incremental build.
- Guaranteed reproducible?
- Parallelizable?
- Tool installation per node:
  - Special hardware requirements?
- Cross compilation vs. native.



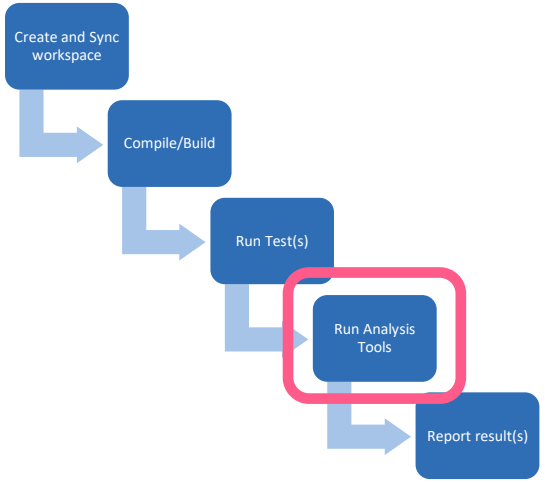
Testing Approaches

- Automation vs Manual
  - Not necessarily easy to achieve for embedded.
  - May require re-design/re-architecting for easy testing.
- Unit Tests vs. Integration/Functional
  - What is required?
- Don’t forget traceability requirements!
  - What information needs to be retained and how long for?

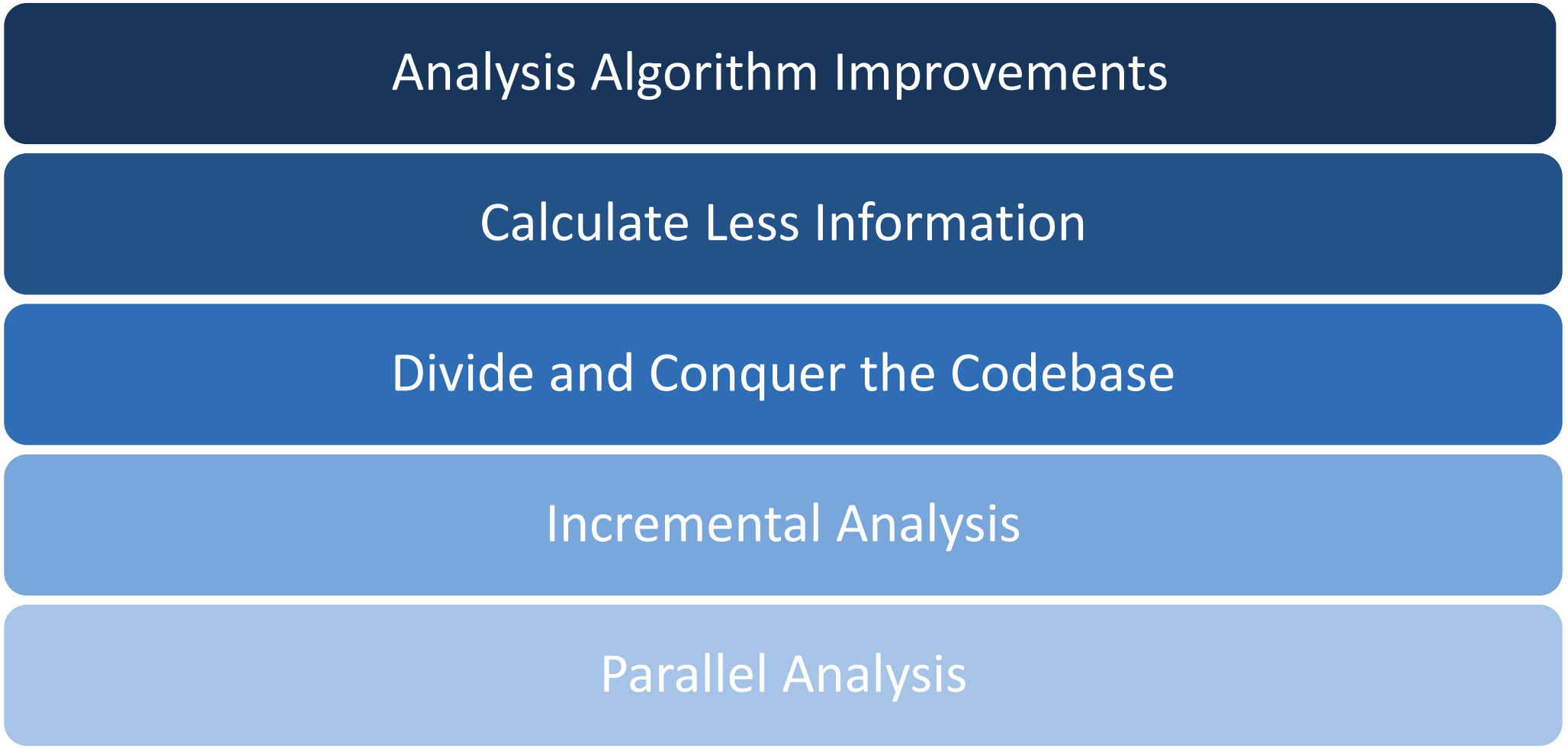


Analysis Options

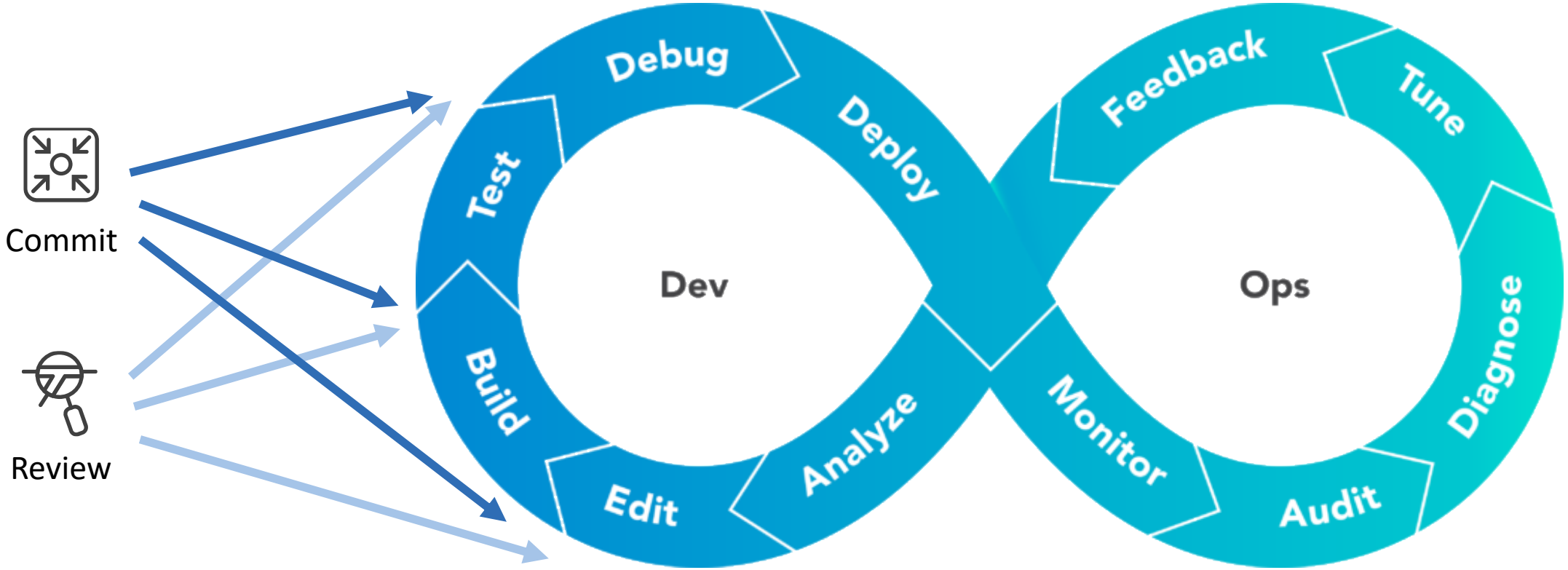
- Static Analysis
  - Quality, security, and compliance checks.
  - Benefits all software, but particularly embedded.
  - Can take a long time to perform.



Analysis — Practical Long-Term Solutions

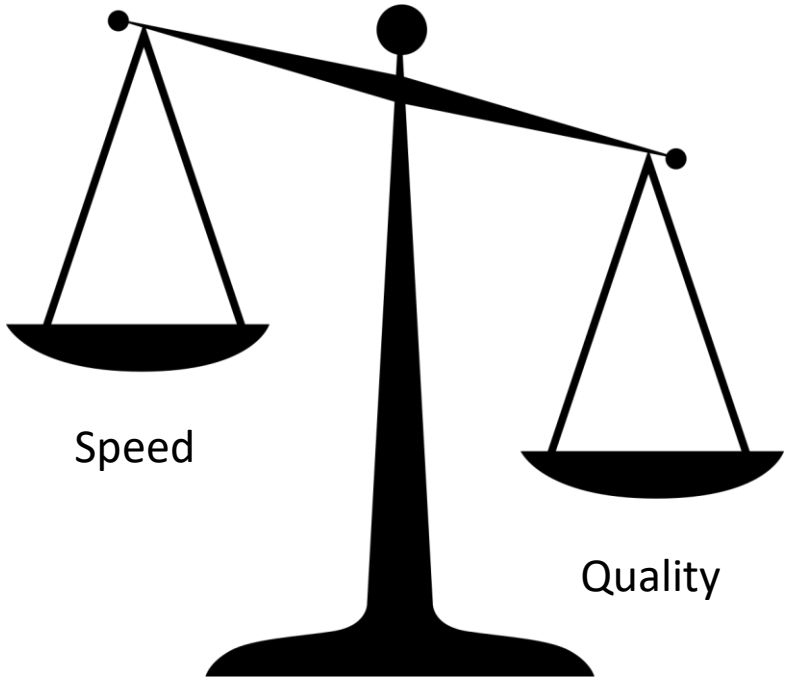


Code Review Process Models

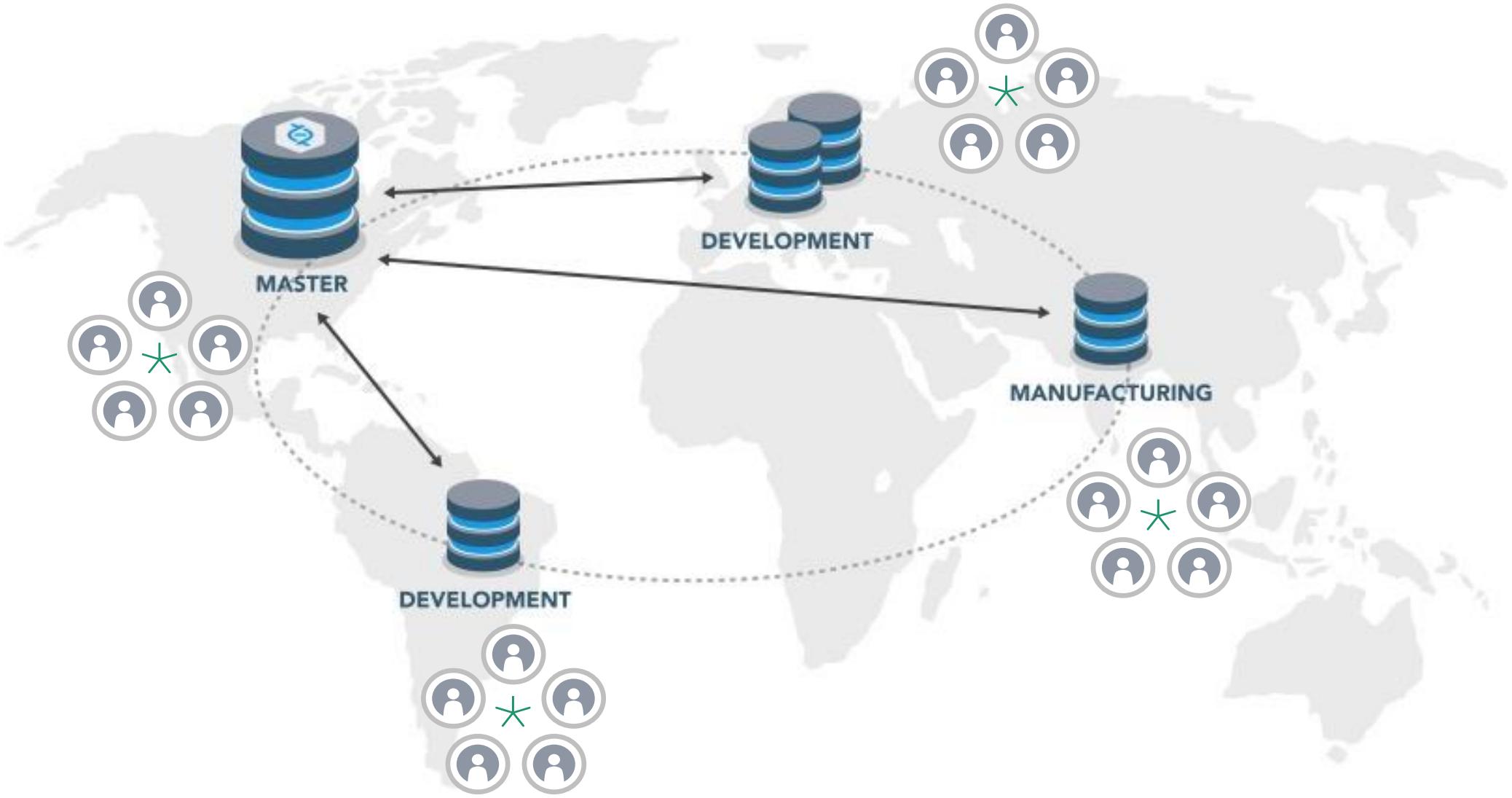


When to Review / When to Commit

- Pre-commit build (& test) / pre-commit review
  - Highest quality but slows developers down.
- Pre-commit build / post-commit review
  - Pragmatic balance between speed and quality.
  - Build doesn't get broken – not waiting for humans!
- Post-commit build / post-commit review
  - Risks breaking build, which impacts other team members.



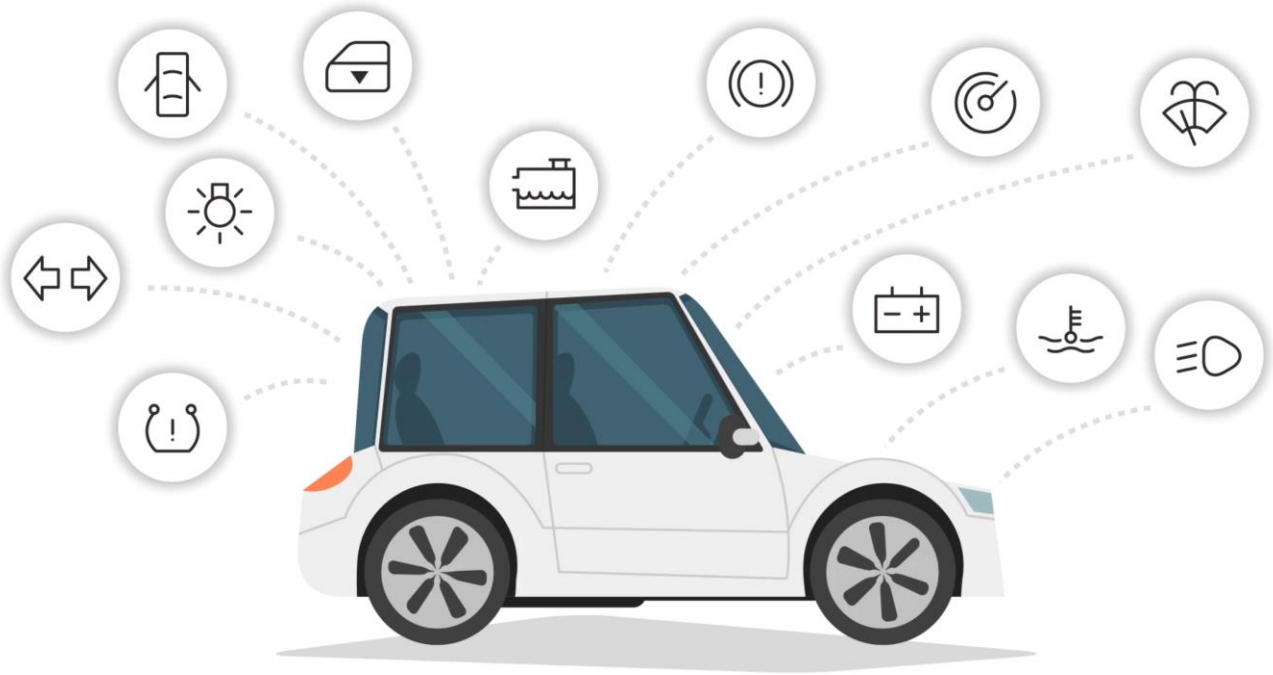
Distributed @ Scale



Case Study

“Drivers” for Embedded Software Development

- Very big mixture of projects — some small, some very large.
- Component-based development, always starts from certain base layers.
- Regulations and industry standards defines most of the process (AUTOSAR, Automotive SPICE).
- Products and components stay around for many, many years.



Summary

Summary – Optimizing Your CI Pipeline

- Understand the options for your pipeline.
  - Tools etc.
- Collect data for monitoring.
- Continuously improve.
- Achieve Nirvana!

Q & A