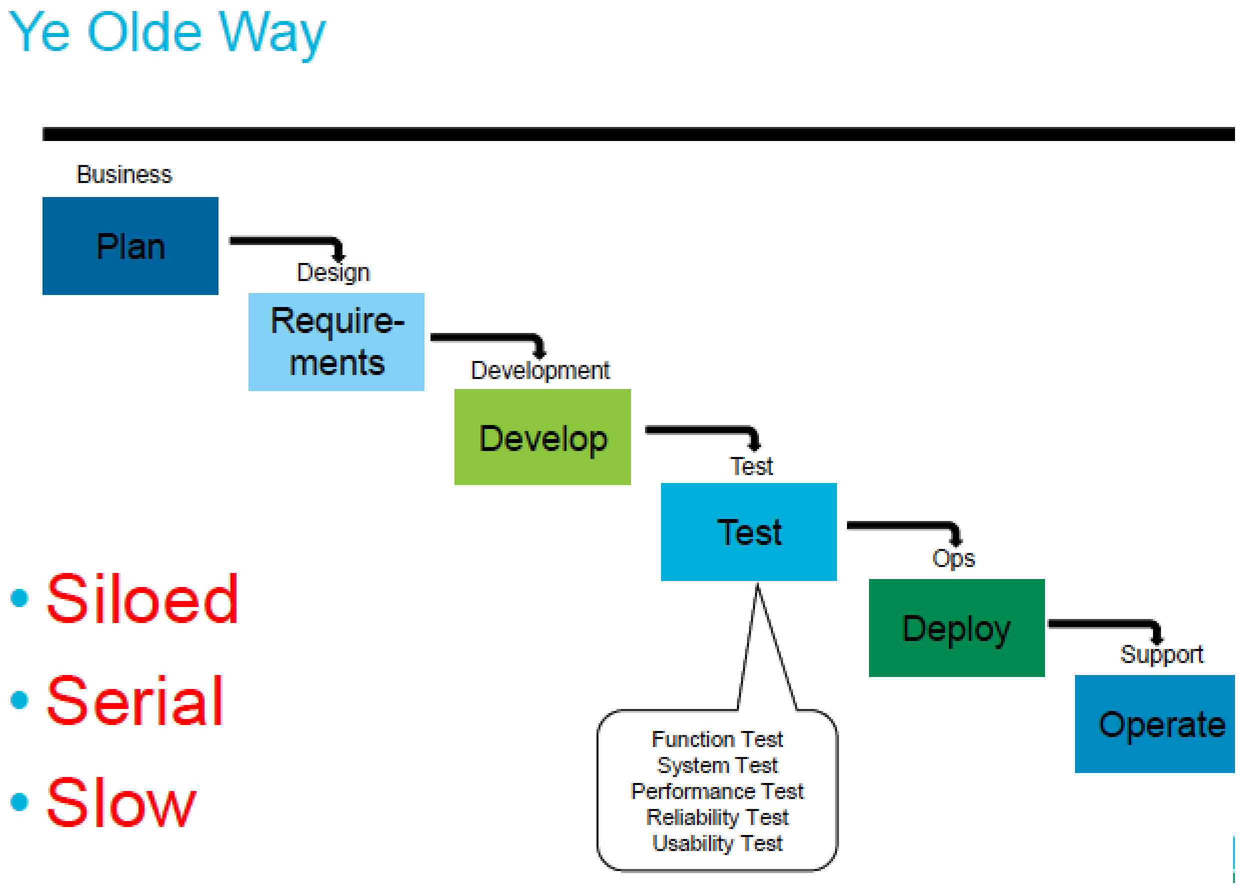
TATA CONSULTANCY SERVICES
Experience certainty.

DevOps

The Convergence of People, Process and Tools to enable Business Agility

Software Development Life Cycle – Waterfall Method

The Waterfall Model was first Process Model to be introduced. It is also referred to as a**linear-sequential life cycle model.**
The old way of doing things was siloed, serial and slow. It created tricky development problems, long test cycles and difficult deployments.



Ye Olde Way

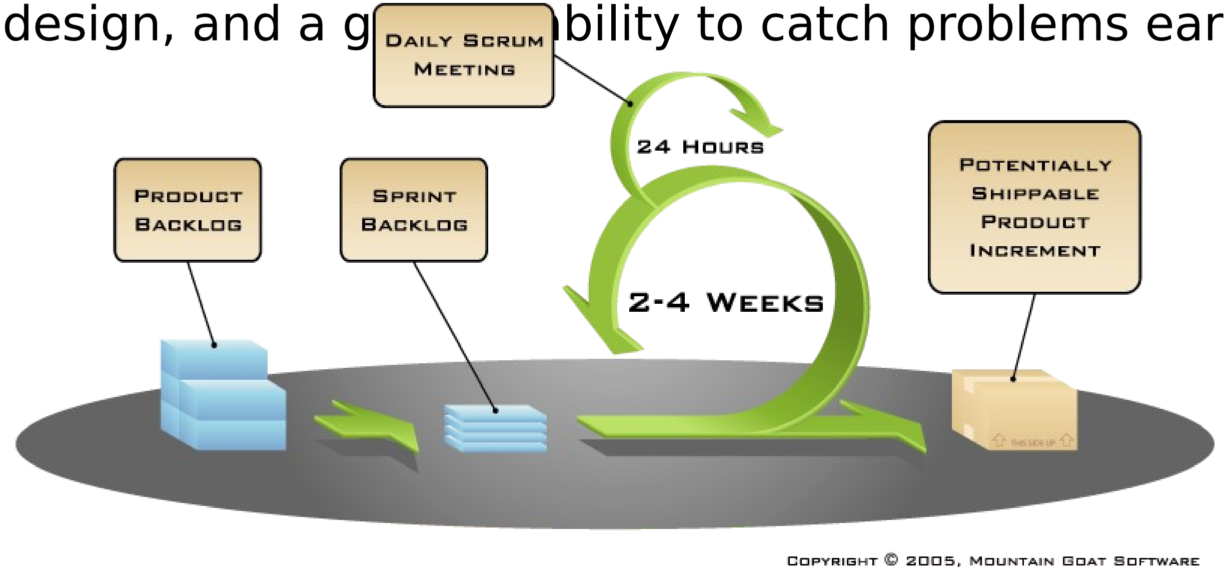
- Siloed
- Serial
- Slow

Challenges:

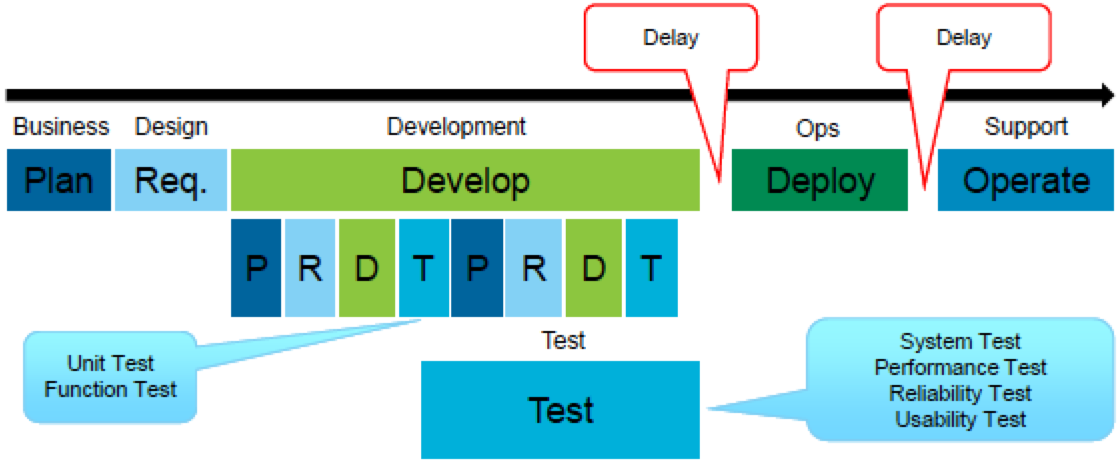
- *Software is planned like buildings: before any actual work is done. Changing plans later is expensive or impossible.*
- *Projects often have fixed agreement with regard to price and requirements.*
- *Testing starts after the development is ready*
- *Visibility into the development process is poor*
- *The system is only tested at the end of the execution phase when making changes is expensive. Some of the necessary fixes may even be impossible to implement*

Software Development Life Cycle – Agile Development

Eventually, organizations figured out a more efficient way of operating with agile. Agile development allows teams to get products to market faster with less up front planning and design, and a g bility to catch problems earlier.



Copyright © 2005, Mountain Goat Software



Agile methods approach software development
as an iterative process and emphasizes communication amongst Business, Dev & QA, but it does not address the collaboration with the IT operations.

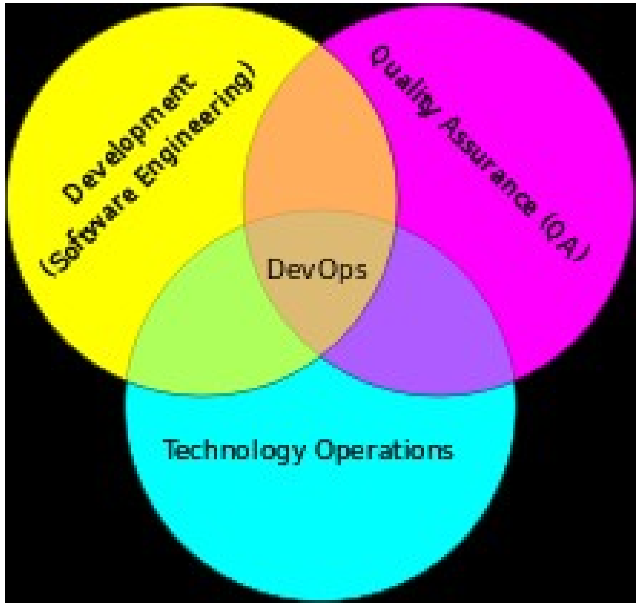
Challenges:

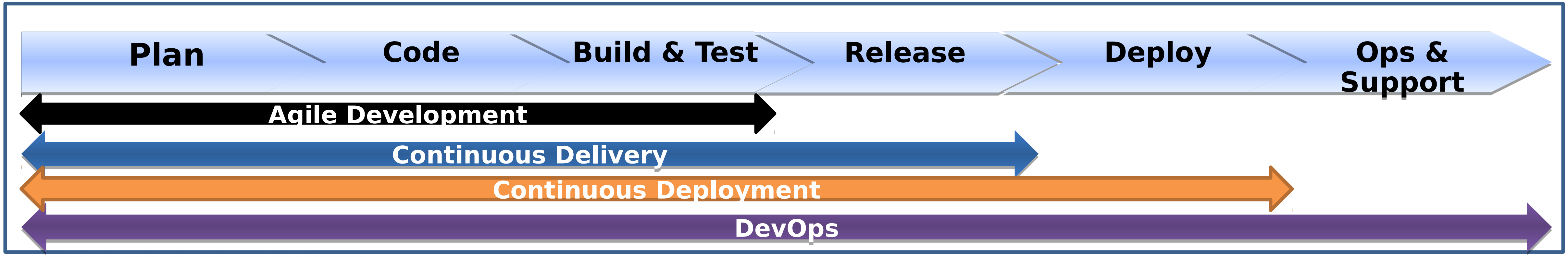
- *Difference in Dev & Ops environments*
- *Minimal/ No handshake with Operations + Less Documentation*
- *Release requires approval cycles*
- *Process & tools being used in Dev & Ops are different. Agility is there in development process and delay in deployment onto production env*
- *Teams in the production chain have their own separate goals that contradict each other*

What is DevOps

In traditional organizations, there are separate and distinct groups for development and operations.
The development team is responsible for writing code and delivering software.
The Ops team is responsible for deploying releases, managing systems, security and
environment stability.

DevOps is about People, Process and Technology.
DevOps is a set of principles, methods, and enabling technologies to address the goal
of rapidly releasing high quality software from development to operations .





Plan
Code
Build & Test
Release
Deploy
Ops & Support

Agile Development
Continuous Delivery
Continuous Deployment
DevOps

Why DevOps?

So, why DevOps?

DevOps breaks down communication barriers between Ops and Dev.
It promotes collaboration to solve problems experienced by both teams. The primary focus is on efficiency and reducing risk when building and deploying software.
Avoid scenarios like, “Sorry the deployment failed. We didn’t test on that platform.”

Improve Deployments

Here are some common things that DevOps teams do to improve software releases:

- Increase efficiency – less waste
- Decrease time to commit software changes
- Automate tests
- Identify defects/issues quickly
- Automate the build process
- Simplify the deployment process
- Make deployments reproducible
- Automate as much as possible

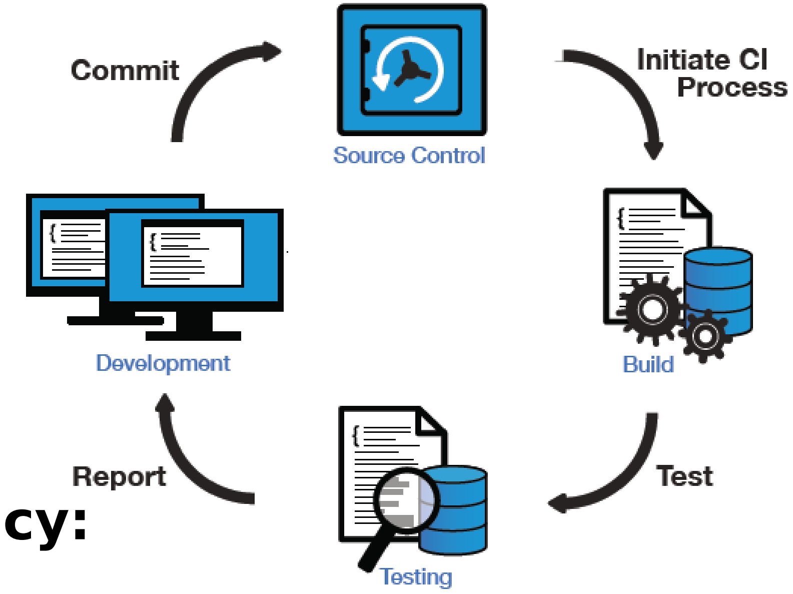
Continuous Integration

Continuous integration is the frequent merging of work with a main branch to simplify the merging process and test updates when integrated.

The concept usually involves a unit testing framework and a process to trigger builds and/or tests.

Principles of CI:

- Maintain a code repository
- Automate the build
- Make the build self-testing
- Everyone commits to the baseline every day
- Make it easy to get the latest deliverables
- Everyone can see the results of the latest build
- Facilitate automated deployments

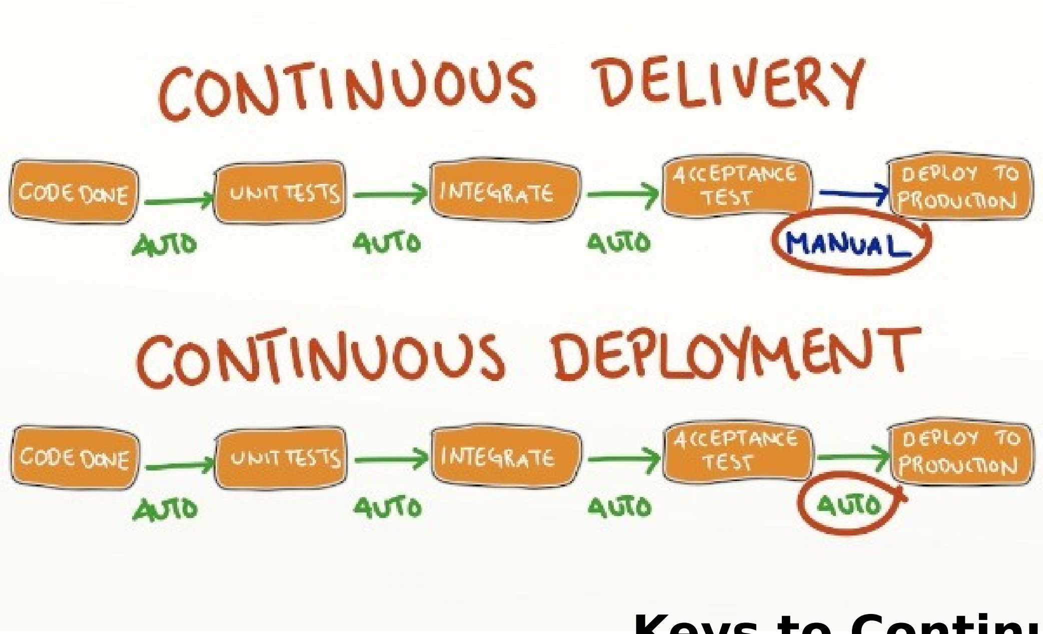


CI Improves Efficiency:

- Rapid Feedback
  - Identify problems early
  - Makes bugs easier to find
- Reduce bug accumulation
- Builds Automated
  - minimizes manual intervention
  - plug-ins (i.e. for static code analysis, gathering metrics)
- Precursor to Continuous Delivery & Deployment

Continuous Delivery/Deployment

Building on Continuous Integration, are the concepts of Continuous Delivery and Deployment



Continuous Delivery is a software development discipline where you build software in such a way that the software can be released to production at any time.

Continuous Deployment is where every change goes through a pipeline and automatically gets put into production, possibly resulting in many production deployments every day.

Keys to Continuous Delivery

- Process should be automated
- Reduces the number of features introduced per release, minimizing shock to users
- Will reduce the standard release cycle
- Changes approach to releasing software from an event to a non-event
- Helps to avoid off-hour, high risk, expensive deployments
- Know your rollback plan (do you rollback or roll forward only)
- Build in health checks.

DevOps is CAMS

CAMS is an acronym describing the core values of the DevOps Movement: Culture, Automation, Measurement, and Sharing

DevOps is Culture

- Eliminate the blame game
- Open post-mortems
- Rewarding failure
- Release often

DevOps is Automation

- Deploy
- Control
- Monitoring
- Config Management
- Orchestration

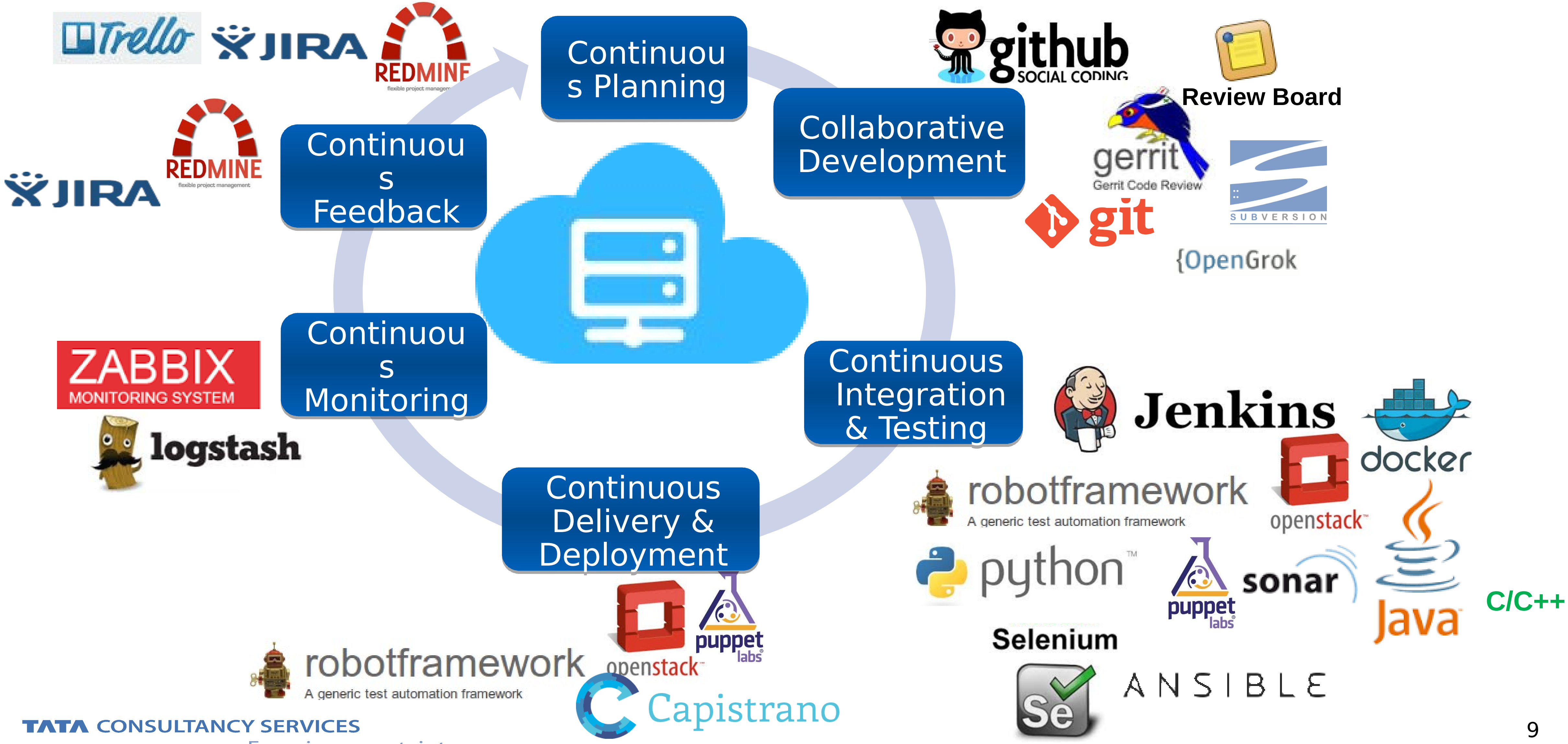
DevOps is Measurement

- Performance metrics
- Logs
- Metrics mapped to business goals
- Integration with people

DevOps is Sharing

- Feedback loops
- Enables the Automation and
- Measurement
- Significantly impacts Culture

DevOps LifeCycle| a few Open Source Tools



Continuous Planning

Collaborative Development

Continuous Integration & Testing

Continuous Delivery & Deployment

Continuous Monitoring

Continuous Feedback

Essence of DevOps - Improve Productivity & Problem Solving Skills

Better Software, Faster

Pride in the Software You Build and Operate

Ability to Identify, Respond and Improve Business Needs

TATA CONSULTANCY SERVICES
Experience certainty.

TATA

Thank you