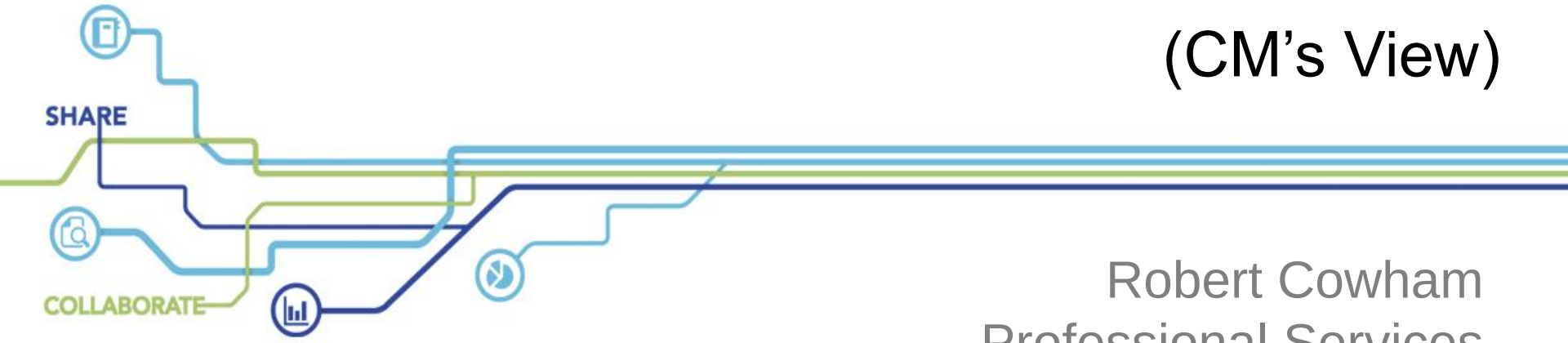
PERFORCE
Version everything.

bcs The Chartered Institute for IT
Configuration Management Specialist Group

The Real CD

Continuous Delivery Success

(CM's View)



Robert Cowham
Professional Services
Perforce Software

BCS CMSG Vice Chair

Agenda

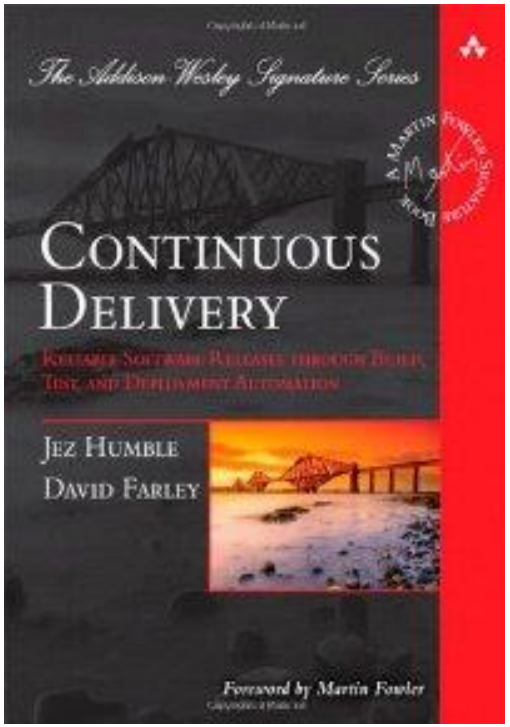
PERFORCE
Version everything.

- What is Continuous Delivery?
  - Management View vs Reality!
- Key metrics
- Key challenges
  - CM challenges
- Case studies

Continuous Delivery - Background

PERFORCE
Version everything.

- Continuous Delivery ("The Book") was published in 2011 by Jez Humble and David Farley
  - It builds on pre-existing practices and methods
- Agile manifesto
  - Our highest priority is to satisfy the customer through early and **continuous delivery** of valuable software

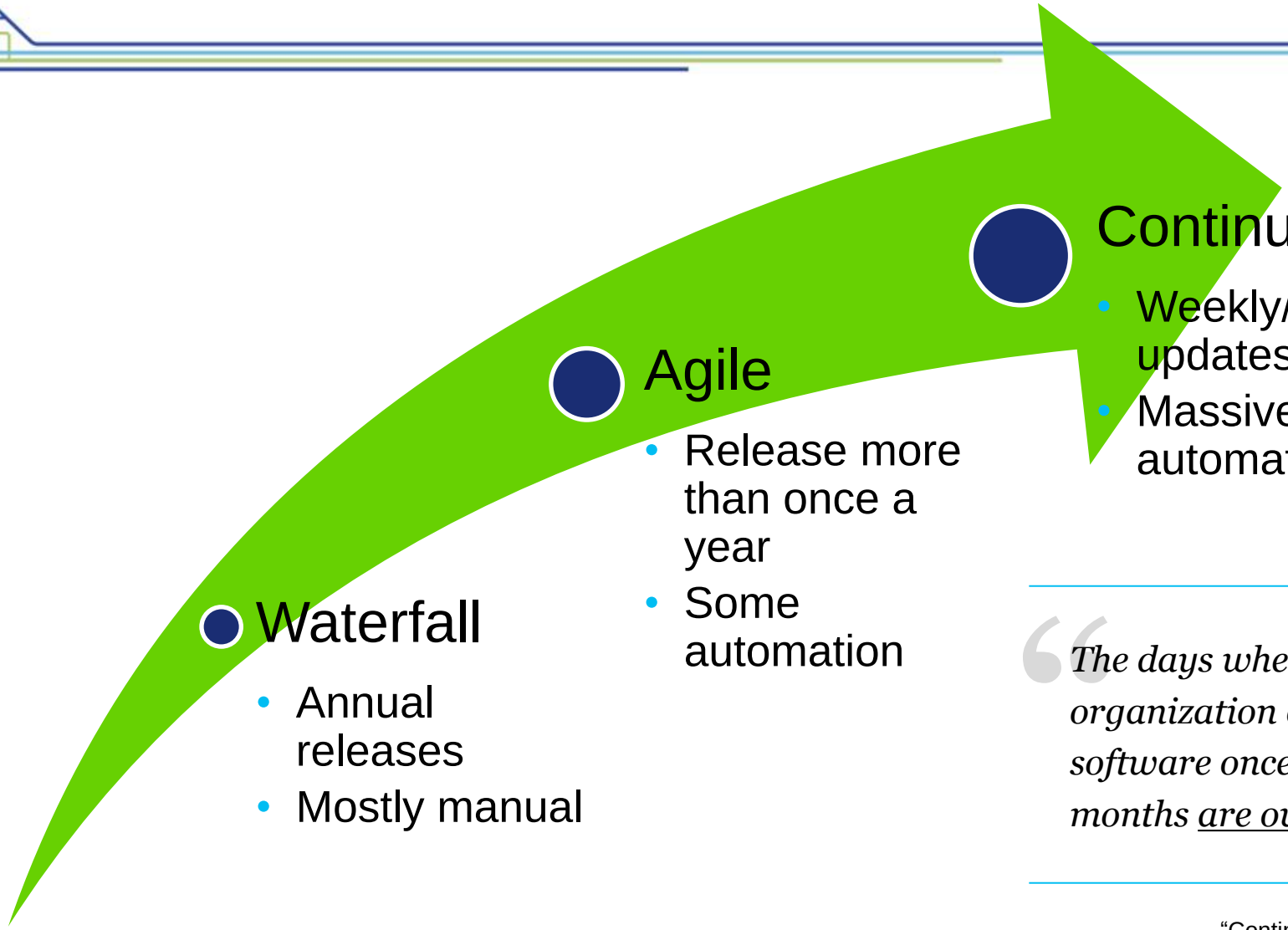


- So – how many of your organisations are doing Continuous Delivery?

bcs The Chartered Institute for IT
Configuration Management Specialist Group

Industry Perspective

PERFORCE
Version everything.



> *The days when a successful organization could release software once every 12 to 18 months are over.*

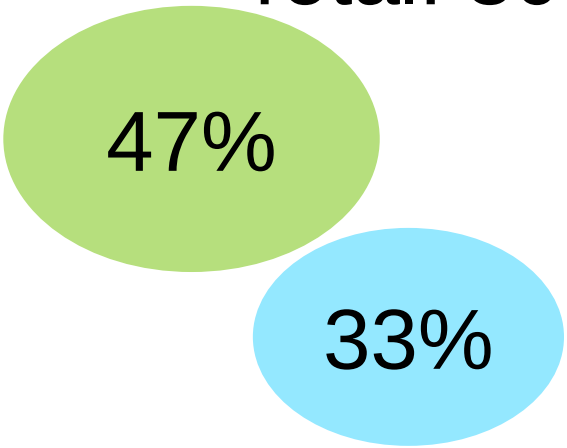
"Continuous Delivery is Reshaping the Future of ALM,"
Kurt Bittner, Forrester, July 2013

It's Not Just for SaaS Anymore

PERFORCE
Version everything.

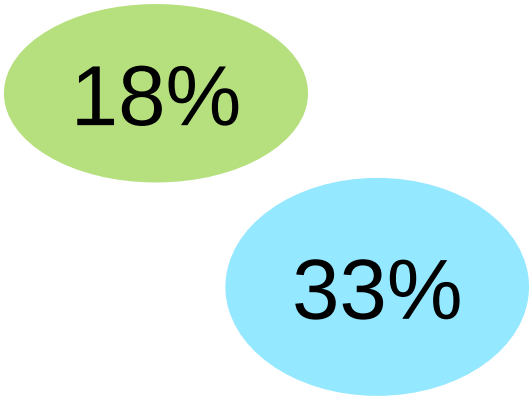
SaaS Companies

Total: 80%



Non-SaaS Companies

Total: 51%



All Projects

At Least Some Projects

bcs The Chartered Institute for IT
Configuration Management Specialist Group

Source: Evans Data research 2013

Keeping up with the Joneses

PERFORCE
Version everything.

Reality
(All companies)

28%

All Projects

Vs.

Perception
(Competitors)

46%

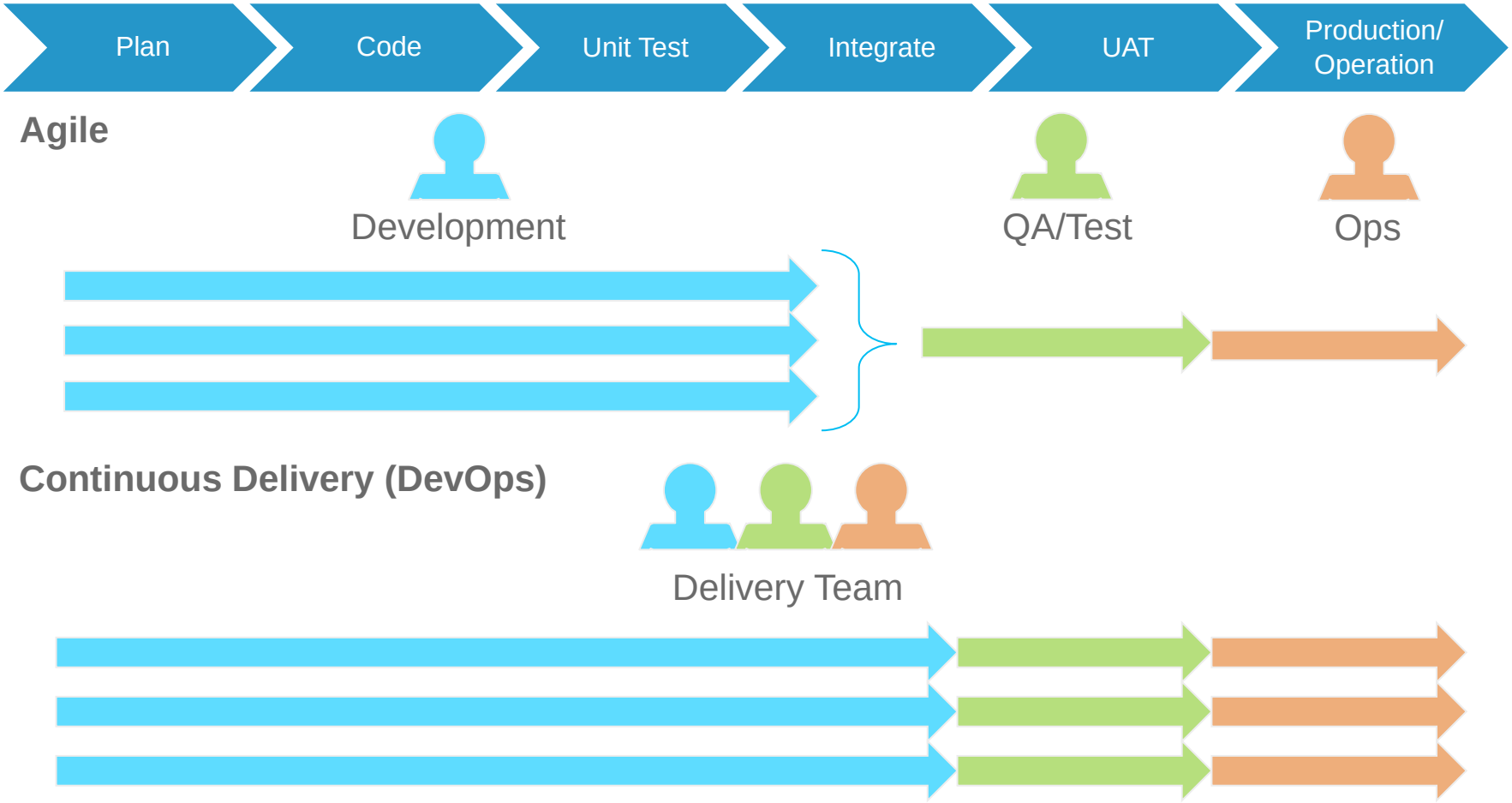
think their competitors have fully embraced Continuous Delivery

bcs The Chartered Institute for IT
Configuration Management Specialist Group

Source: Evans Data research 2013

Agile vs Continuous Delivery

PERFORCE
Version everything.

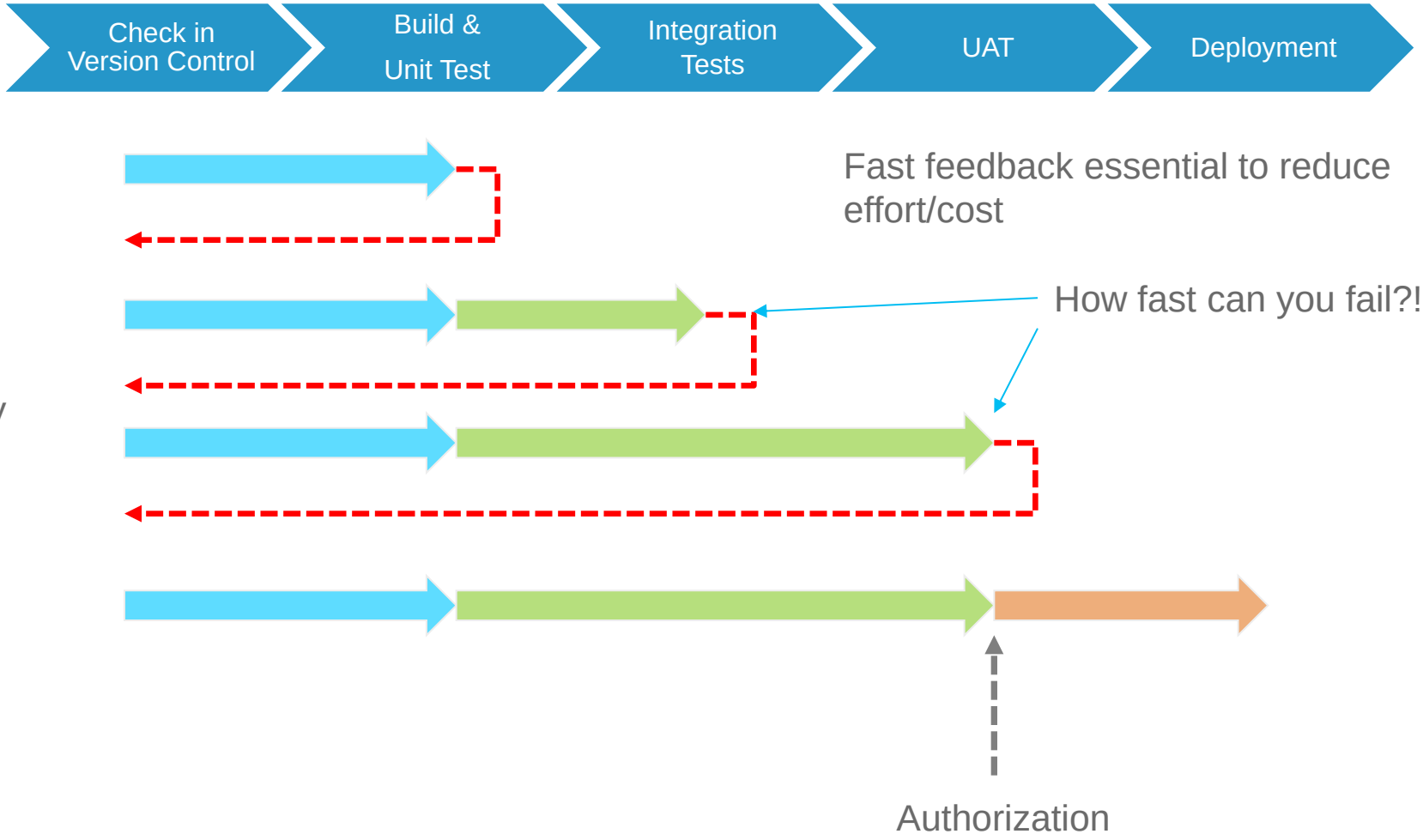


Key metric – cycle time

- How long would it take your organization to deploy a change that involved a single line of code?
- Do you do this on a repeatable, reliable basis?

Mary and Tom Poppendieck, *Implementing Lean Software Development (2006)*

Maximize flow/improve feedback



CD Challenges

PERFORCE
Version everything.

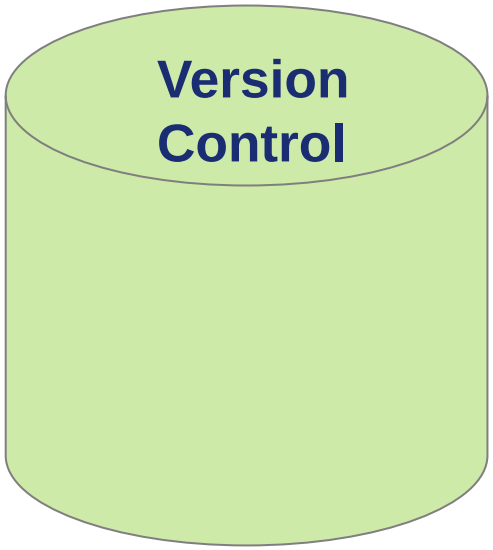
- Automation
  - Build & Unit test (isn't everyone doing this already?!)
  - Integration/Acceptance Tests
  - Other tests (performance/capacity)
  - Deployment
- Legacy technologies/code bases
  - Architecture
- Environments
- Anything else?!

What do we control/version?

PERFORCE
Version everything.

- Source code
- Build scripts
- Test scripts
- Environment configuration information
- Deployment tools
- Built executables?
- Other binaries/assets?
- Other configuration data?

How do we control it?

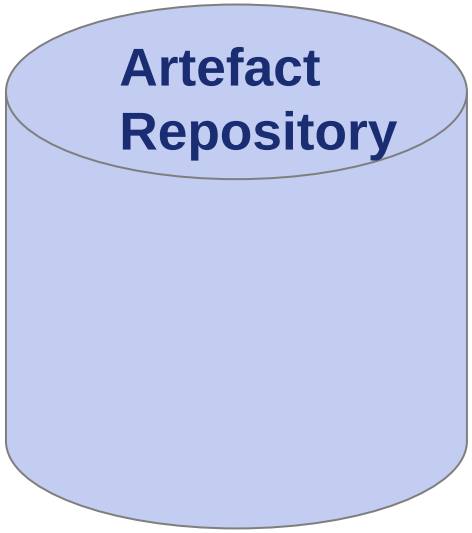


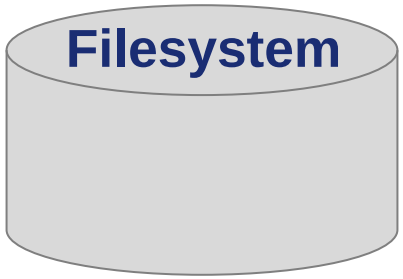
Source

Configuration

Binaries

Tools

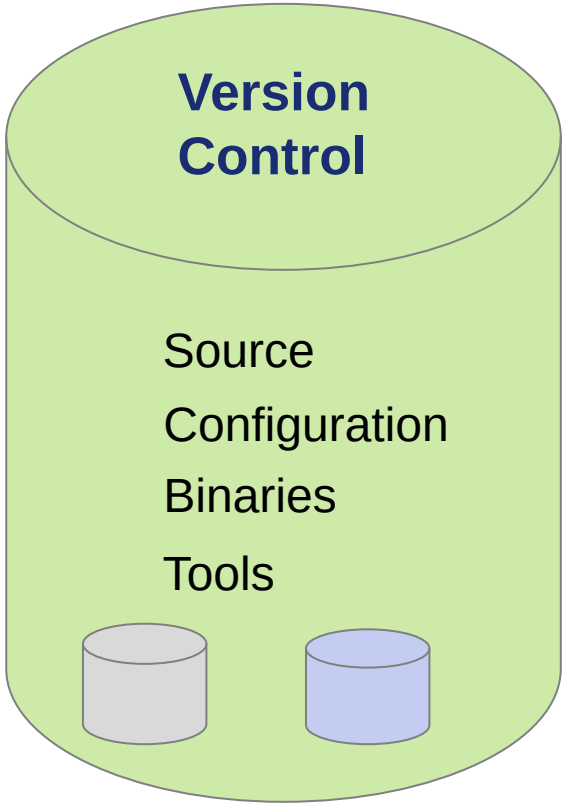




Where do we control it?

PERFORCE
Version everything.

- How to ensure traceability?
- How to provide audit trails?
- How to manage access controls?
- How many admins, processes, tools, …?



CD: the aggregation of marginal gains

PERFORCE
Version everything.

- Sir Dave Brailsford (Team GB Cycling Performance Director):
  - “If you broke down everything you could think of that goes into riding a bike, and then improved it by 1%, you will get a significant increase when you put them all together”
- Big things – training and conditioning
- Little things
  - Sleeping in the right position
  - Having the same pillow when you are away
  - Washing your hands properly

- Advantage over the French: “our wheels are round!”

(Marginal) Gains

PERFORCE
Version everything.

- Automation
- Reporting/Communicating Status/progress
  - Information radiators
  - Name and shame…
- Testing
  - Smoke tests
  - Test in parallel
- Deployment environments
  - Infrastructure as Code (DevOps)
  - Virtualisation technologies
- Database versioning

Case study – deployment scripts

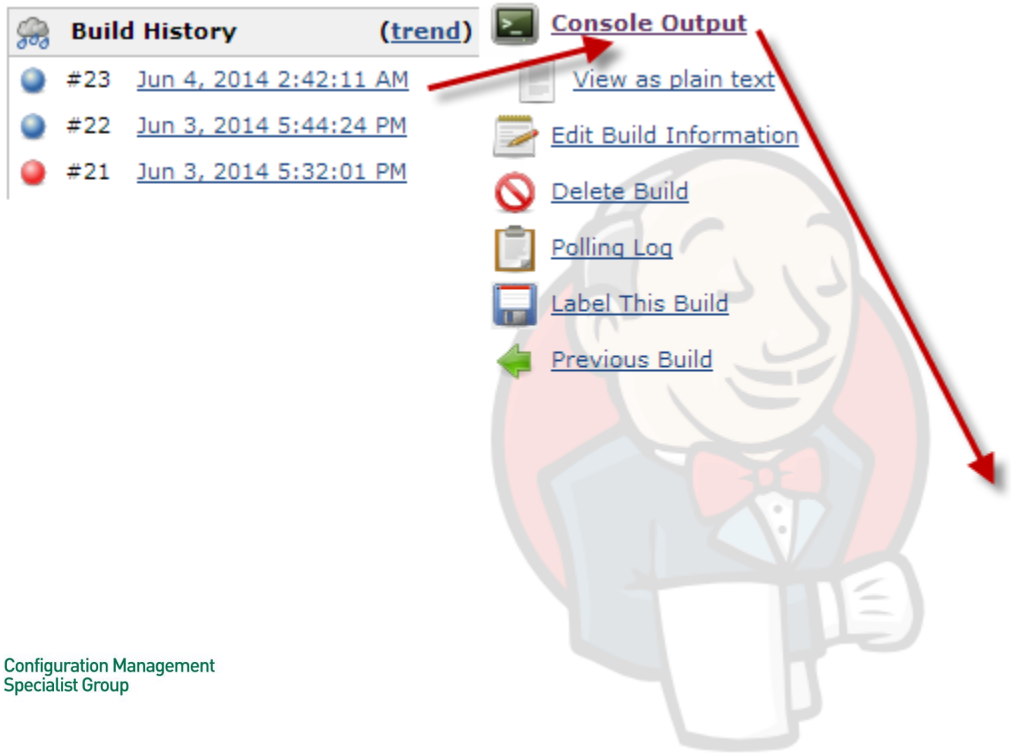
PERFORCE
Version everything.

- SDP (Server Deployment Package)
- Supports Multiple platforms:
  - Windows (.bat files)
  - Unix (bash)
    - Linux (CentOS/Redhat, Ubuntu/Debian, etc…), Solaris
  - Some python/perl

- Challenges
  - “It’s just scripts…!”
  - Different functionality on different platforms
  - Manual testing
  - Manual deployment

Case study – solution

- Phase 1
  - Test harness (Python)
  - Virtual machines (Vagrant) – Windows now supported too!
  - Cross platform installer (Python)




```
Building in workspace /var/lib/jenkir
SCM Task: cleanup workspace: jenkins-
... [list] = revert //...
... rm [list] | ABANDONED
... [list] = reconcile -n -a //...
... rm [list]
... [list] = reconcile -n //...
SCM Task: syncing files at change: 87
... sync //...@872000
[workspace] $ /bin/sh -xe /tmp/hudson
+ bash -ex run_tests.sh
+ echo Running SDP tests
Running SDP tests
+ vagrant up
+ vagrant ssh usdpmaster -c 'sudo -u
+ vagrant ssh usdpmaster -c 'sudo -H
+ vagrant ssh usdpmaster -c 'cp /tmp/
+ cat ./sdp/test.out
Wed Jun  4 01:42:25 UTC 2014
Starting /p4/1/bin/p4d_1: Perforce db
Rotating /p4/1/logs/journal to /p4/1/
.
----------------------------------------
Ran 1 test in 2.973s

OK
Finished: SUCCESS
```

Case Study – Financial Institution

PERFORCE
Version everything.

- 10 year old application (started life as spreadsheets)
- Successful - but now has >100 people involved!
- .Net and SQL Server
- 8 week cycle time minimum
- Automated build

- Issues
  - Few automated tests
  - Lots of manual process steps
  - "**Everything would be better if** *Developers / testers / CM team* **just followed the process**"

(*delete one!)

bcs The Chartered Institute for IT
Configuration Management Specialist Group

Improvement activities

PERFORCE
Version everything.

- People/team education/integration
- Test environments
  - Production vs “lite” database
  - Shared environments
  - Use same process for development/test/production deployment
- Automation
  - Expand test harnesses
  - Trial deployments – make it easy for developers to test
  - Merging

Version Everything – CCP Games

PERFORCE
Version everything.

- Version absolutely *everything*
  - Config data, tools, art assets…
  - Instant workstation configuration
- Make it easy for people to use

{ Versioning is the nerve center of the organization }

System of Record – NYSE

PERFORCE
Version everything.

- 14,000 servers, 6,600 production relea per year, 198+ active projects, only 6 people
- Build artifacts stored in SCM Repository
- SEC Audited
  - “the paradigm for maintaining production distribution and production auditing is so parallel to what you're trying to do in development”

Continuous Delivery in Real Life…

PERFORCE
Version everything.

A physical demonstration!

Best Habits for Success

PERFORCE
Version everything.

1. It's a whole team responsibility
   - CM needs to work with other teams
2. Think beyond the code - Version *everything*
3. Automate, Automate, Automate
4. Track every change
5. Put it all in one place

Questions?!

PERFORCE
Version everything.

Robert Cowham

rcowham@perforce.com

bcs The Chartered Institute for IT
Configuration Management Specialist Group