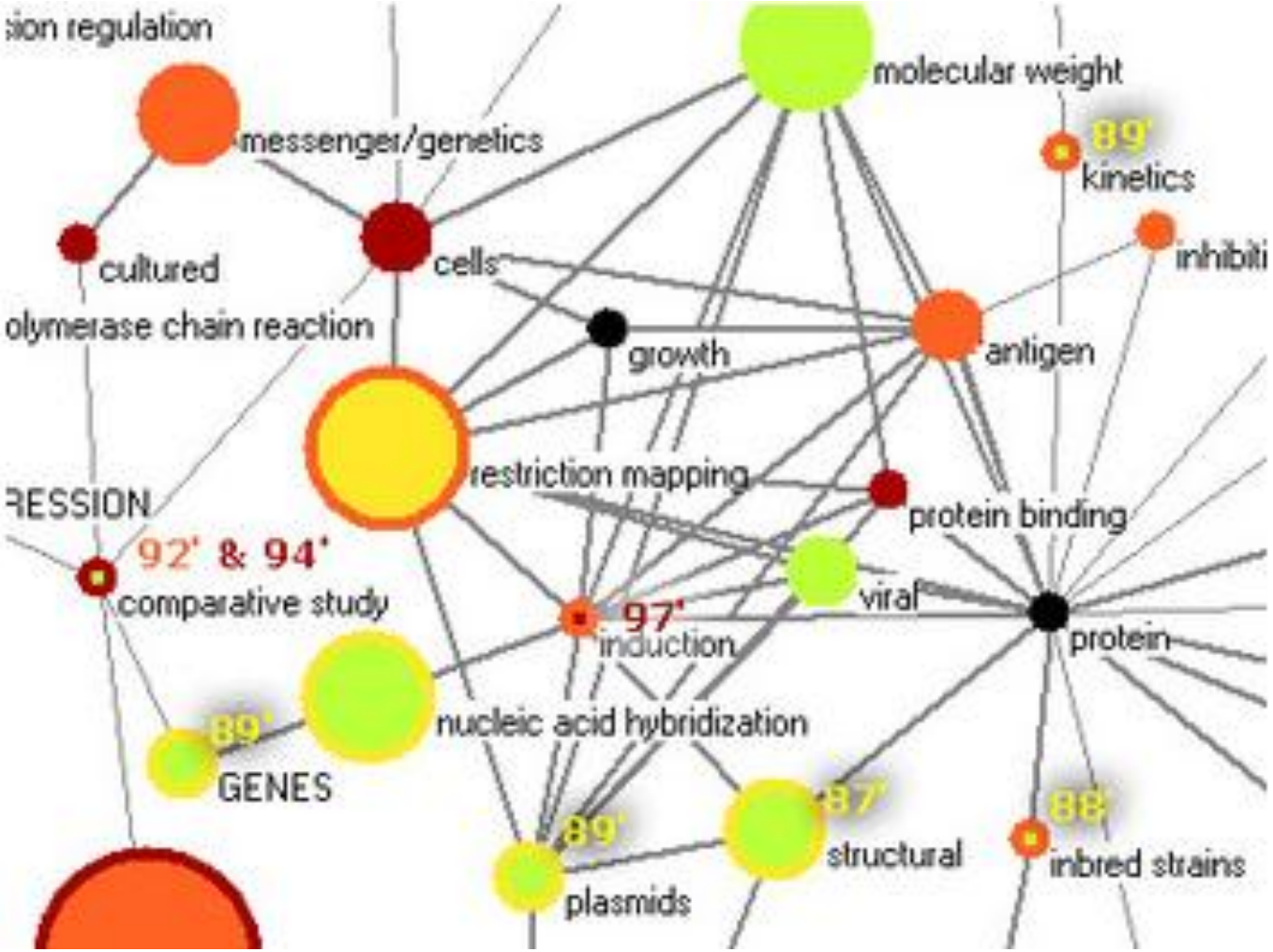
Getting to one button deploy using KANBAN

by Dominica DeGrandis
twitter: dominicad
email: dominica@djandersonassociates.com

“So – just when **can** we upgrade SQLServer on the DB servers?“

Crazy complicated DB system

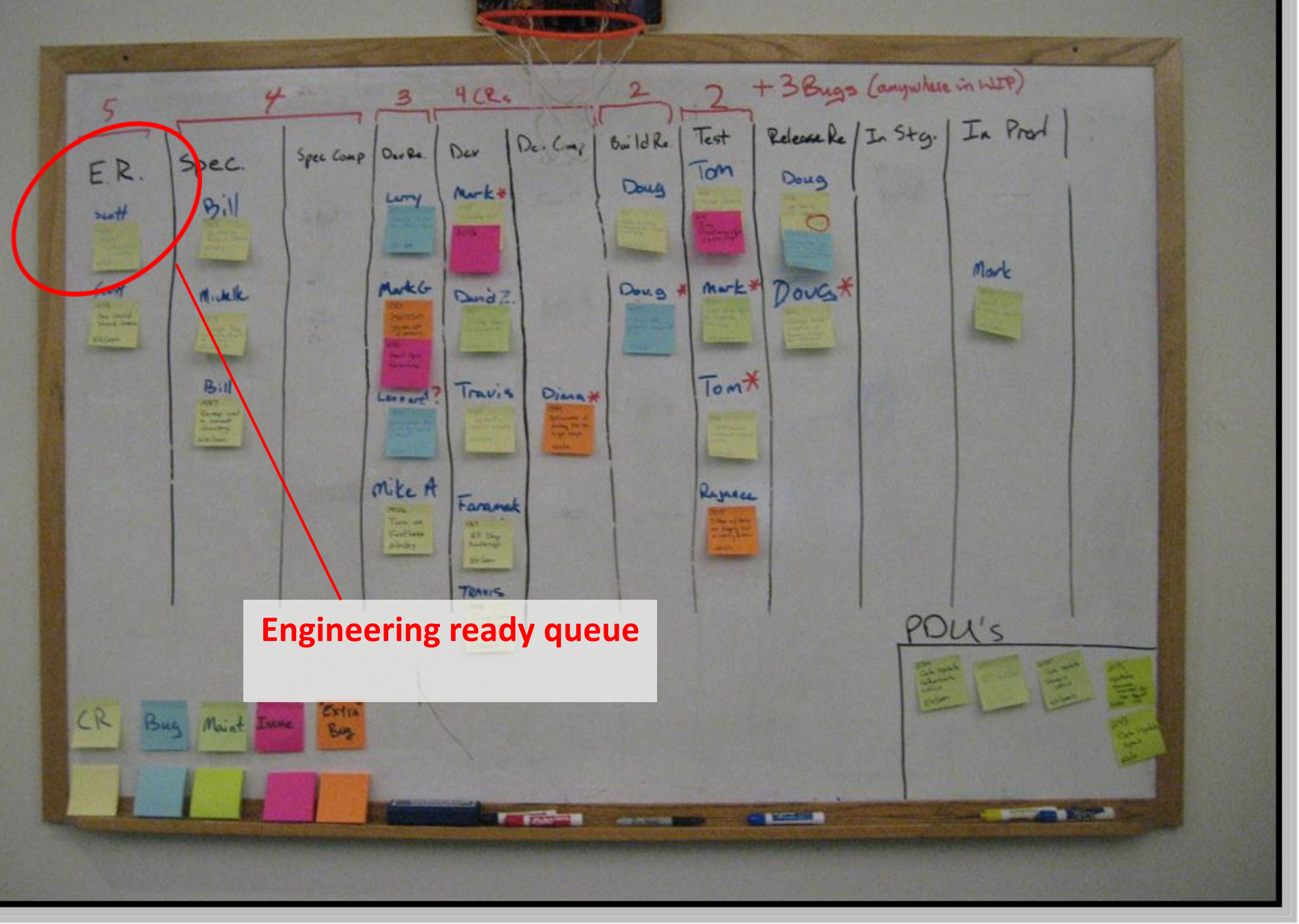


Off-the-shelf software customized to the point of no return

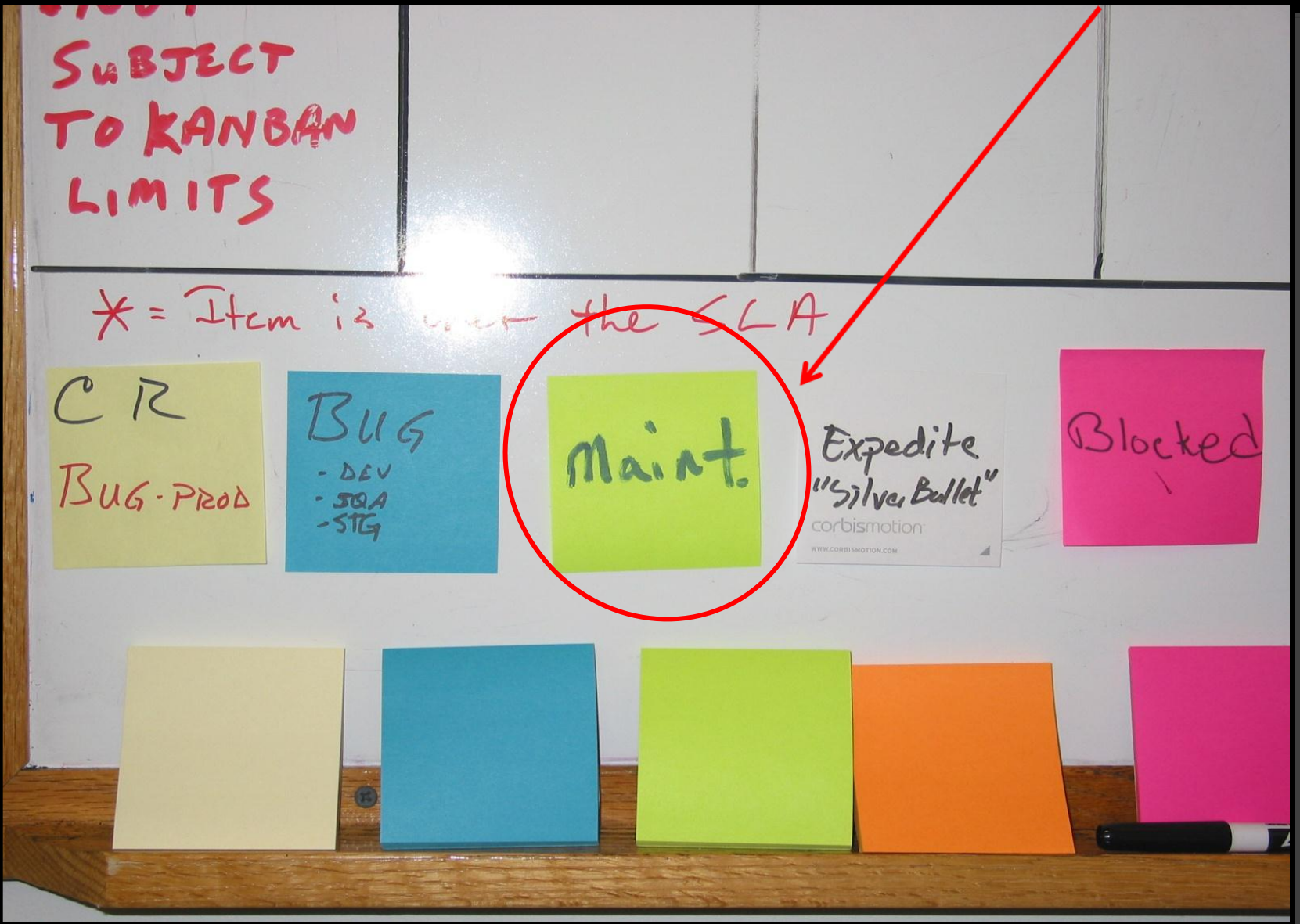
“What does it take to get recent data in test?”

“Don’t hate me, but…
the data is out of sync in test!”

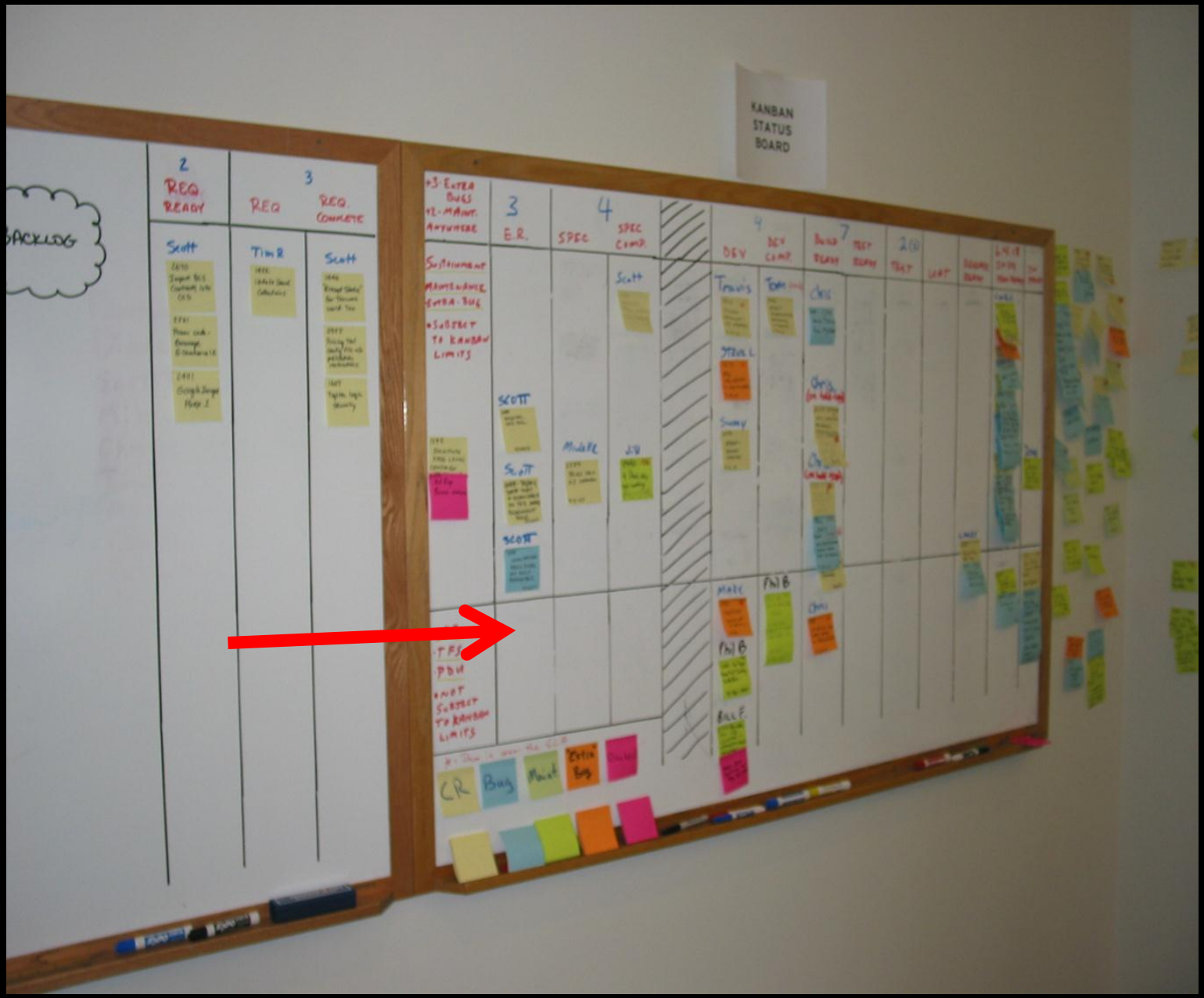
"Actually, mine should go next!"


5
4
3
4 CRs
2
2
+3 Bugs (anywhere in WIP)
E.R.
Spec.
Spec Comp
Dev Re.
Dev
Dev Comp
Build Re.
Test
Release Re
In Stg.
In Prod
Bill
Bill
Mark*
Mark G
Travis
Mike A
Faramak
Travis
Diana*
Doug
Doug*
Tom
Mark*
Tom*
Rajanee
Doug
Doug*
Mark
PDU's
CR
Bug
Maint
Issue
Extra Bug
Engineering ready queue

New intangible work item type created called “Maintenance”



Swim lane created for maintenance work

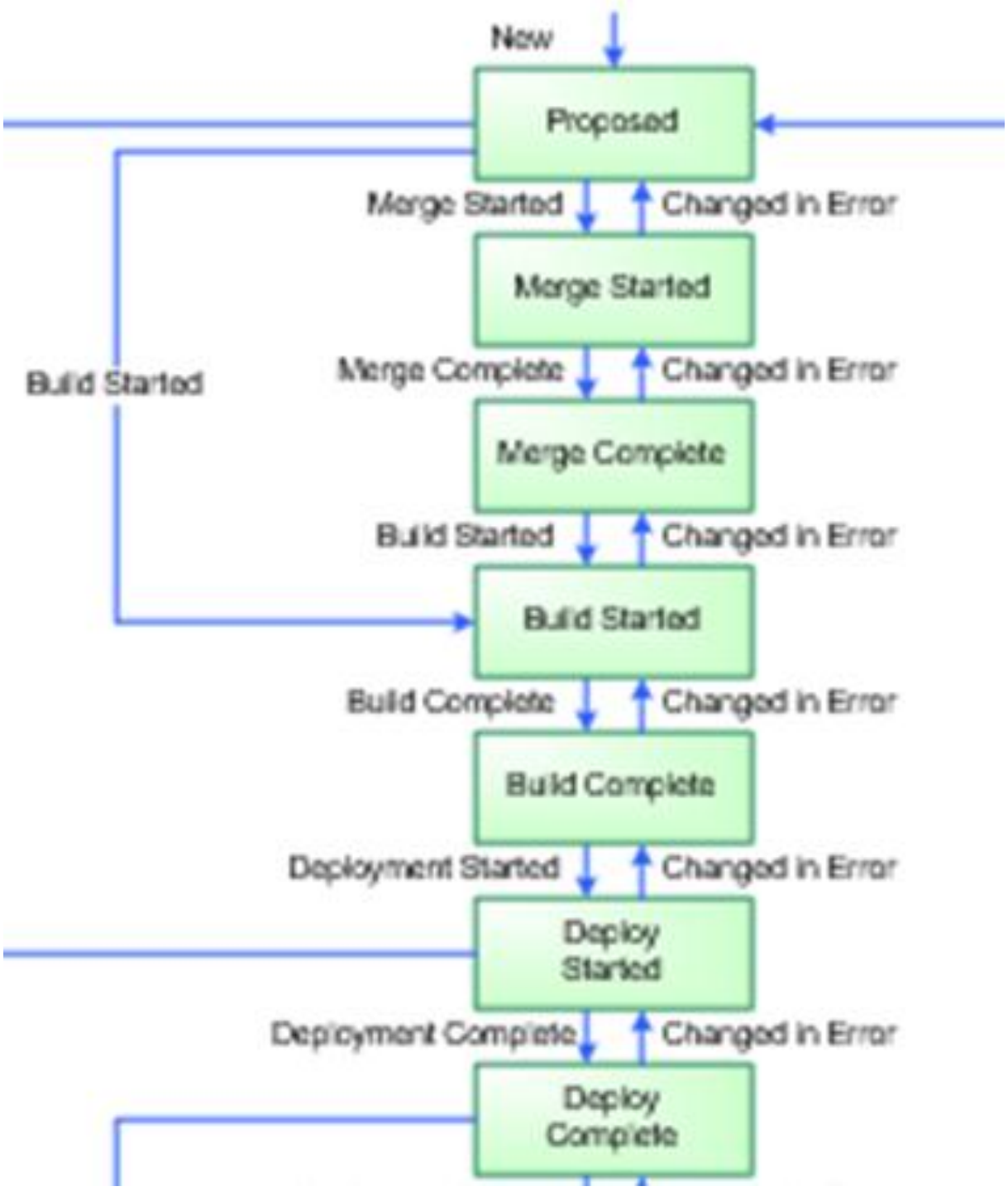


“Dude, It’s really not a configuration issue.”

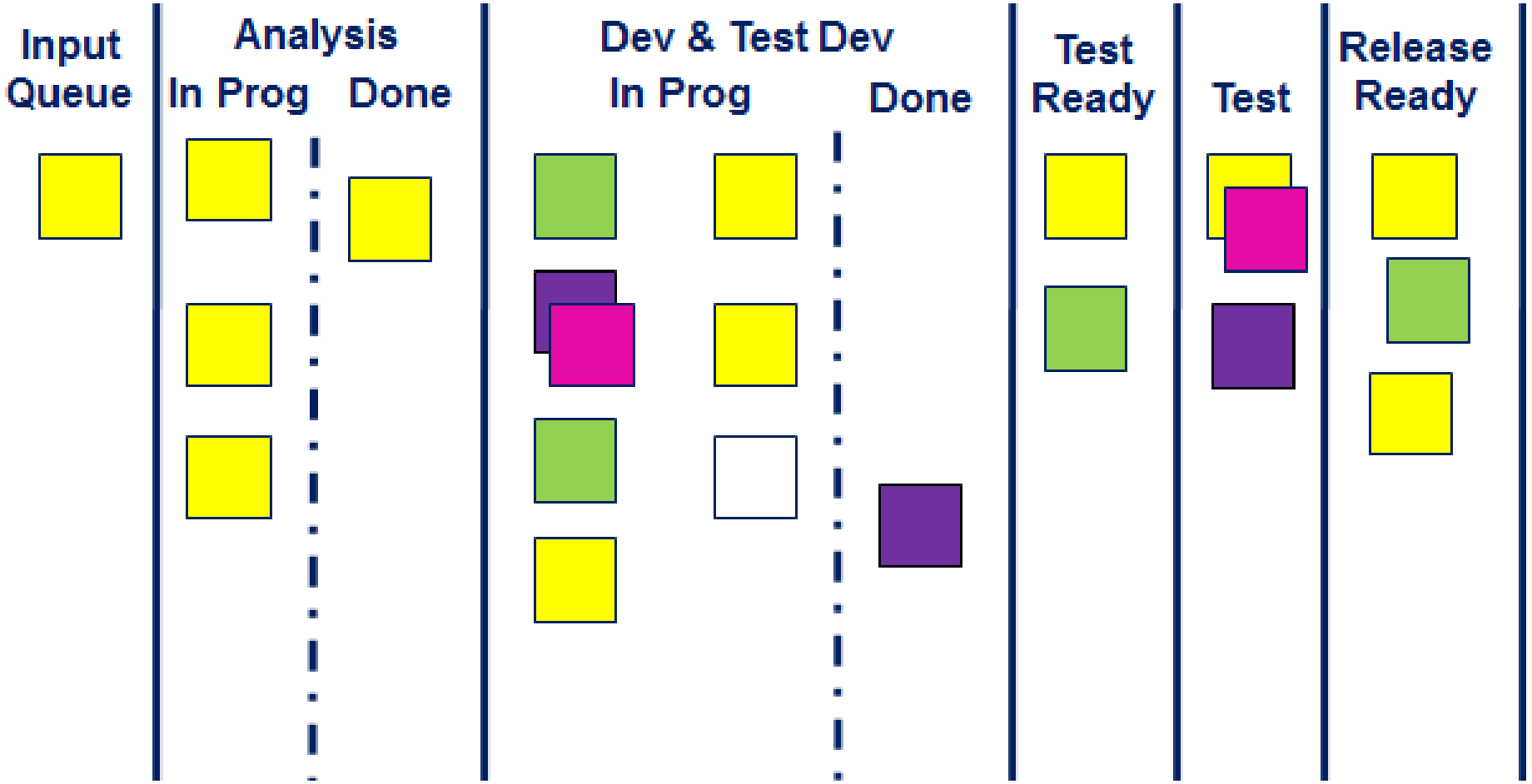
David Anderson comes to Corbis

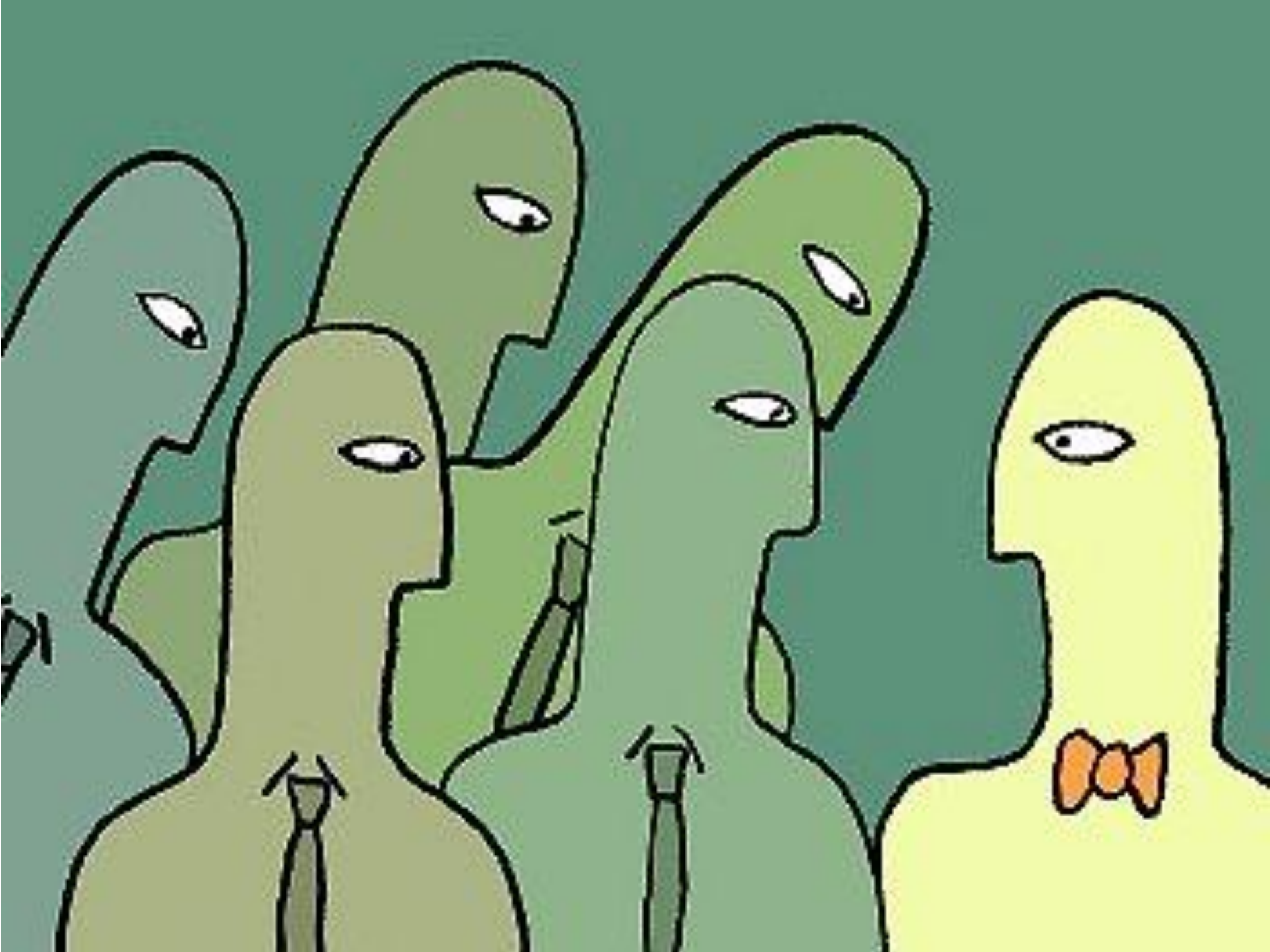
Featured Driven Development (FDD)
Eli Goldratt's "Theory of Constraints"

Configuration Management state model

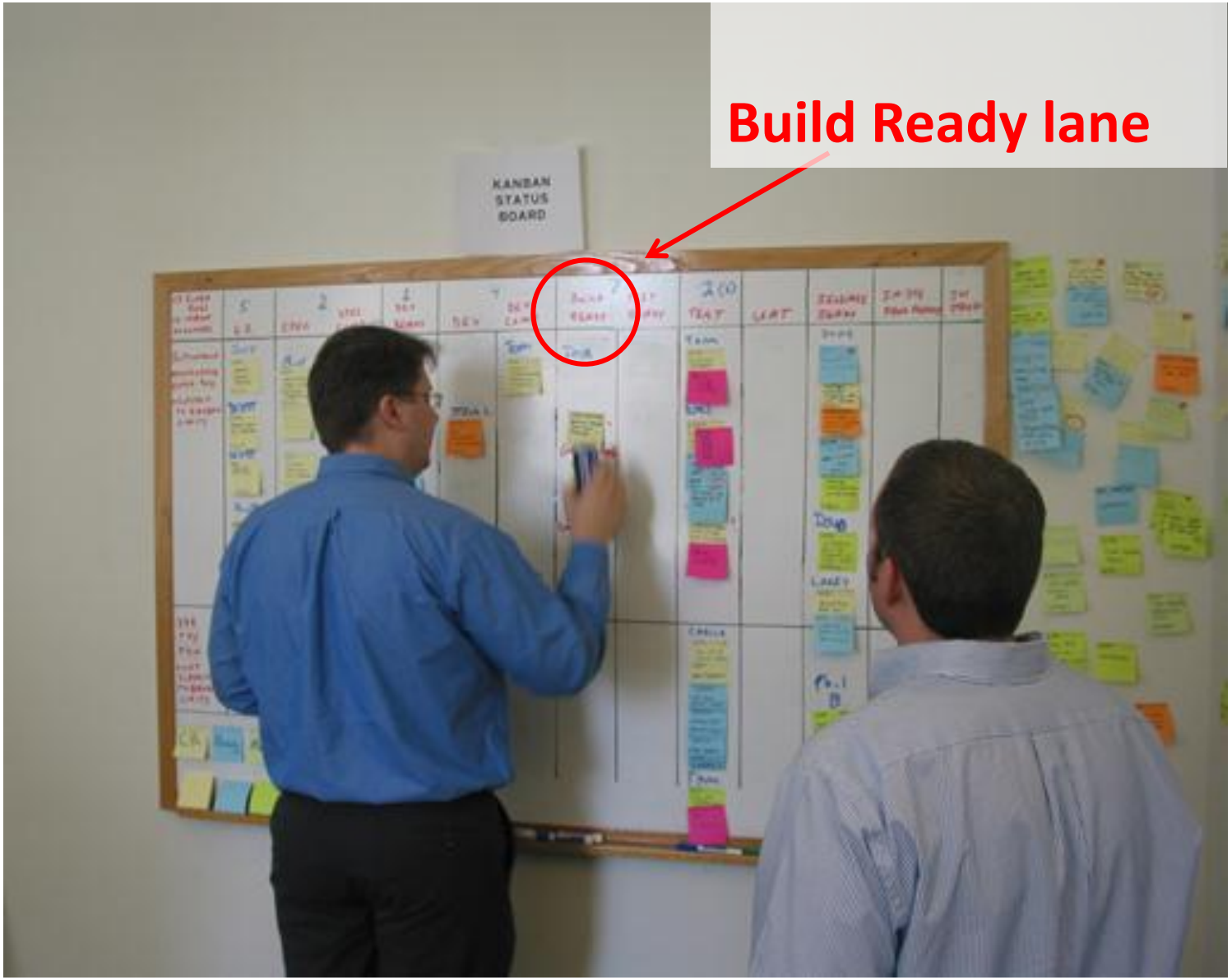


Um, where is Configuration Management?



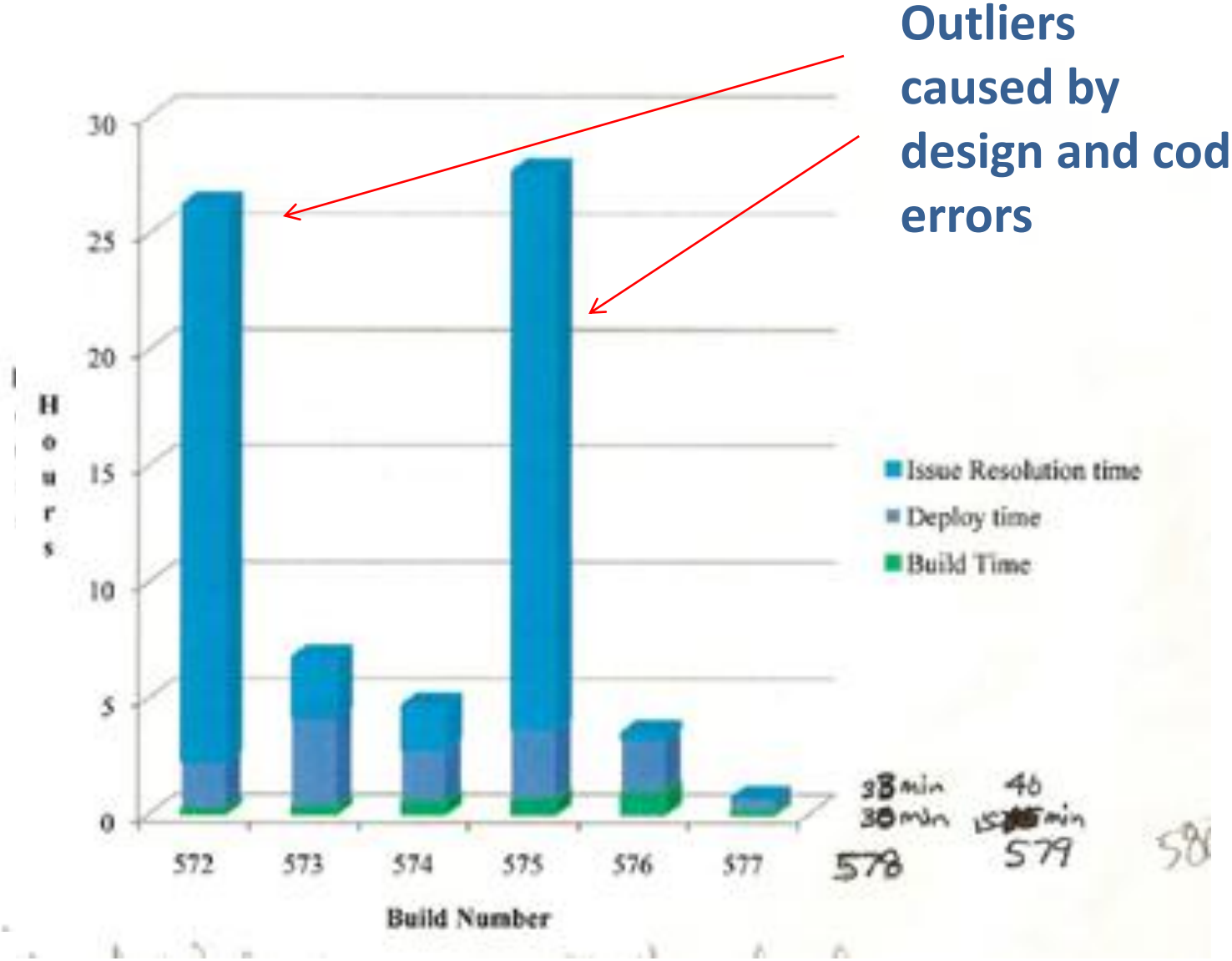


Configuration Management part of the team



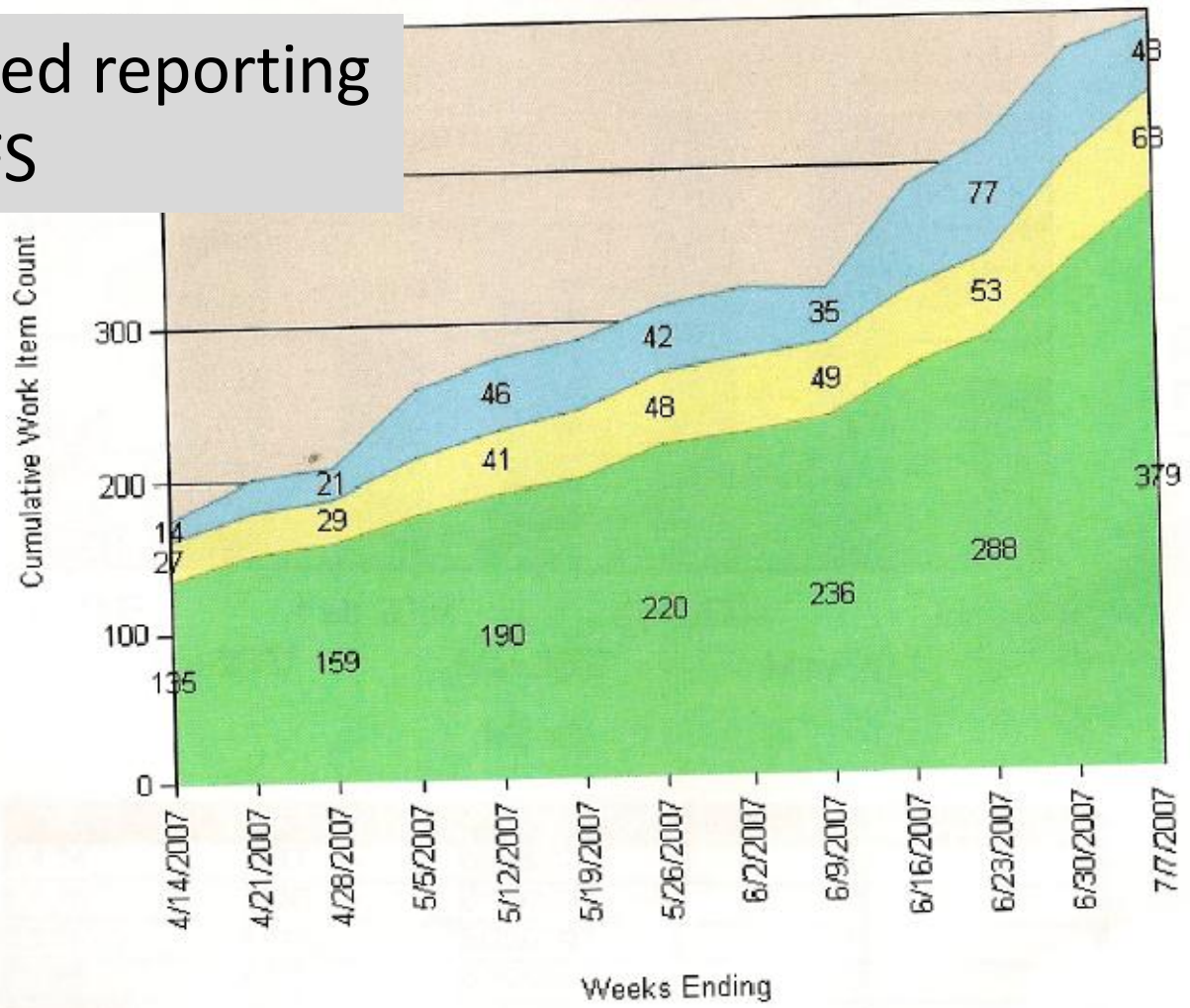
Configuration Management website

Builds don’t take that long



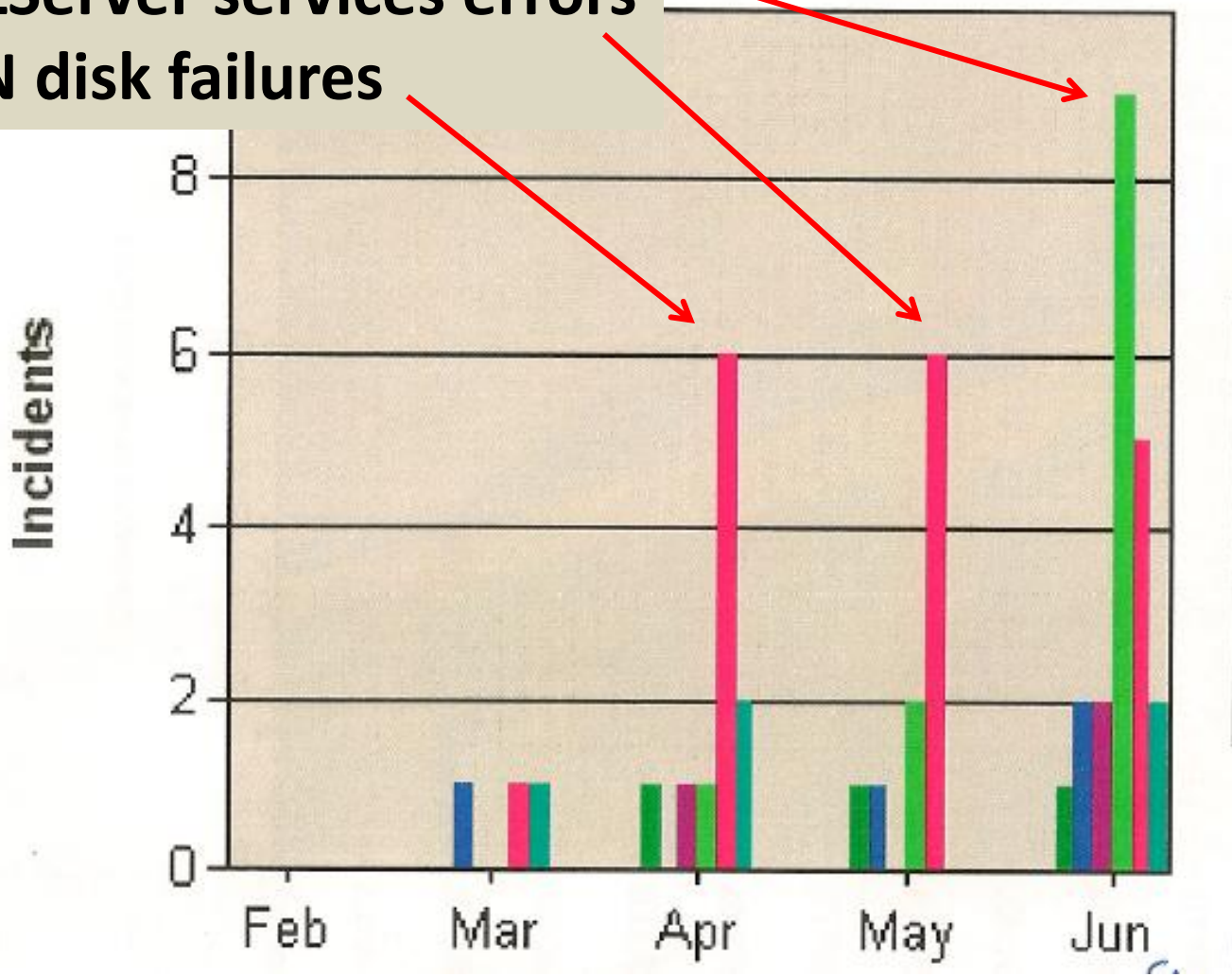
Configuration Management: Work In Progress spiked up in June

Automated reporting out of TFS



Configuration Management: Environment Reliability

Data center outage
SQLServer services errors
SAN disk failures



Configuration Management: Environment availability increased.

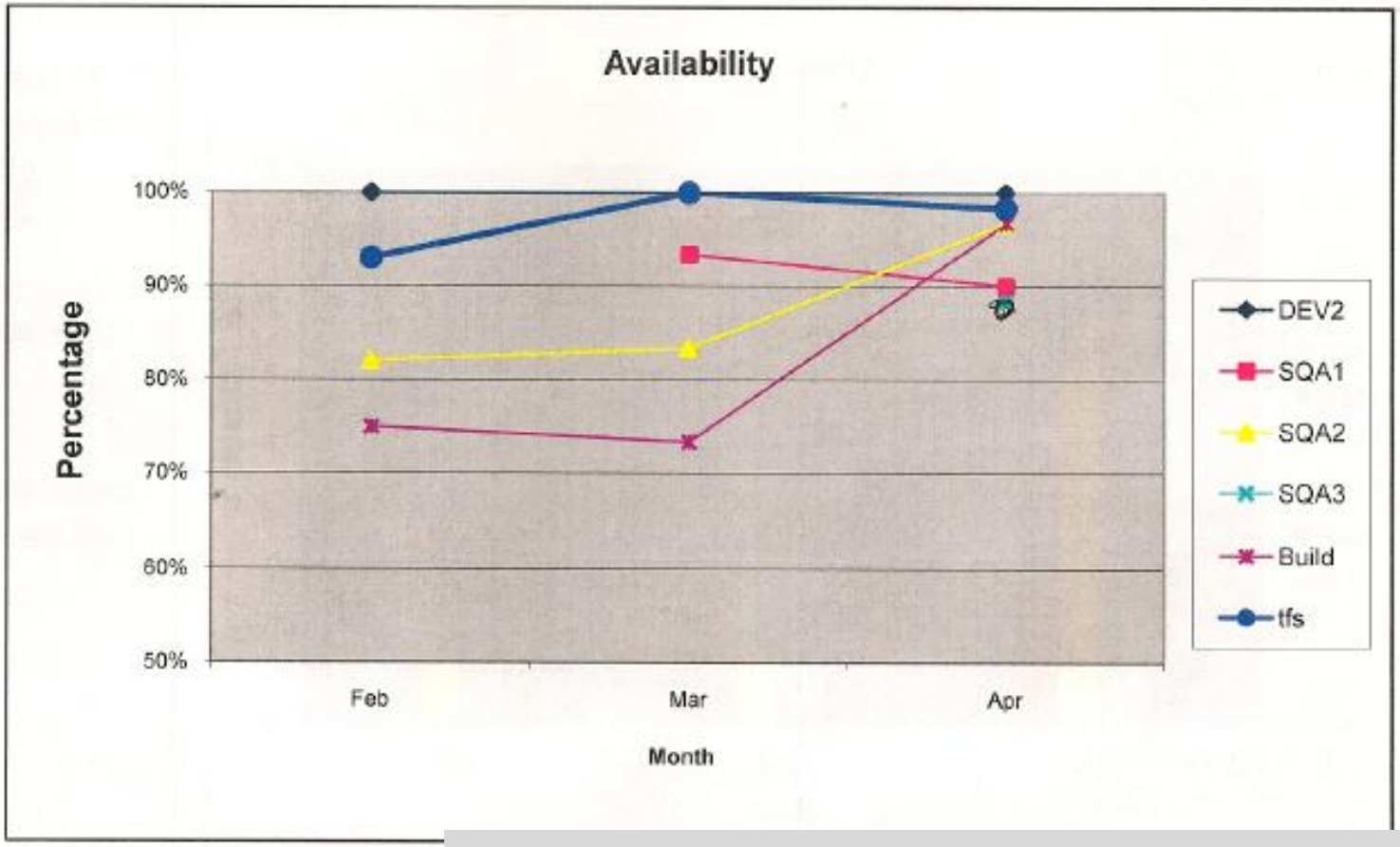
Added SQA3
Insufficient disk space on sql servers.

DEV2 Insufficient disk space on sql server.

SQA1 – SQL server issues with failed replication and DBs not installed.

TFS – bad switch port caused connection issues for 90 min.

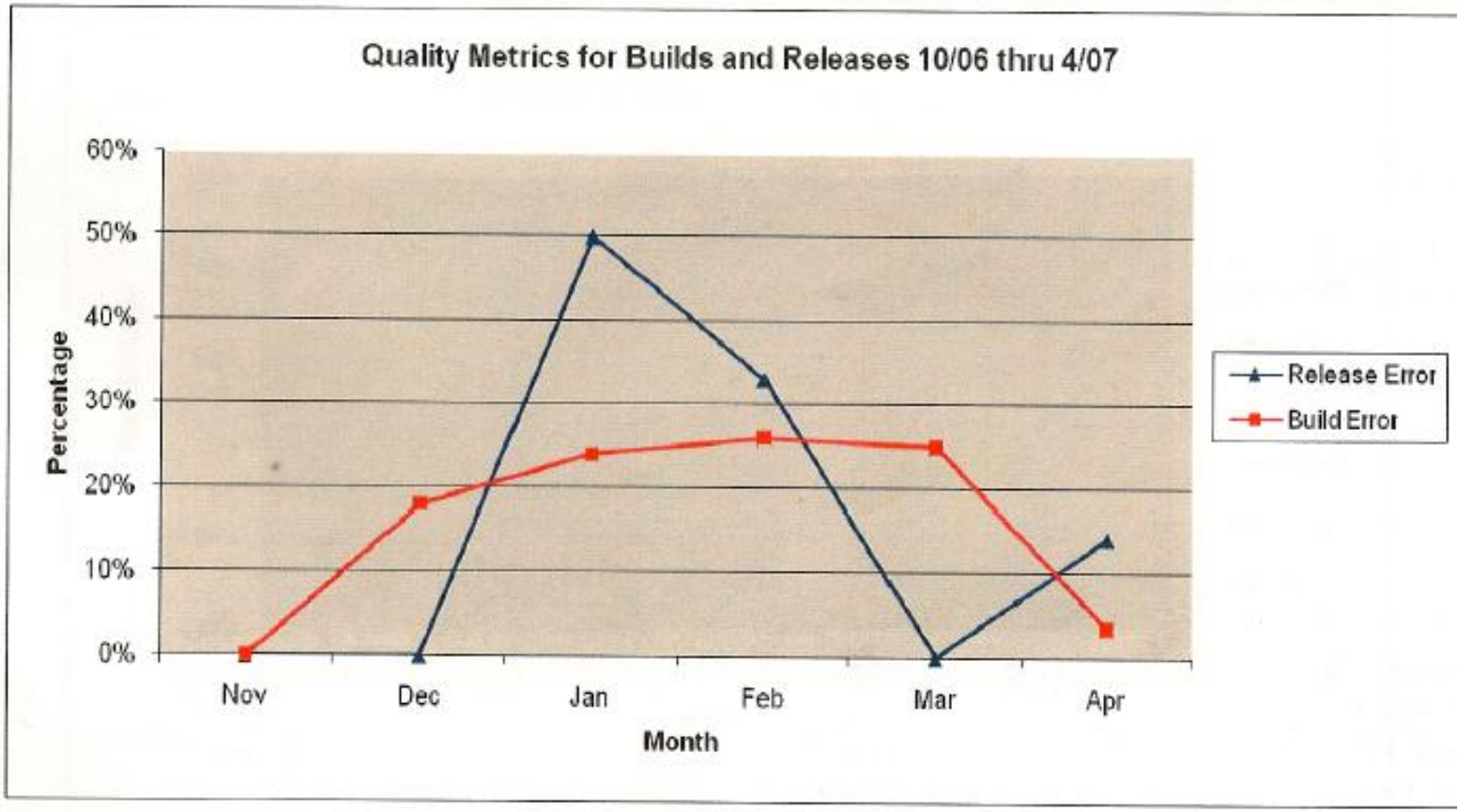
Build box – insufficient disk space



Manually created slides may seem like a burden, but resulted in us getting help and budget.

Configuration Management: Build Quality is improving but Release quality is not.

- **Production release error**: ExternalUI step missing in base install doc causing website to not run http compressn.
- **Merge issues**
  Shelvesets in wrong codeline, mi ssing shelvesets
  unresolved conflicts

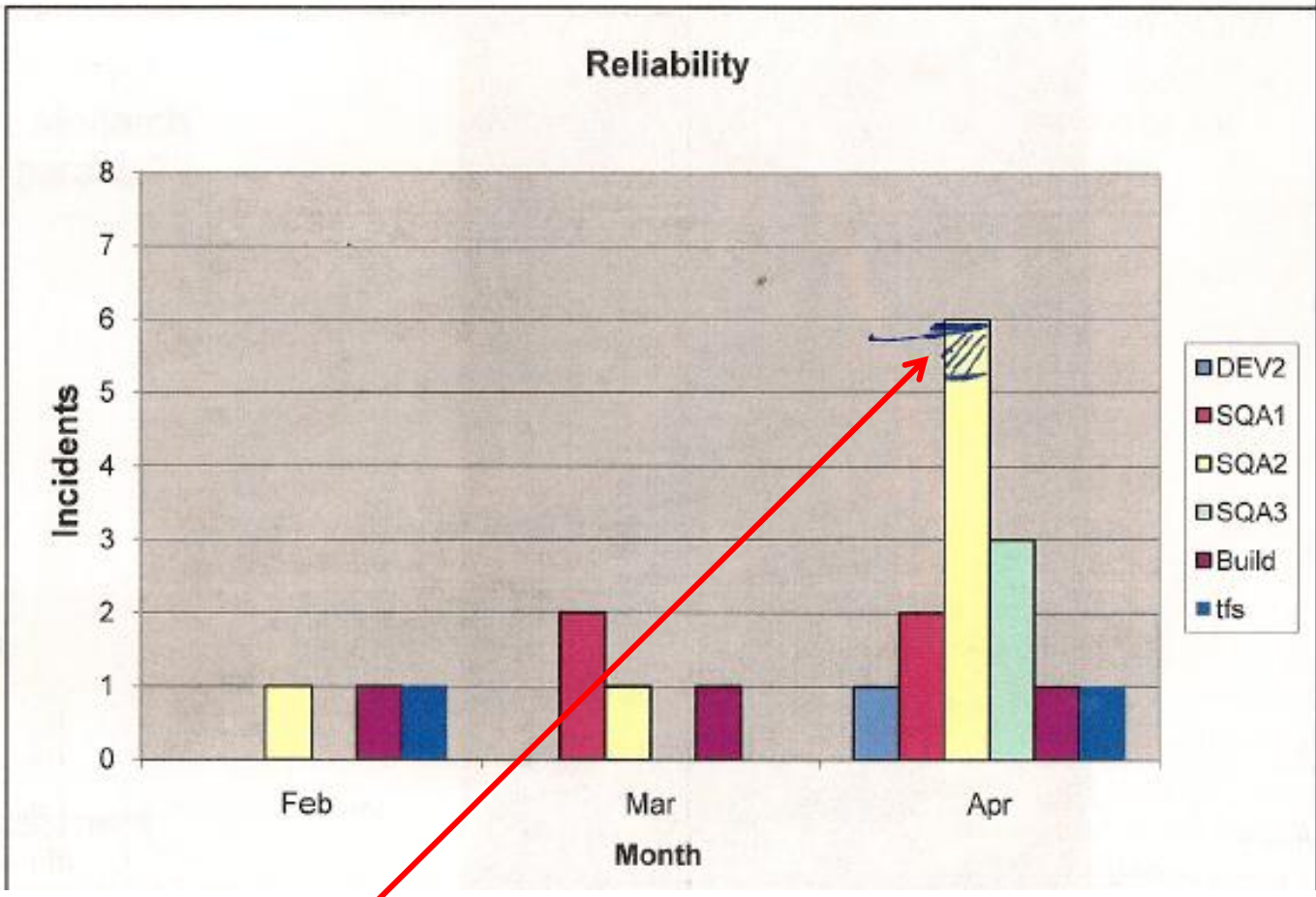


Configuration Management: The number of environment incidents increased.

Increased incidents due to capturing data from sqa3 and the team actually logging issues in TFS.

SQA3 – Server issues with disk space

SQA2 – Environment issues with pools, permissions and config settings.



Configuration issue in web.config

“um, It’s Thanksgiving Eve
and you want me
to do what?”

“We would be hosed without **you**”

The Night Owl

Positive
Punishment?

Quality doesn't
happen on
5 hours of sleep.

Bi-weekly release cadence
Overtime decreased
Heroic efforts not needed anymore.

Production deployments over time evolved into a non-event.

Other contributors:

Development standards included backwards compatibility.

The test team began to accept latent code patterns (“feature switches”).

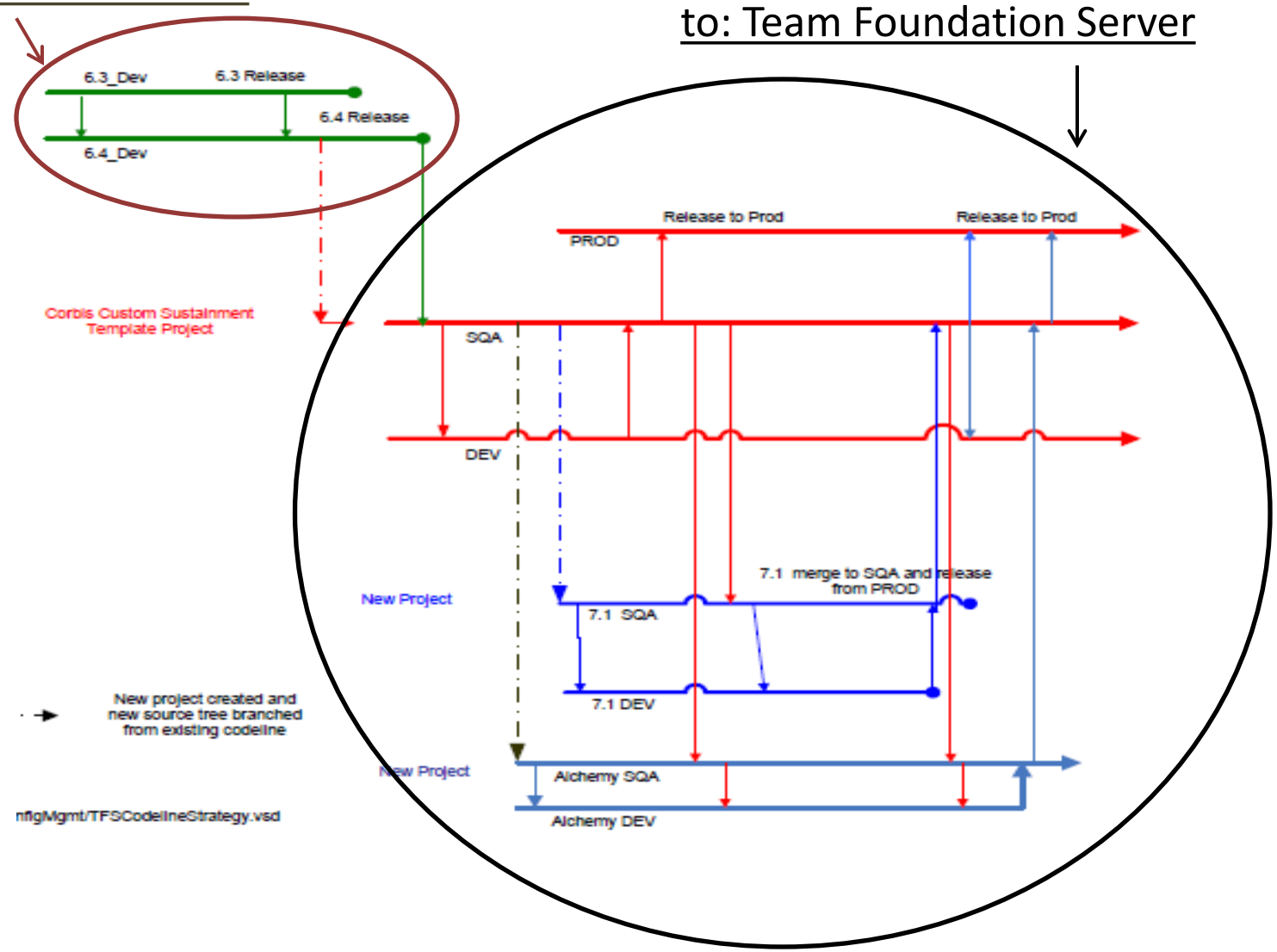
Merge issues became less frequent.

“M” is for Merge

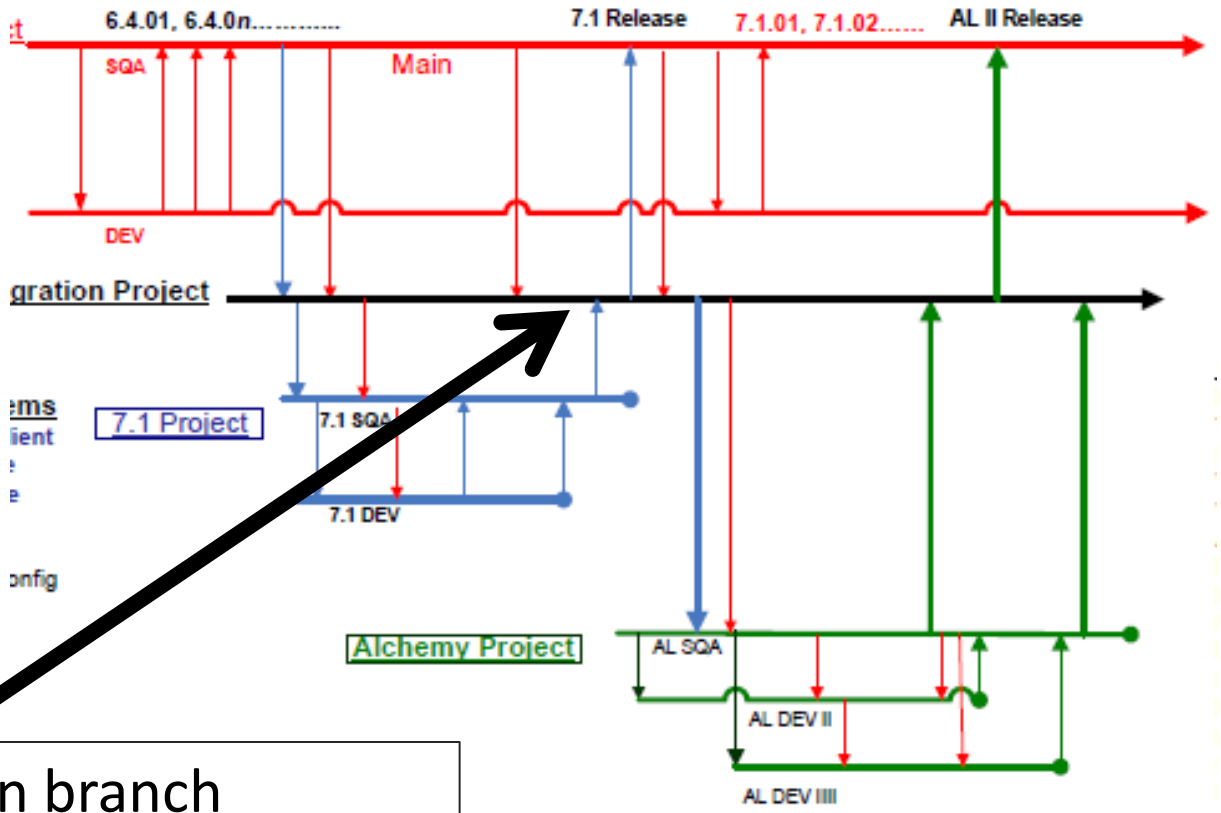
Codeline Migration Strategy - Oct 2006

from: SourceGear Vault

to: Team Foundation Server



Codeline strategy - Dec 2006



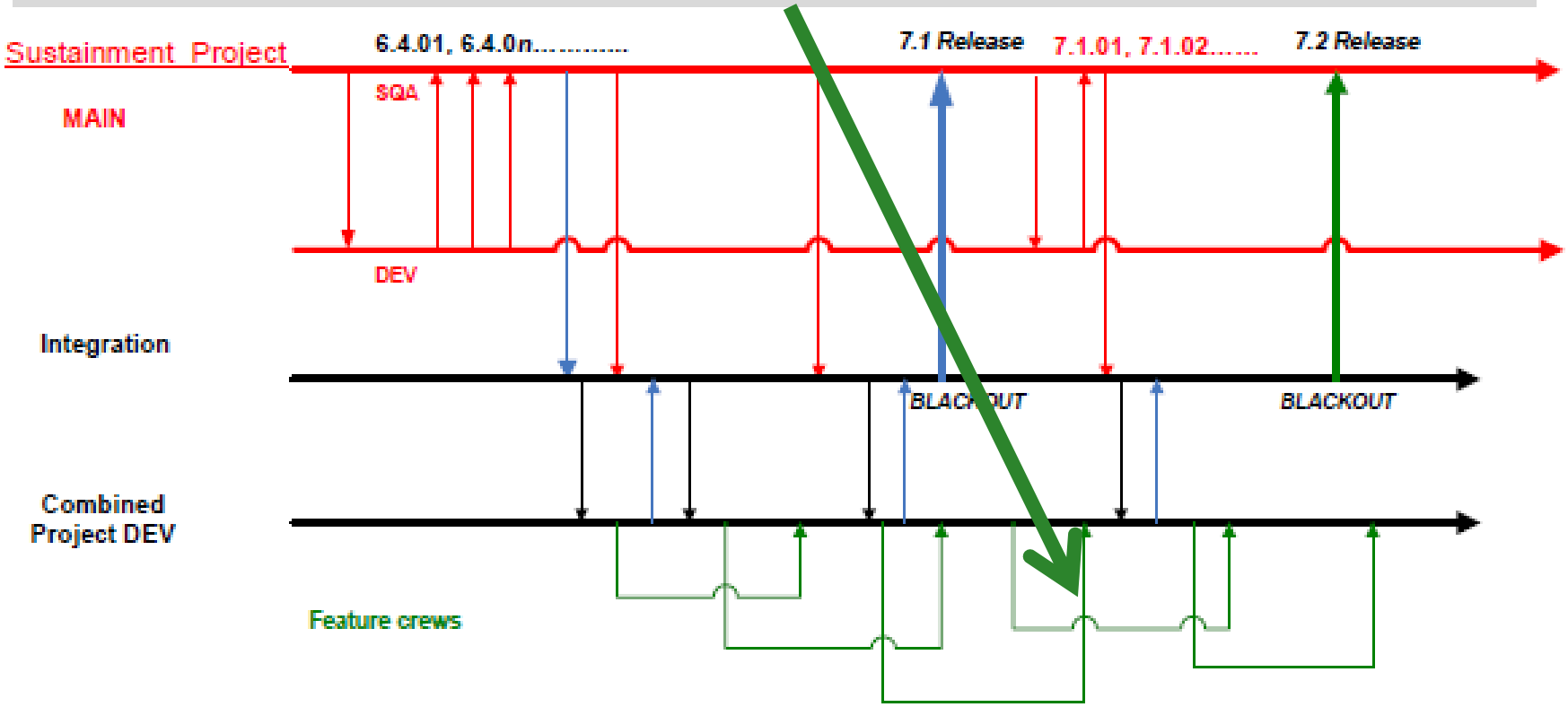
Integration branch concept emerged to avoid baseless merging in TFS

Codeline strategy - Jan 2007

Major Merge issues reported at Ops reviews

Merging consumed more capacity than development.

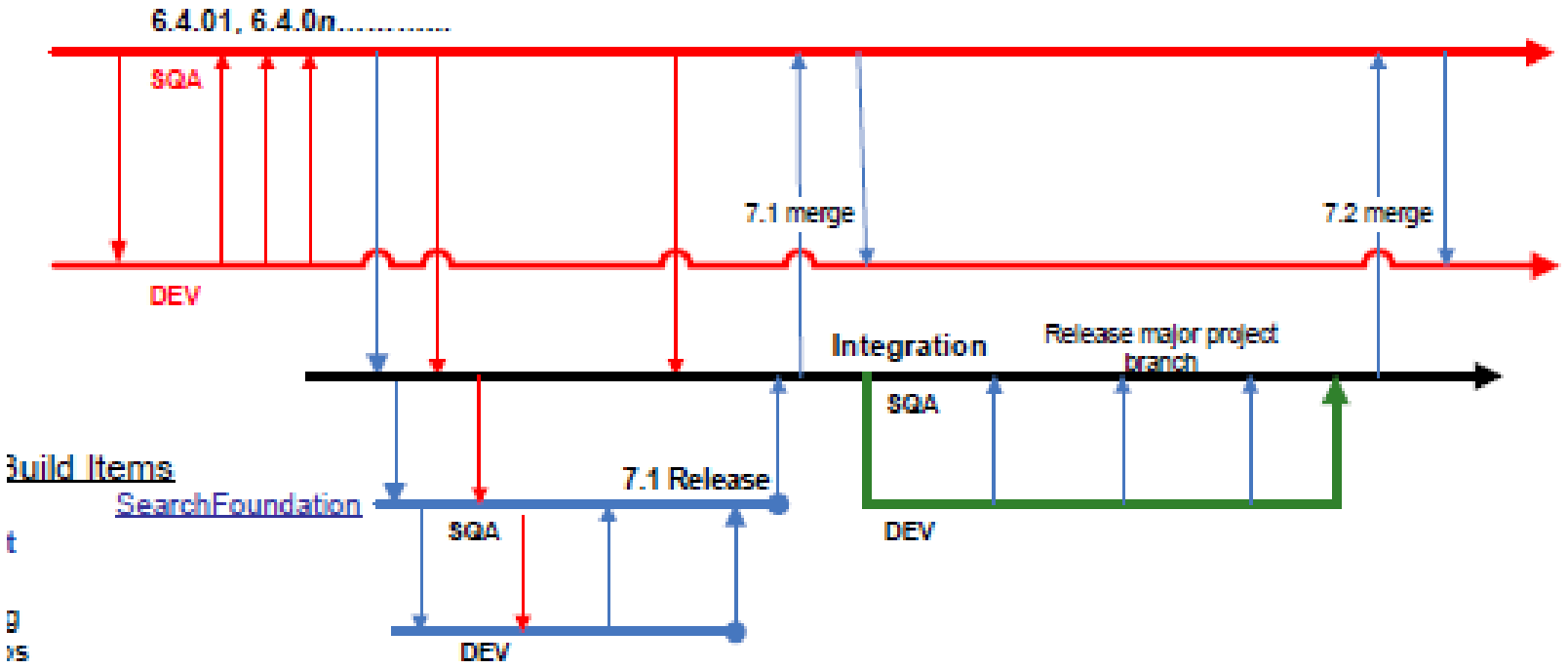
Experimented with Feature crew branches for large project work all branched from combined dev branch but folks were uncomfortable with it.



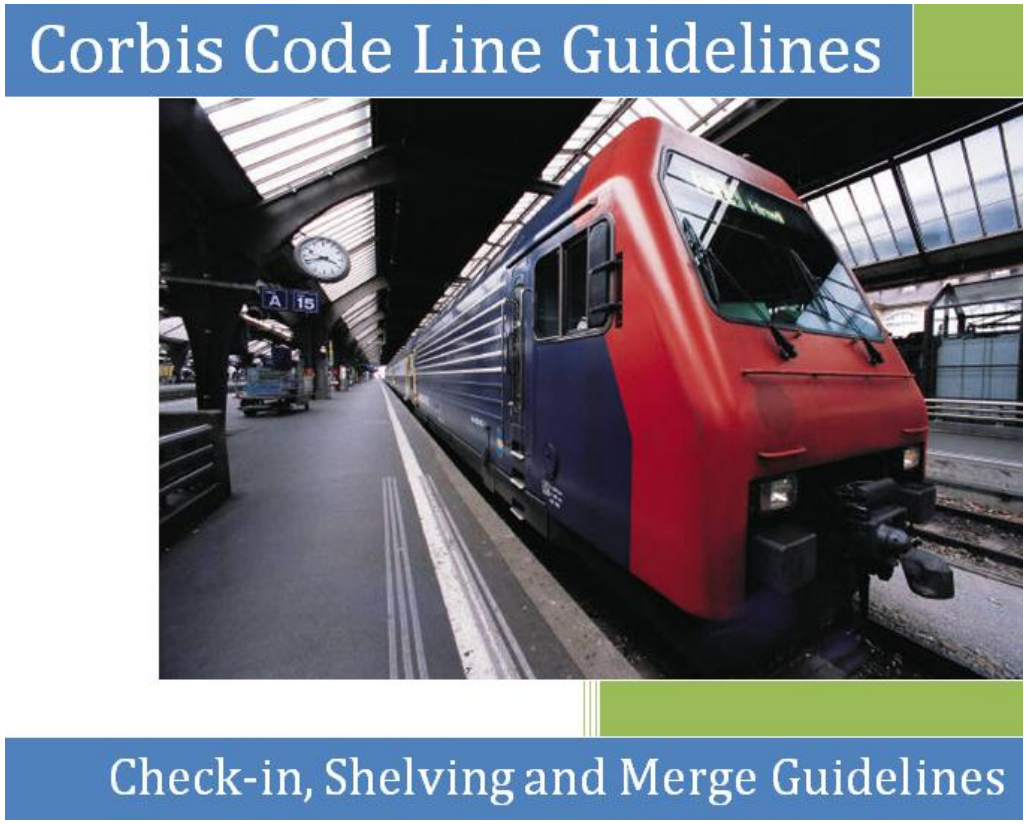
Codeline strategy – July 2007

Major Merge issues reported at Ops reviews

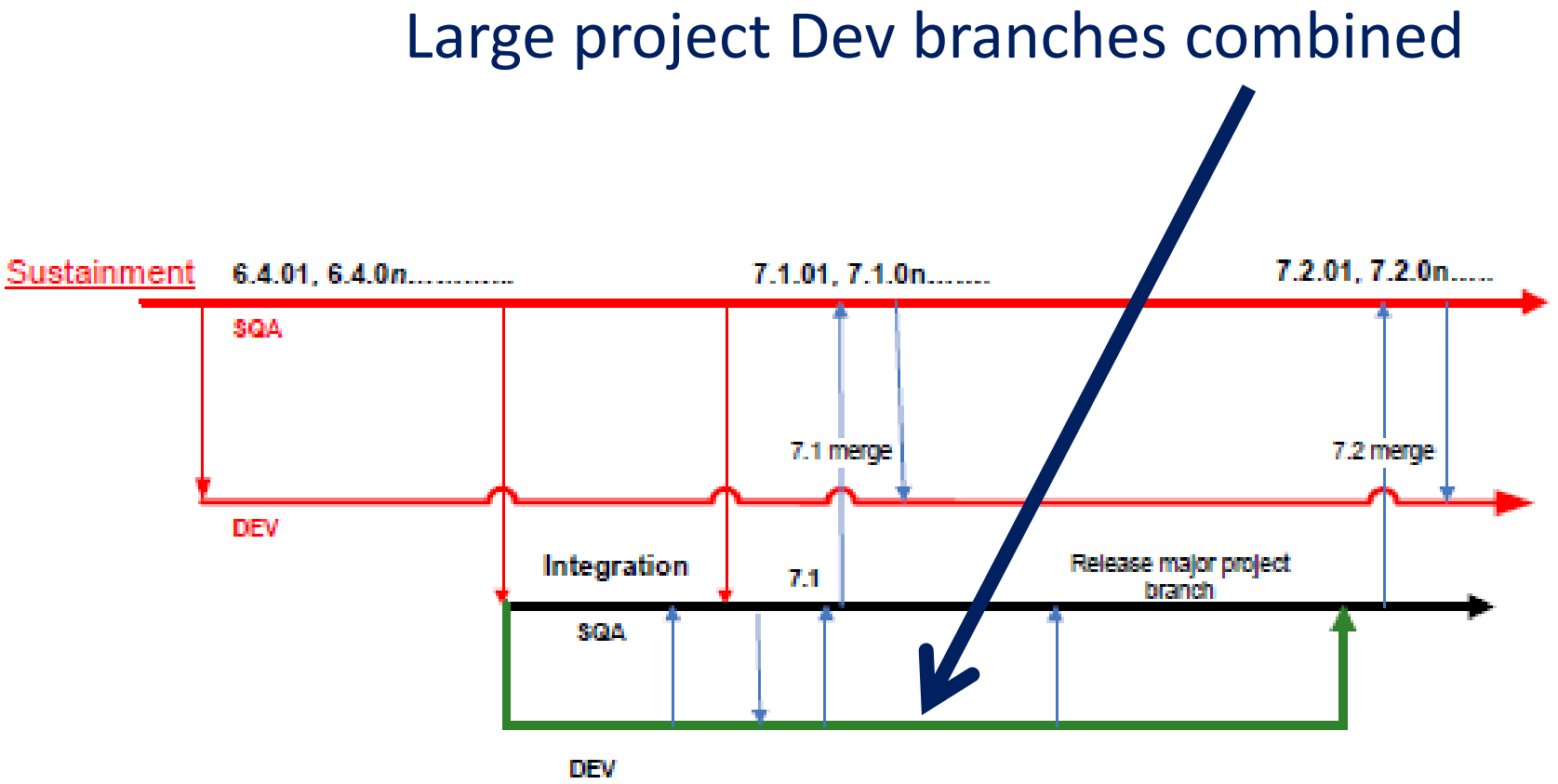
Merging consumed more capacity than development.



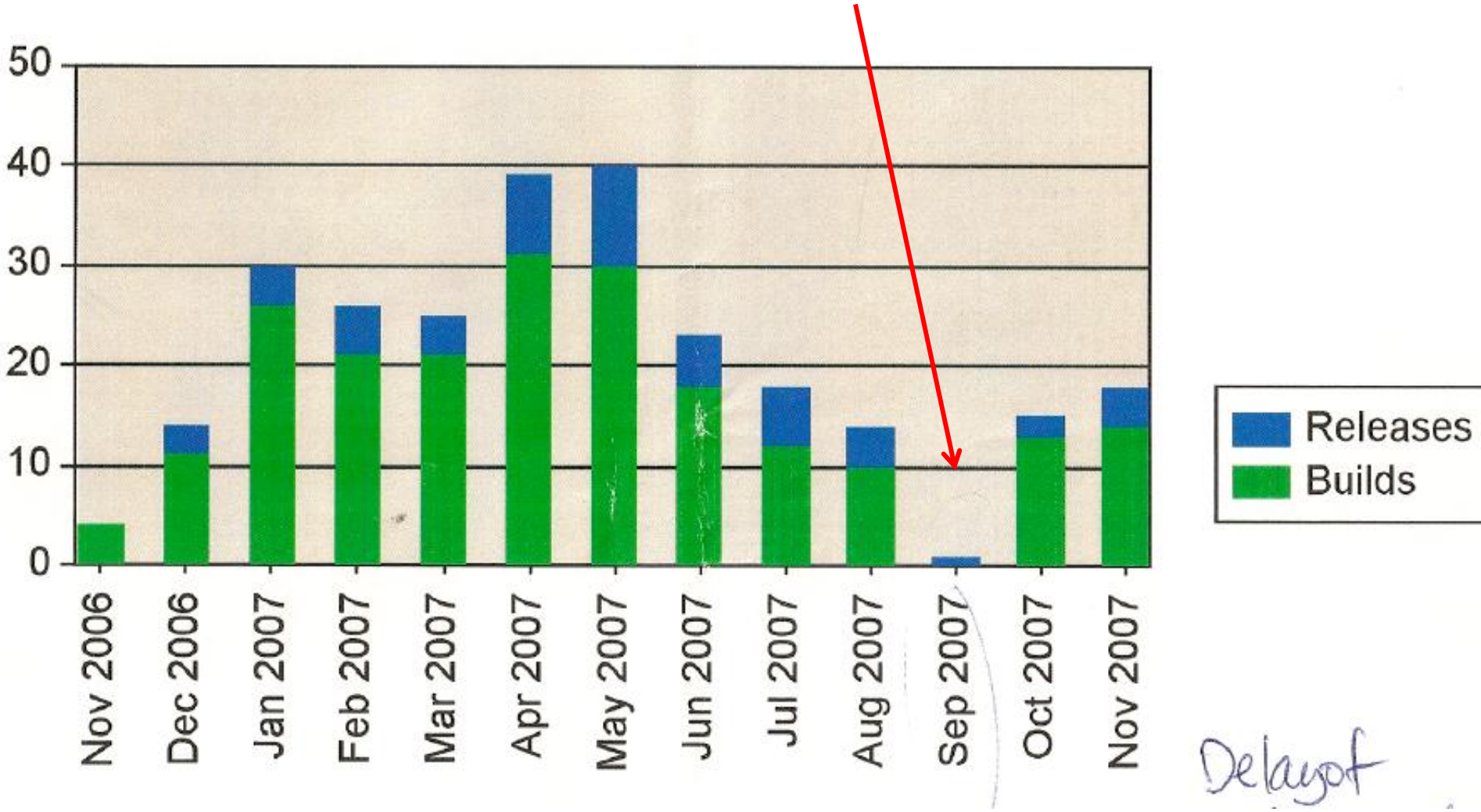
Merge Policy documented but not made explicit enough



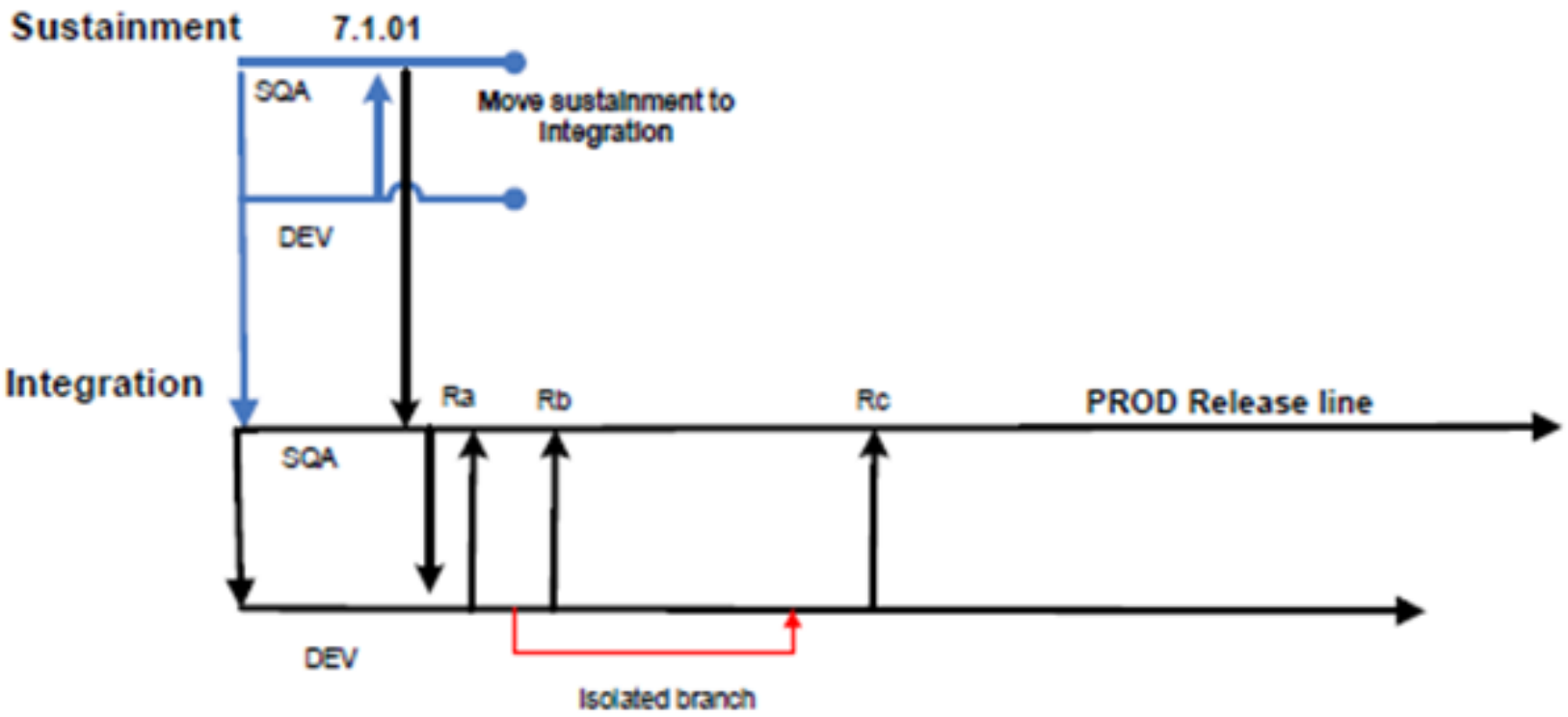
Codeline strategy - Sept 2007



Sustainment releases stuck waiting a release.
The approach used to branch the code prevented check-ins until the major project release was deployed.



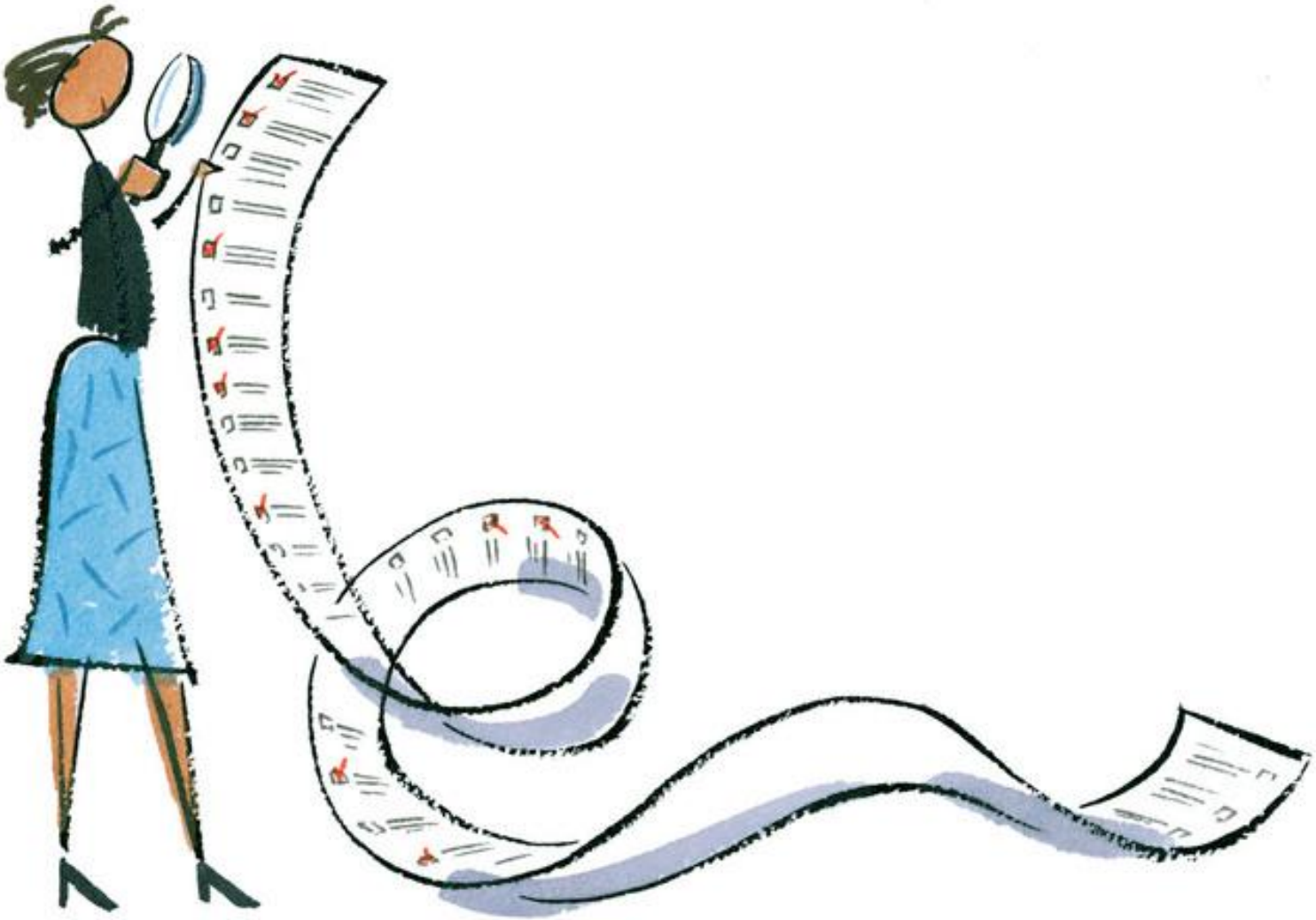
Codeline Strategy – Oct 2007



Those tricky environment specific config files

Production Release Issues log

RRT 8.1.18 - Prod release error due to password error in MediaSuite WCF service for Marketing collections DB connection string. This also occurred with a prior release too.”

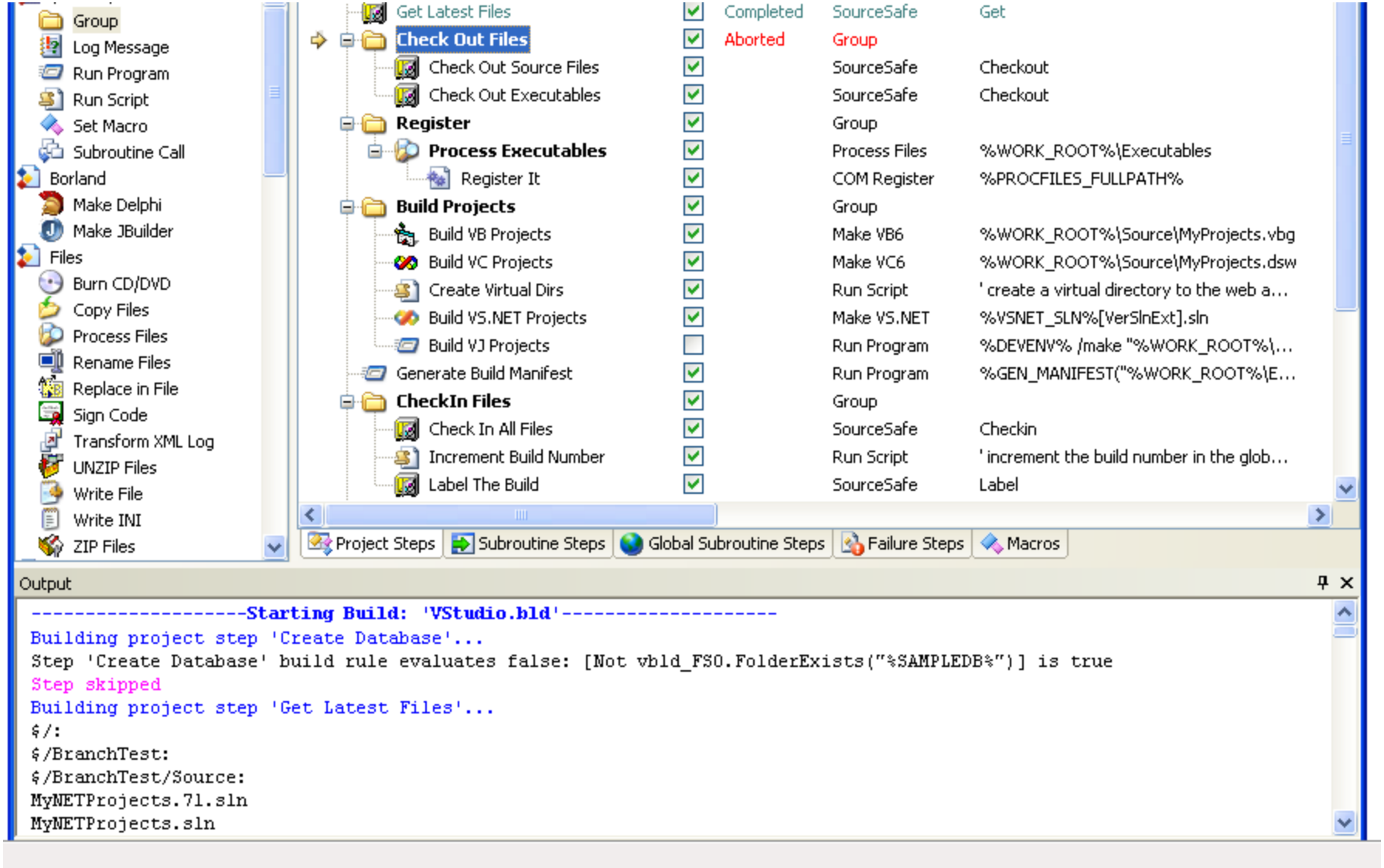


9:30 standup every morning

If interested, be there

The Transition from manually triggered builds to Continuous Integration

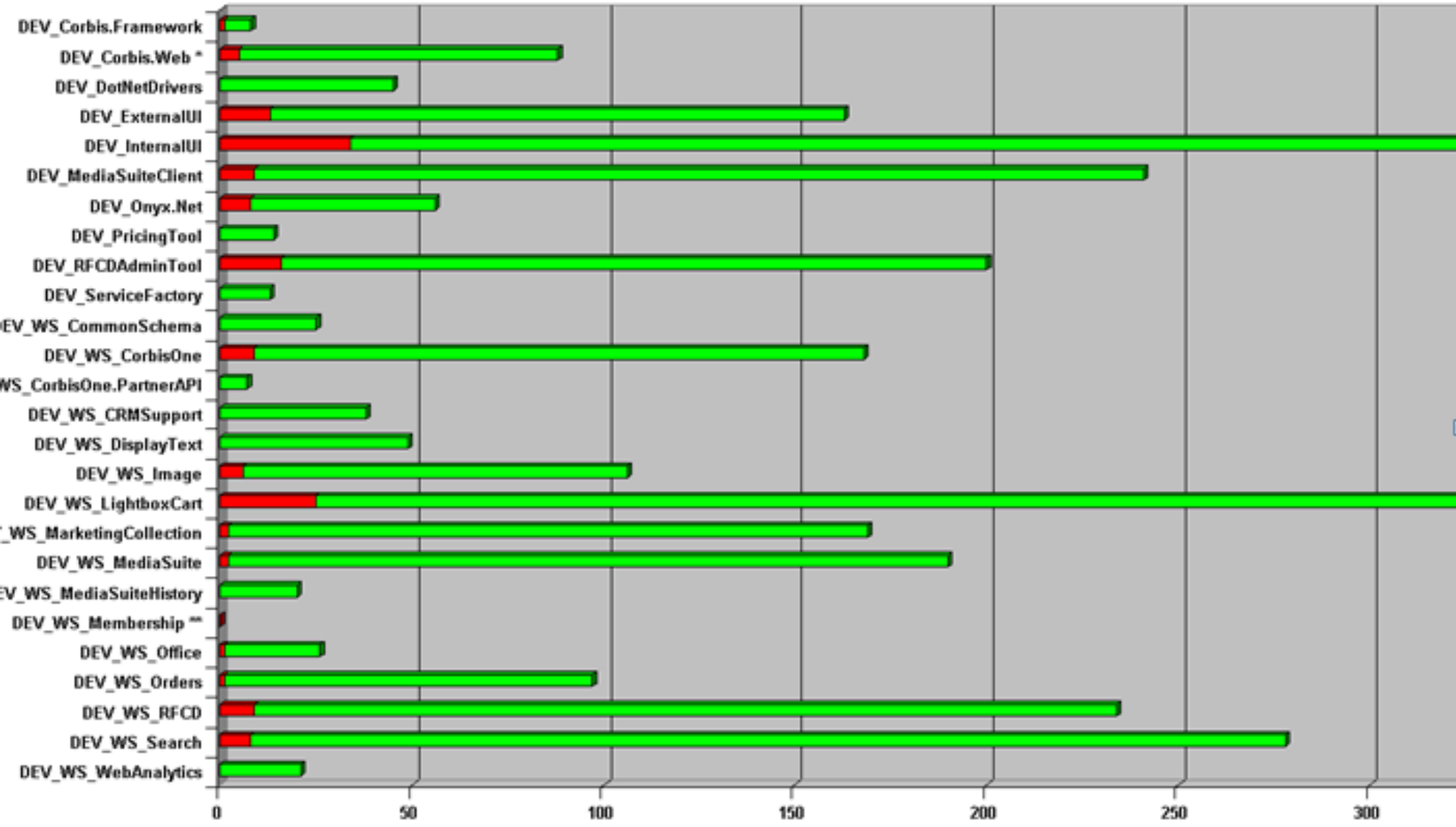
Automating builds and deployments



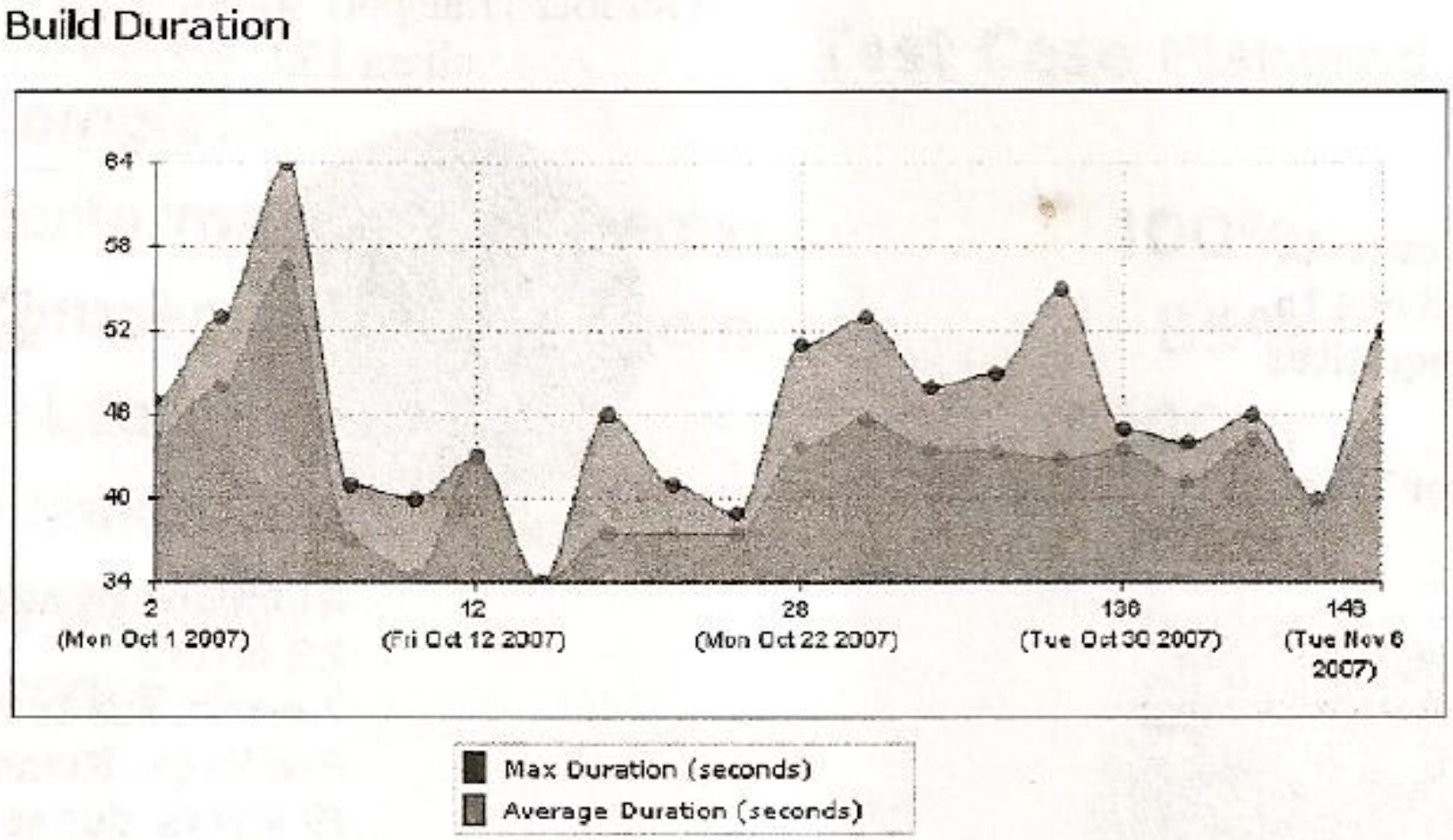
Automated testing was seen as too expensive

Someone was smart enough to collect and present some vital data on the "**cost of delay**" from developers who were unable to see the impact of their changes quickly enough.

Continuous Integration reporting

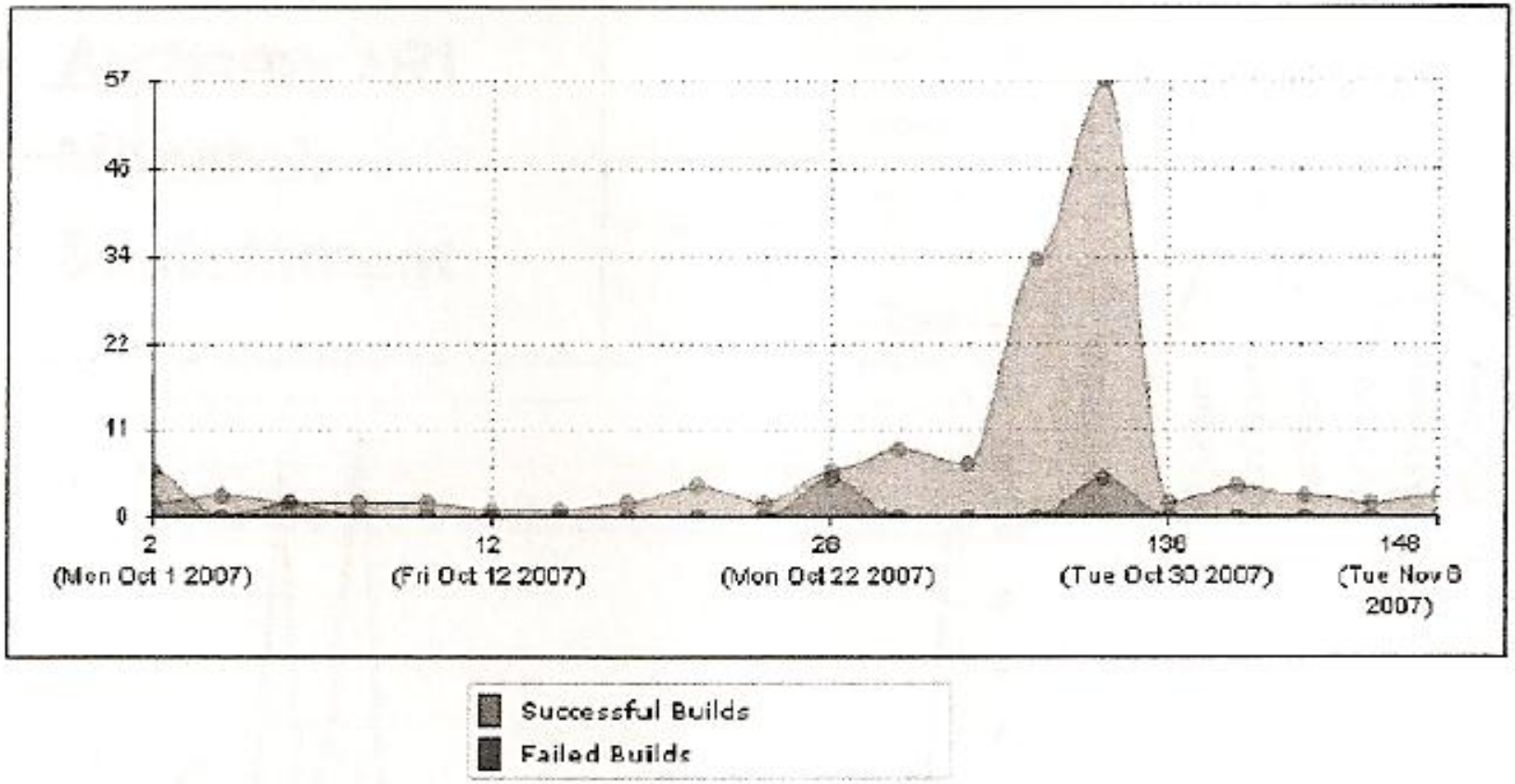


Configuration Management: Build Duration



Configuration Management: Build Quality Reporting available thru Cruise Control

Build Report



“What would it take to get rid of
the 3 day freeze?”

All hands on deck troubleshooting for 4 days and nights resulted in a 3 day post production freeze in prod and test environments.

Joking around after release, “Remember when it took all weekend to do a release?”

no heroics needed
no overtime
no chaos

Spontaneous meeting after standup

Not breaking the WIP limit provoked conversations which led to improvement.

Configuration Management reporting demonstrated incremental improvements over time

For much of 2007,
CM reporting
consisted mostly of problems.

By 2008,
CM reporting
consisted mostly of
improvements.

Getting to one button deploy using KANBAN

by Dominica DeGrandis
twitter: dominicad
email: dominica@djandersonassociates.com
Yahoo! group: kanbanops