The Register

AGILITY WITHOUT ANXIETY

Deliver early, deliver often

In association with PERFORCE

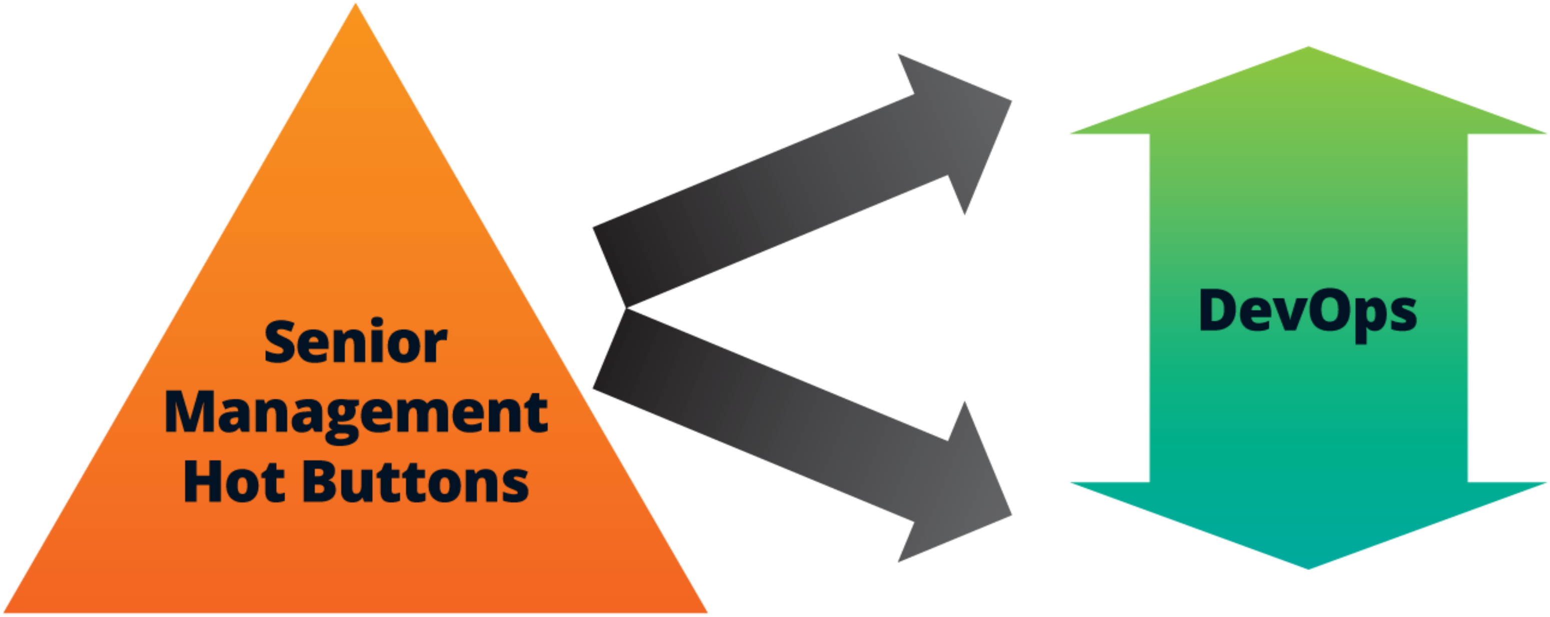
On our Regcast today

Tim Phillips	The Register
Dale Vile	Freeform Dynamics
Mark Warren	Perforce

Why are we here?

deliver early + deliver often = deliver badly ?

40,000ft context: Exec expectations



Faster delivery in line with business priorities

Implications for Development

Senior Management Hot Buttons

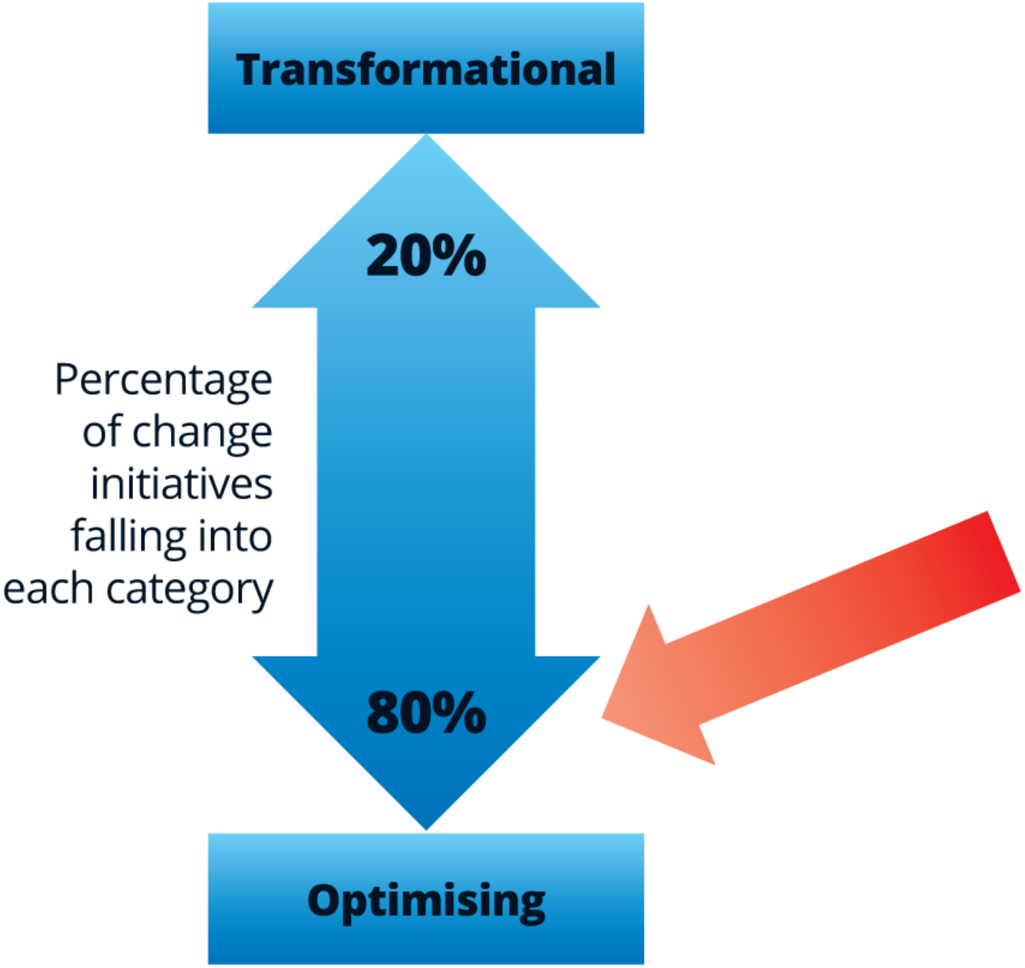
DevOps

Implications for Operations

Lower and more predictable costs

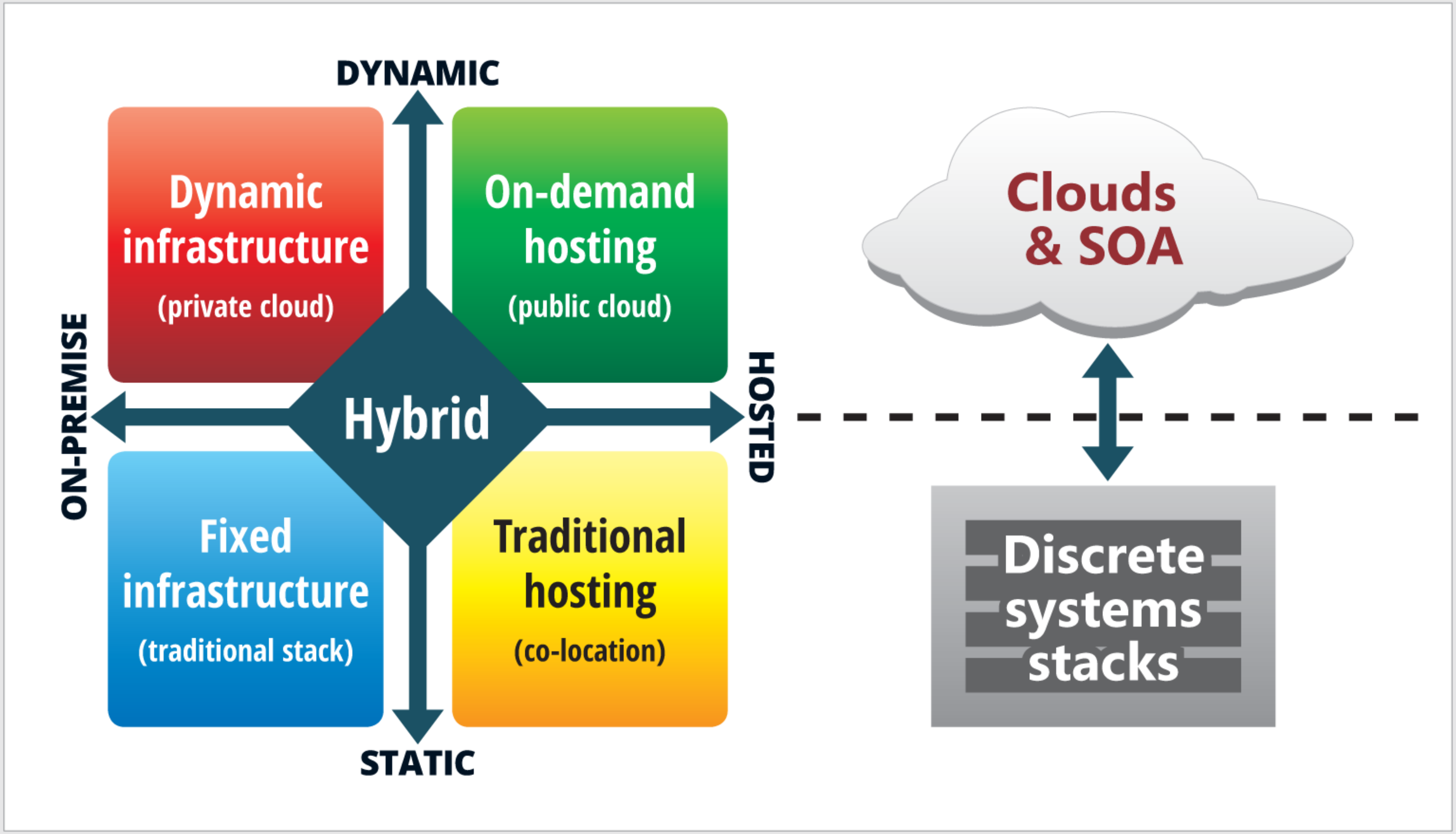
Better systems reliability/performance

30,000ft context: Types of change

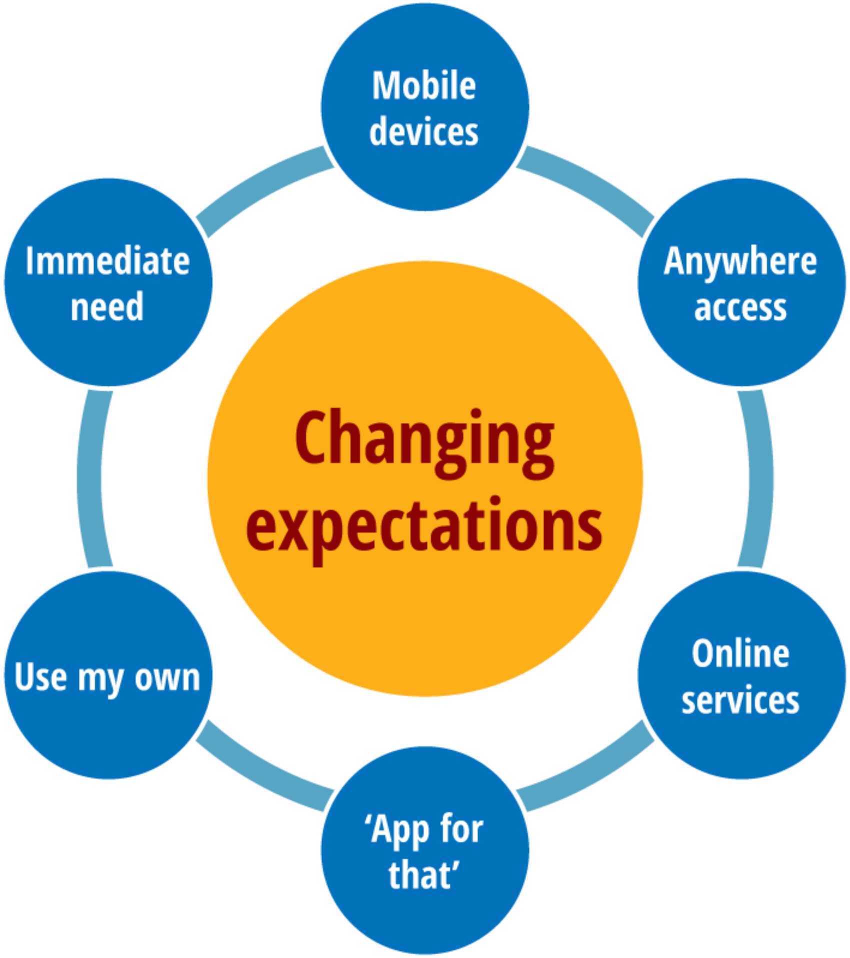


Most business change is incremental

20,000ft context: Moving platform targets



10,000ft context: Users and front ends



Generic browser based access

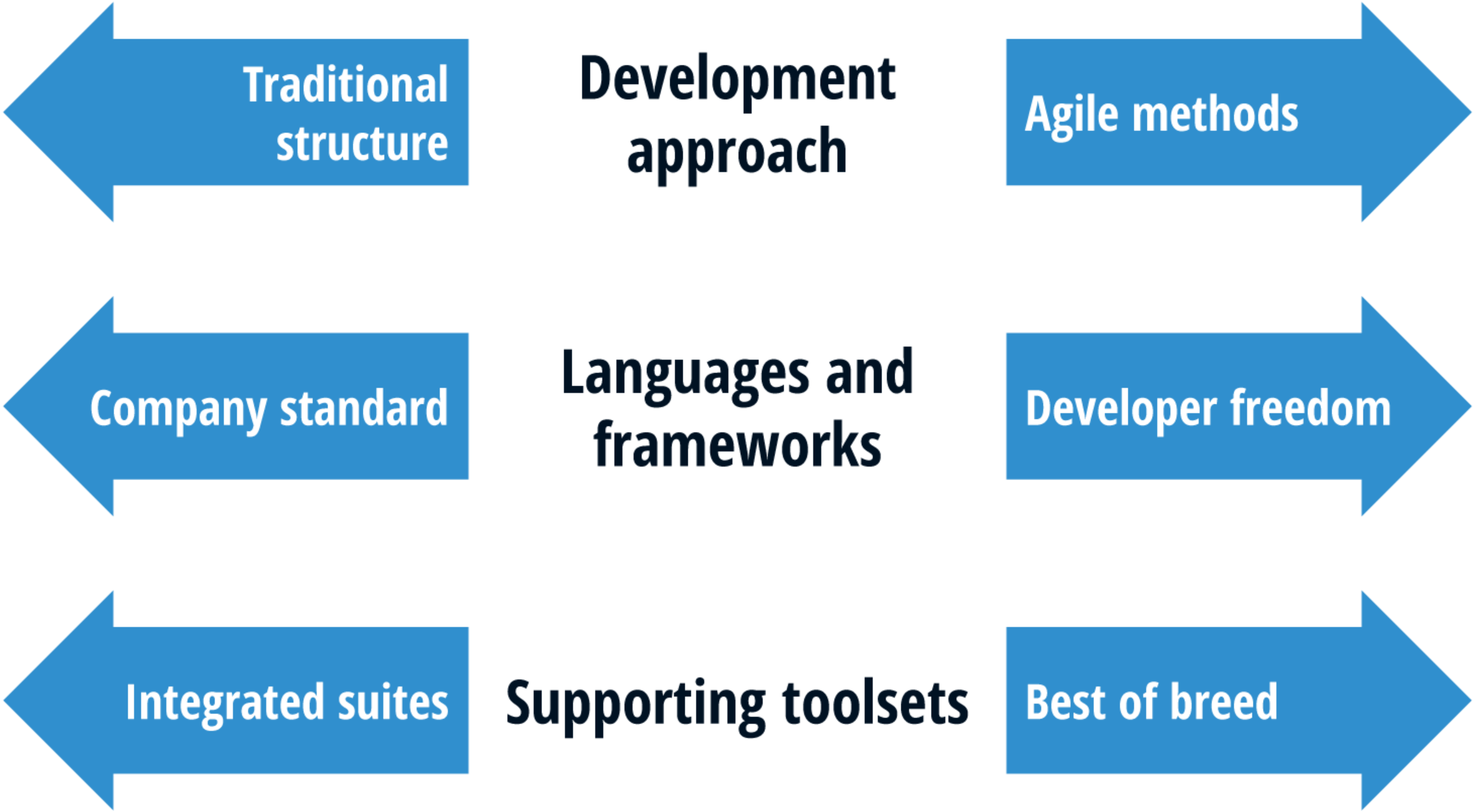
VS

Optimised native front-ends

VS

Client agnostic front-ends

5,000ft context: Tools and techniques



Traditional structure ← **Development approach** → Agile methods

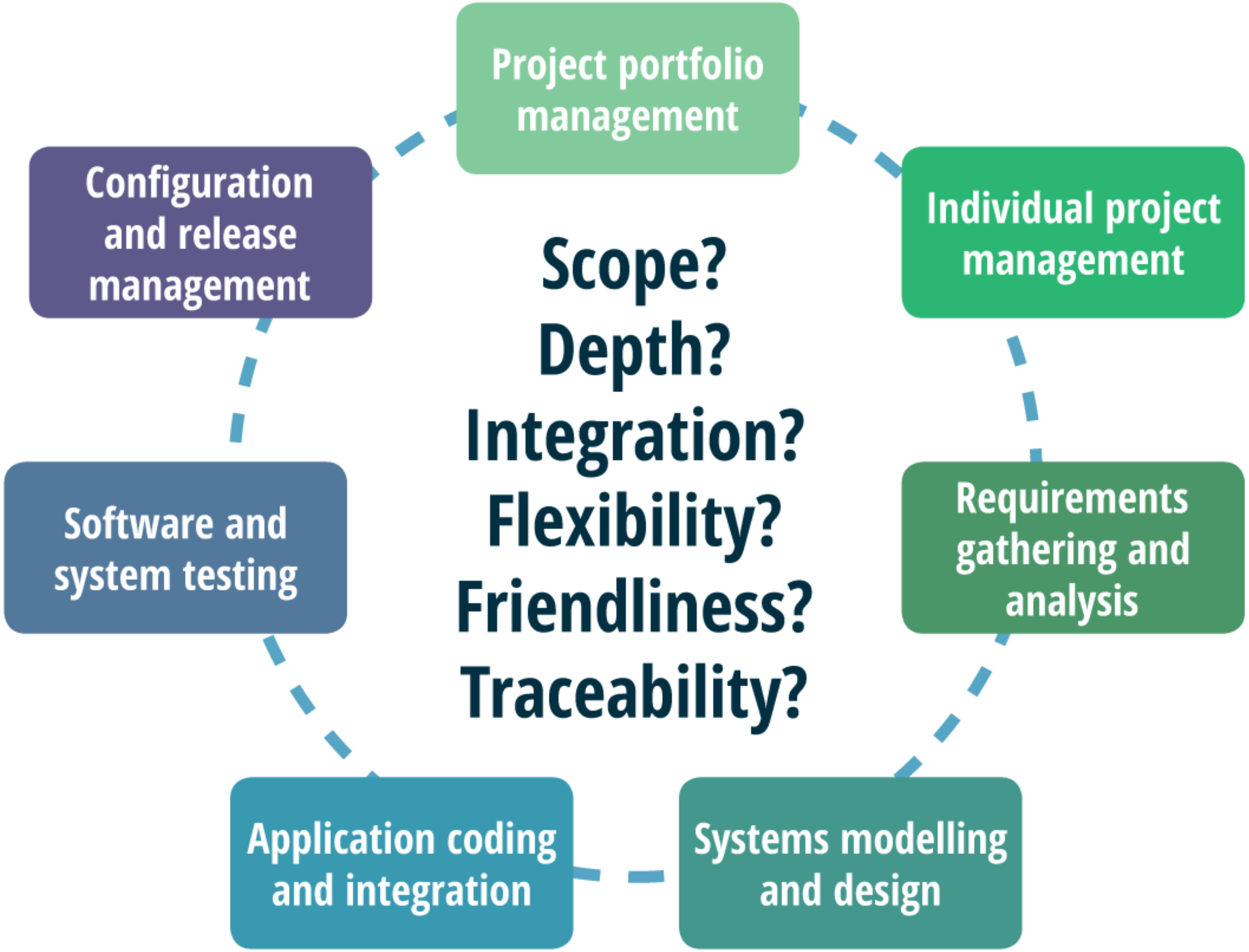
Company standard ← **Languages and frameworks** → Developer freedom

Integrated suites ← **Supporting toolsets** → Best of breed

The Register

PERFORCE

Ground level: Dev support facilities



Project portfolio management

Individual project management

Requirements gathering and analysis

Systems modelling and design

Application coding and integration

Software and system testing

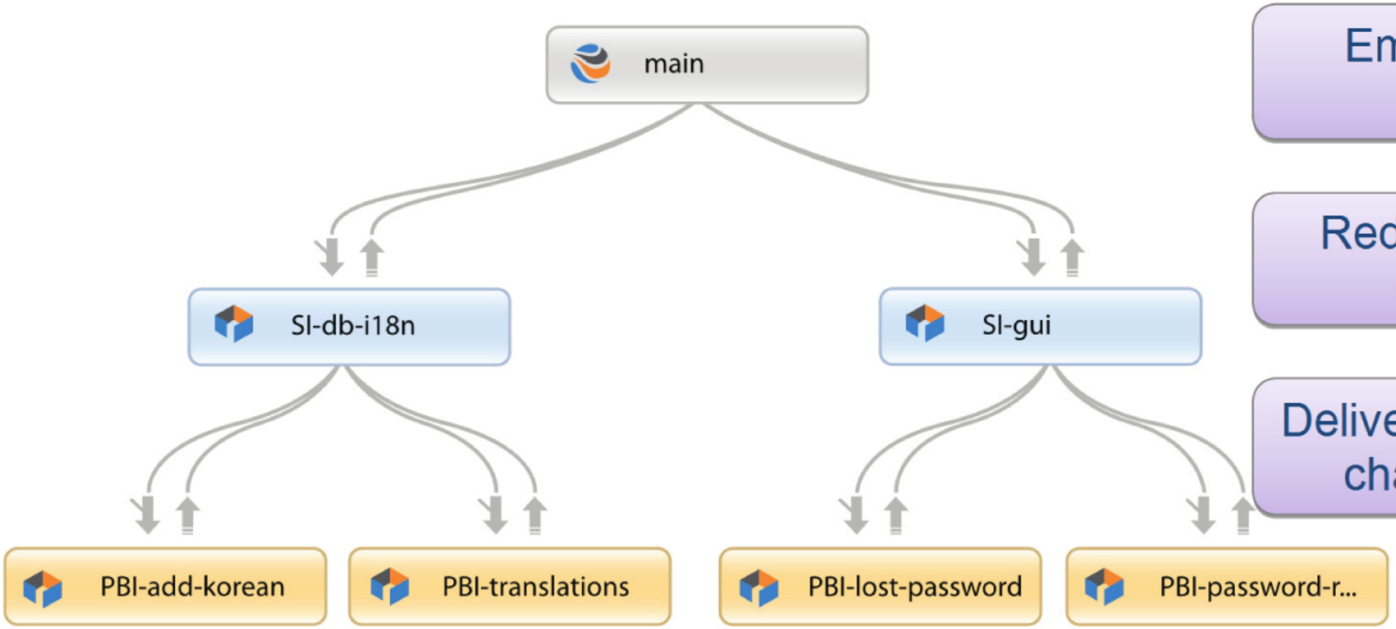
Configuration and release management

Scope?
Depth?
Integration?
Flexibility?
Friendliness?
Traceability?

NYSE

- Six staff running 14,000 servers
- Deploys server configurations as well as applications
- Nightly comparisons of servers with master definition in Perforce - System of Record
- 7,000 - 8,000 production releases per year

Managing development



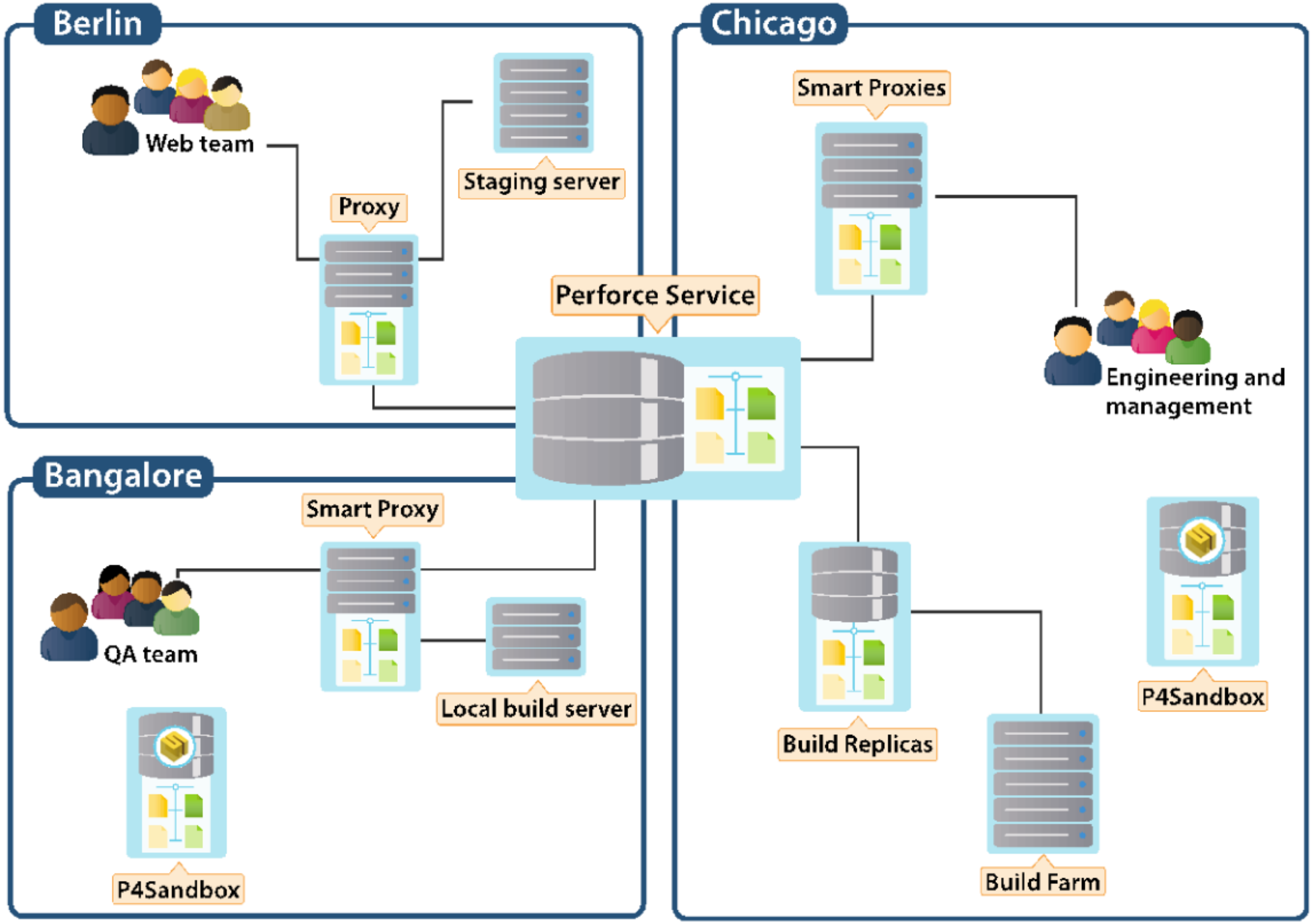
Emerging best practice

Reduce planning overhead

Deliver small units of change rapidly

Backlog items worked in branches and merged, tested, and shipped when ready

A typical dev scenario



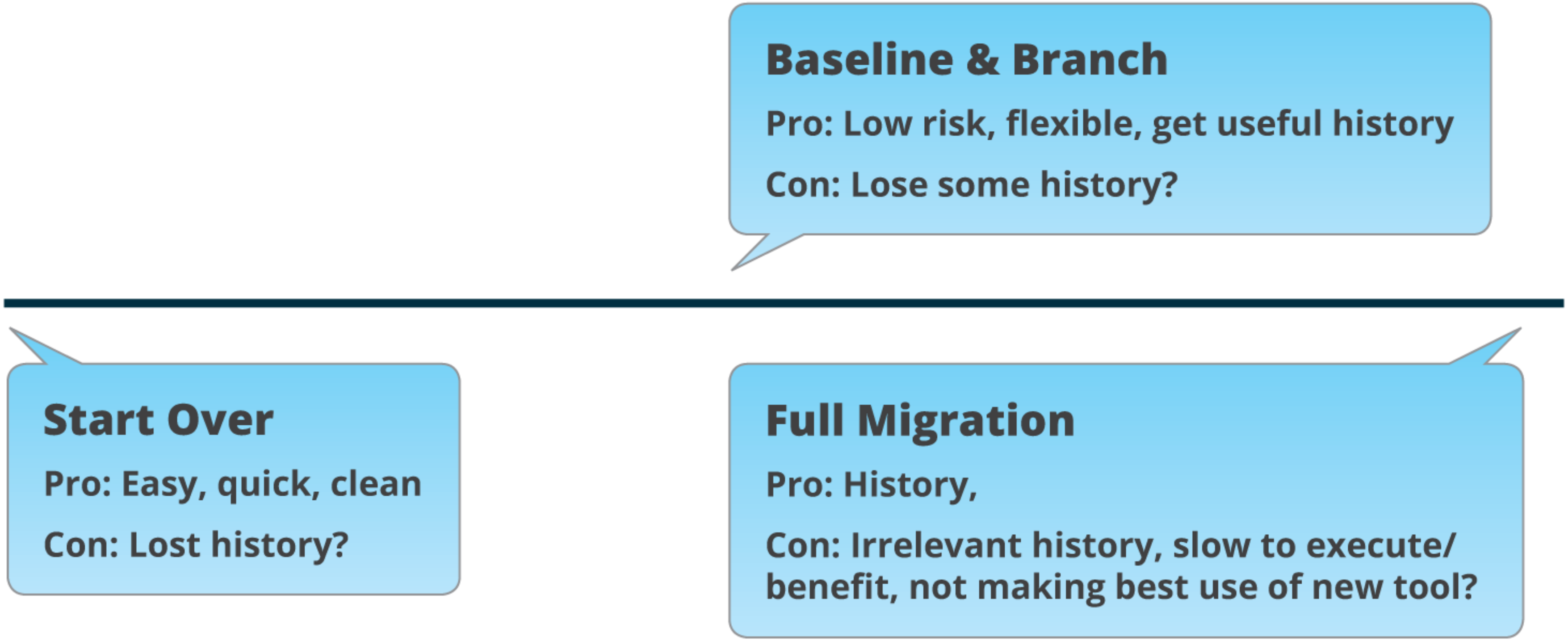
The Register

PERFORCE

Agile in the real world

- More than 100,000 customers use Salesforce for 10 billion transactions per month
- "Extreme Agile" development: 125+ scrum teams and 90,000+ automated tests

More traumatic: moving house or moving your version management?



Baseline & Branch

Pro: Low risk, flexible, get useful history

Con: Lose some history?

Start Over

Pro: Easy, quick, clean

Con: Lost history?

Full Migration

Pro: History,

Con: Irrelevant history, slow to execute/ benefit, not making best use of new tool?

PIXAR

- Perforce versions every frame, image and sound
- 40 million changelists
- 12 million transactions per day
- Largest single depot: 70 Tb
- Largest single check-in: 987 Gb

- Perforce used for software development process, chip designs
- 1,700 Perforce users
- 16TB storage, backups every 4 hours
- Mirrored data recovery site

What Reg readers say

- “Methodology was, is, and always will be a sideshow.”
- “[It] doesn’t scale well unless the requirements engineers and architects are heroically gifted.”
- “Code that is rushed through development and rollout is quite often of a lower standard, and hence the system will inevitably be less reliable.”

What Reg readers say

- “If you don’t have really good people, you’re screwed. Period.”
- “[The problem is] lack of communication between workstreams, leading to several wheels being reinvented… which causes a longer-term support and maintenance nightmare.”
- “Segmenting the code across multiple agile teams creates the necessity to manage dependencies.”

Further reading

- "Flow of Change" white paper
  http://bit.ly/pfagile1
- Successfully scaling an agile innovation culture
  http://bit.ly/pfagile2
- Seven things I hate about Agile
  http://bit.ly/pfagile3
- Ovum report on DVCS
  http://bit.ly/pfagile4

Thanks for joining us

An archived version of today's events will be made available on The Register in the near future.