Adopting Agile Approaches for the Enterprise

CollabNet Solutions

Presented by: Brian Zeichick, CollabNet Product Manager
Email: bzeichick@collab.net

Today's Agenda

- Developing in the Cloud for Governance
- Distributed Development Challenges
- Agile Development Methods
- The real story of Agile Adoption
- Agile Development Challenges
- Agile Development Tools
- Continuous Integration
- Object "Traceability"
- Staying Up-to-Date on Everything
- Agile Training

Why is "Development in the Cloud" so important?

- Development tools are still a distributed mess in most enterprises
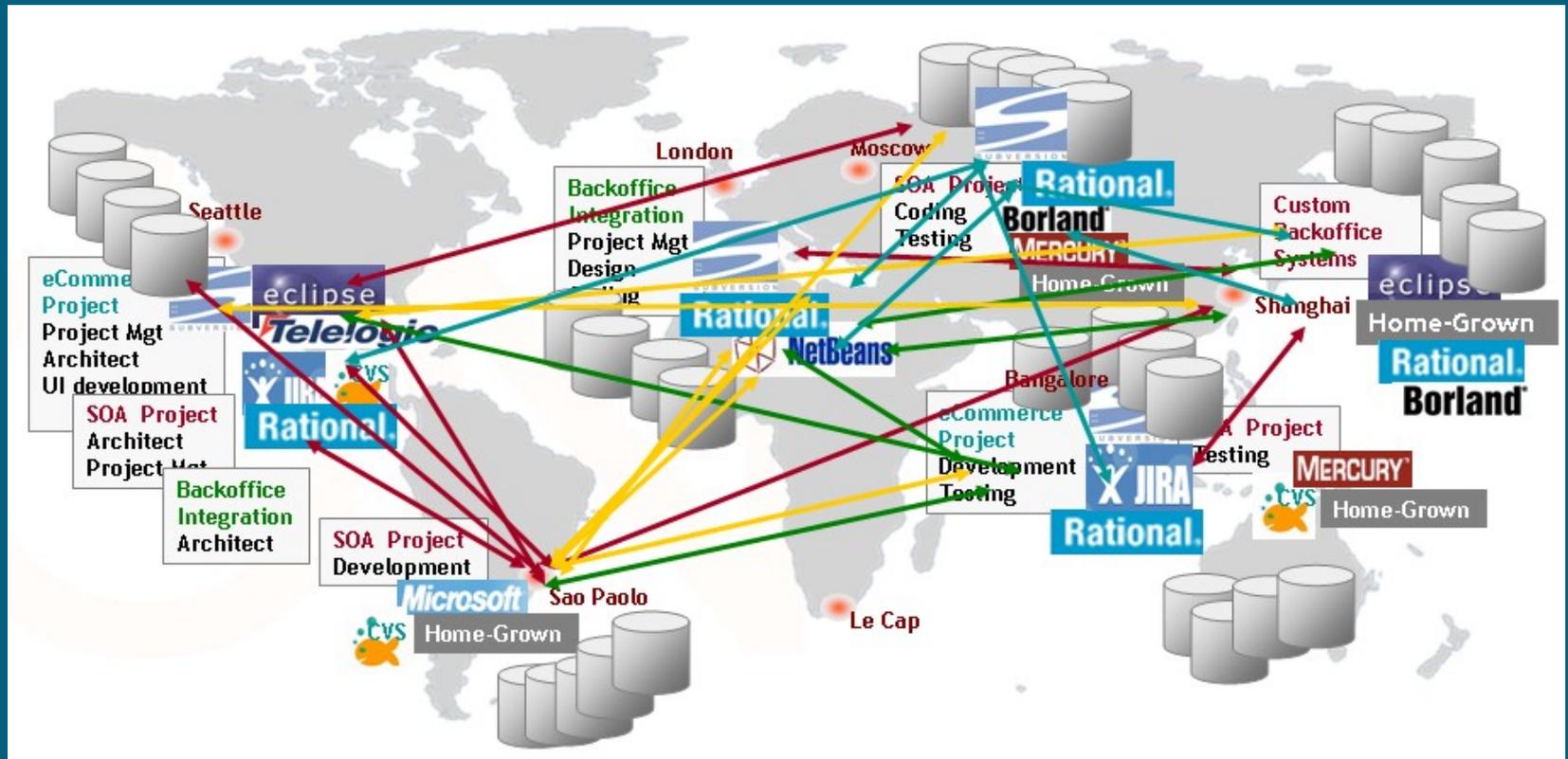
Why is "Development in the Cloud" so important?

- The fix is simple… centralize all development assets in the cloud

Dramatically speed global development with improved quality, visibility, and traceability
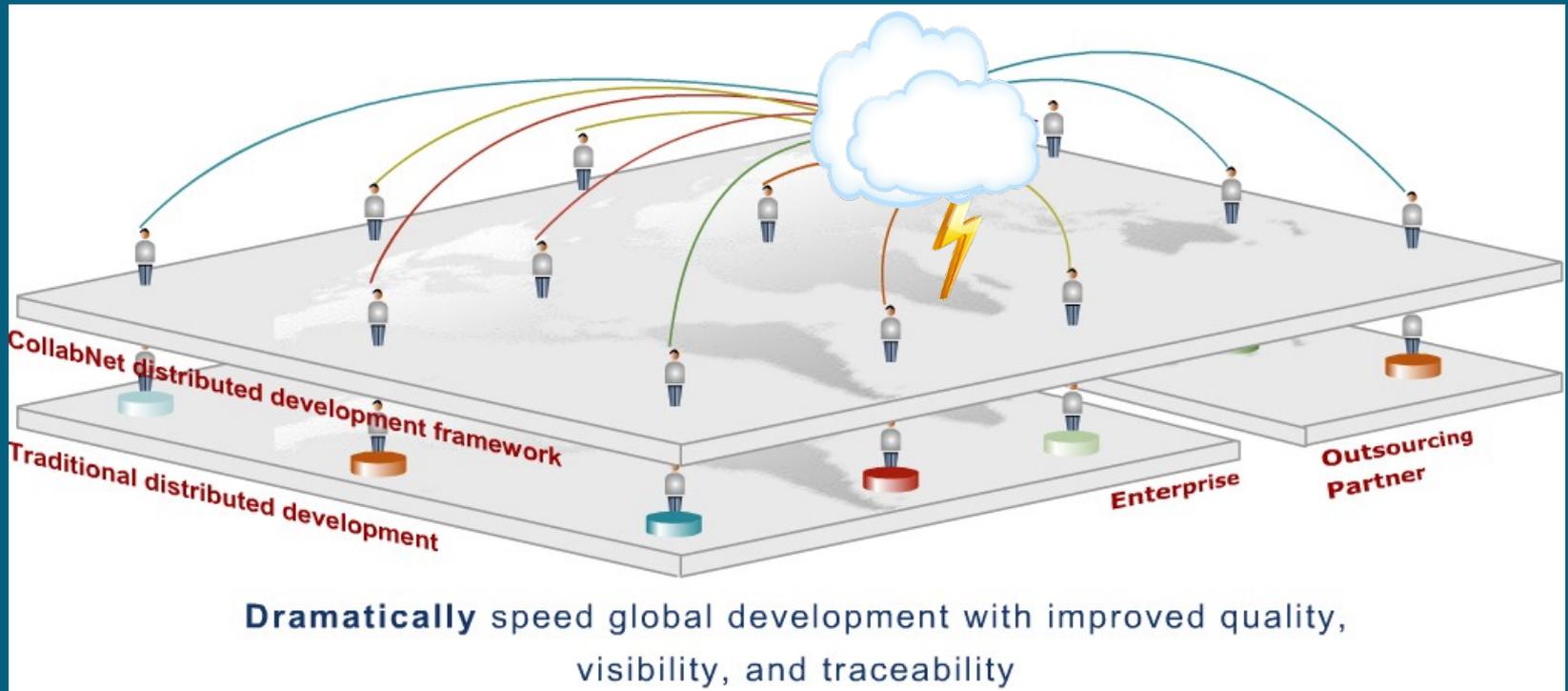
TRANSPARENCY breeds TRUST which drives REUSE

Common Distributed Development Challenges

- Communication

- Collaboration

- "Throw it over the Fence" development
 - Siloed Waterfall Approach

- Different Development Tools

The Appeal of Agile Development Methods

Characteristics of Agile Methods:

- **Shorter Planning Cycles**
  - Smaller set of requirements identified more often

- **Frequent Delivery**
  - Focus on working software for every iteration
  - "The only accurate measure of software development is the delivery of software"

- **Closer Working Relationship with the Business or Customer**
  - Constant feedback and frequent user acceptance testing
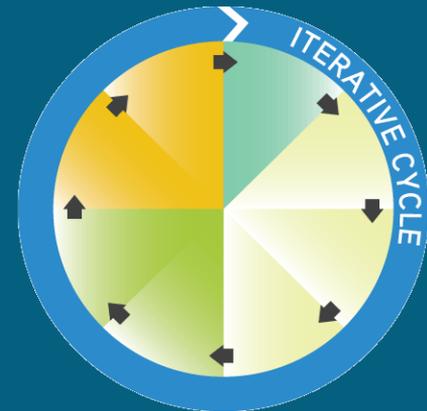
Agile Methods Allow Organizations to:

- **Innovate faster and shorten time to market**

- **Better respond to changing customer needs**

- **Reduce Costs**

ITERATIVE CYCLE

6

Enterprise Agile Adoption – The Real Story

Enterprise Agile Adoption

Methodology Maturity Timeline

Agile (e.g. Scrum, Kanban)

Agile (Hybrid)

Iterative (FDD)

Iterative (RUP)

Witches Brew

Waterfall (…)

- Templates
- Transparency
- Mutual Adoption
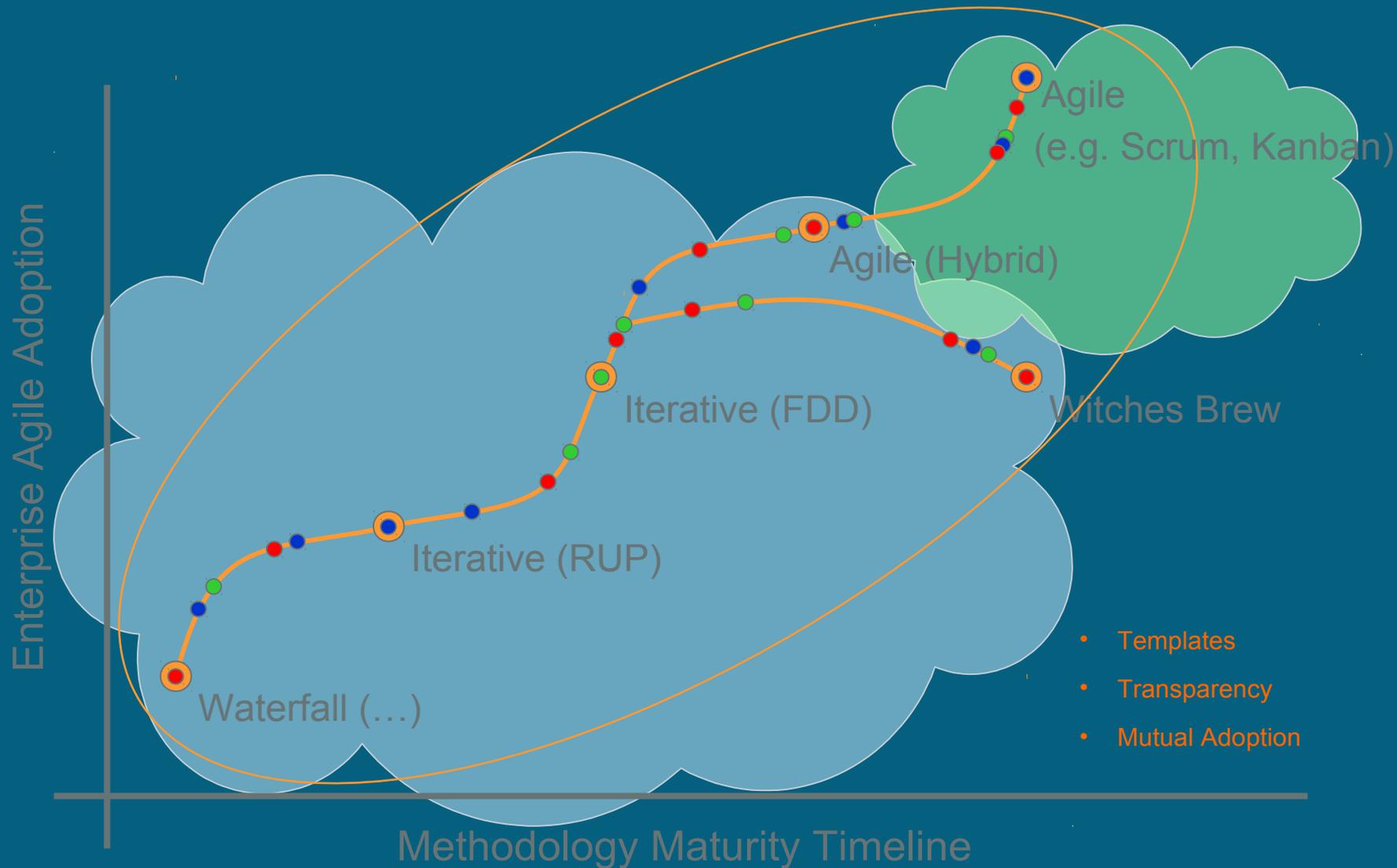
Enterprise Agile Development Challenges

- Classic ALM practices and role definitions can seriously impede the pace (and ultimate success) of agile development

Requirements Management

QA Test Management

Development Management

Infrastructure & IT support

Deployment & Ops support

Requirements, Defects, etc.

The Backlog: Stories, Defects, Tasks, etc.

Virtualized Dev, Build, Test, and Prod Clouds

Managed Hardware

Enterprise Agile Development Challenges

- Teams who set out to do agile development for the first time often get into hot water with project management and customers
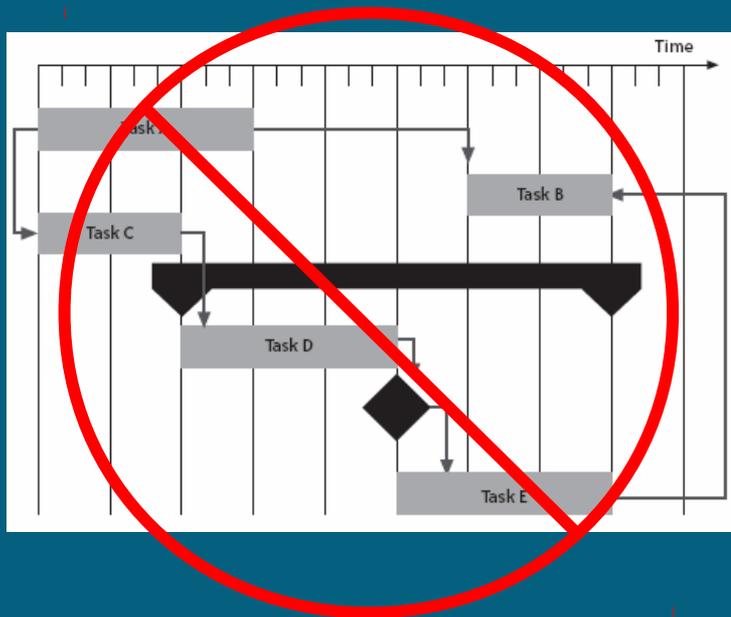
Enterprise Agile Development Challenges

- Improving collaboration and managing more hand-offs within a single team, which is sometimes distributed.

Enterprise Agile Development Challenges

- Agile is hard to distribute geographically when the development teams are not aligned with the application architecture

Enterprise Agile Development Challenges

- Agile can be a developer's worst nightmare if they cannot get comfortable with people picking at their stuff

Team-based Estimation

Daily Standup Meetings

User Acceptance Testing

Programming in Pairs

Progress Transparency

Test-Driven Development

Enterprise Agile Development Challenges

- Agile requires a skilled, experienced, and ultimately self-propelled development team (or it can backfire dramatically)

Enterprise Agile Development Challenges

- Coordinating across teams – some of which may be doing Agile, some not
  - Especially important for SOA and other corporate "reuse" initiatives.
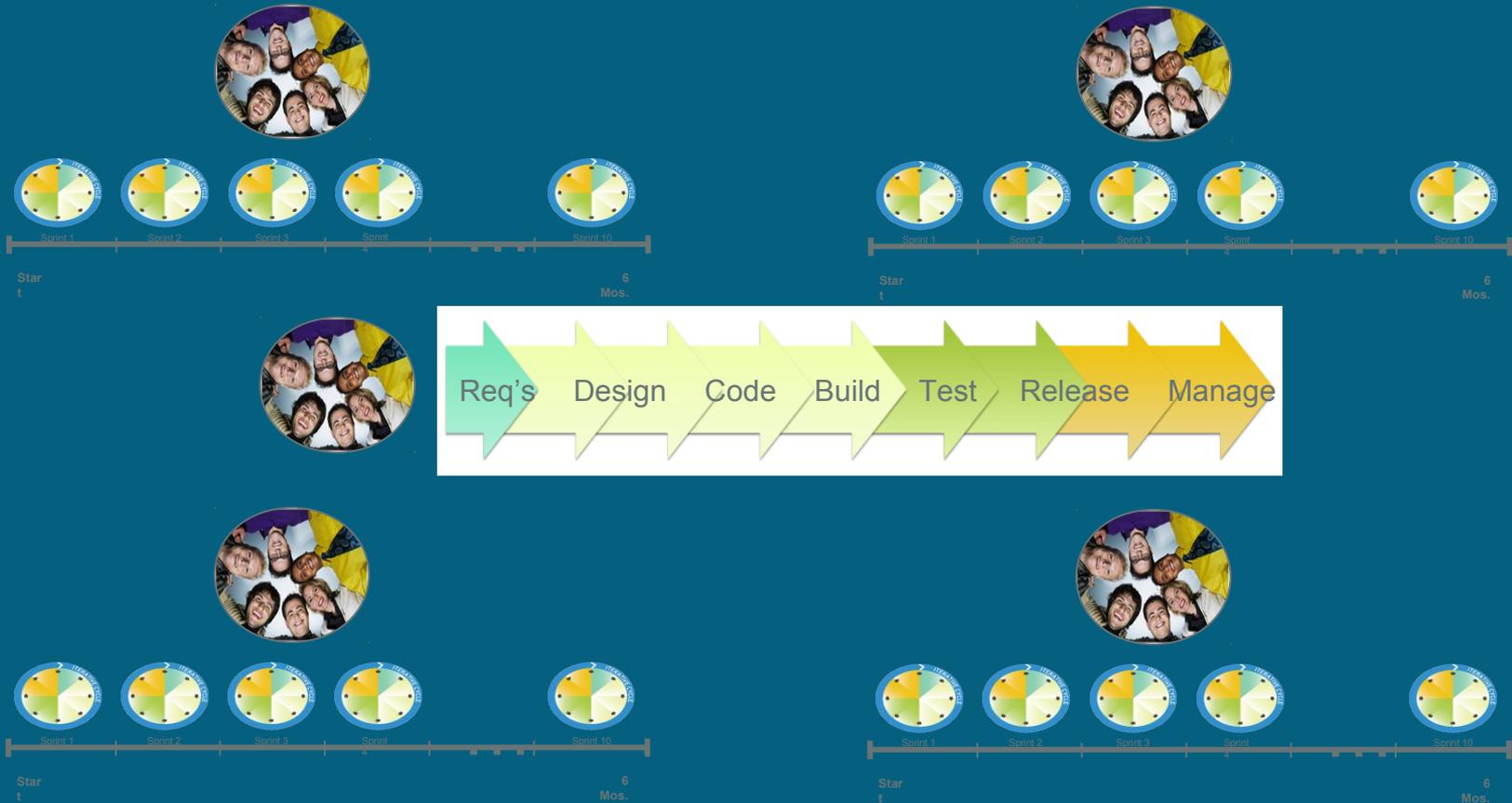
So, you are looking for Agile development tools?

Agile Point Tools

ALM Platforms

?
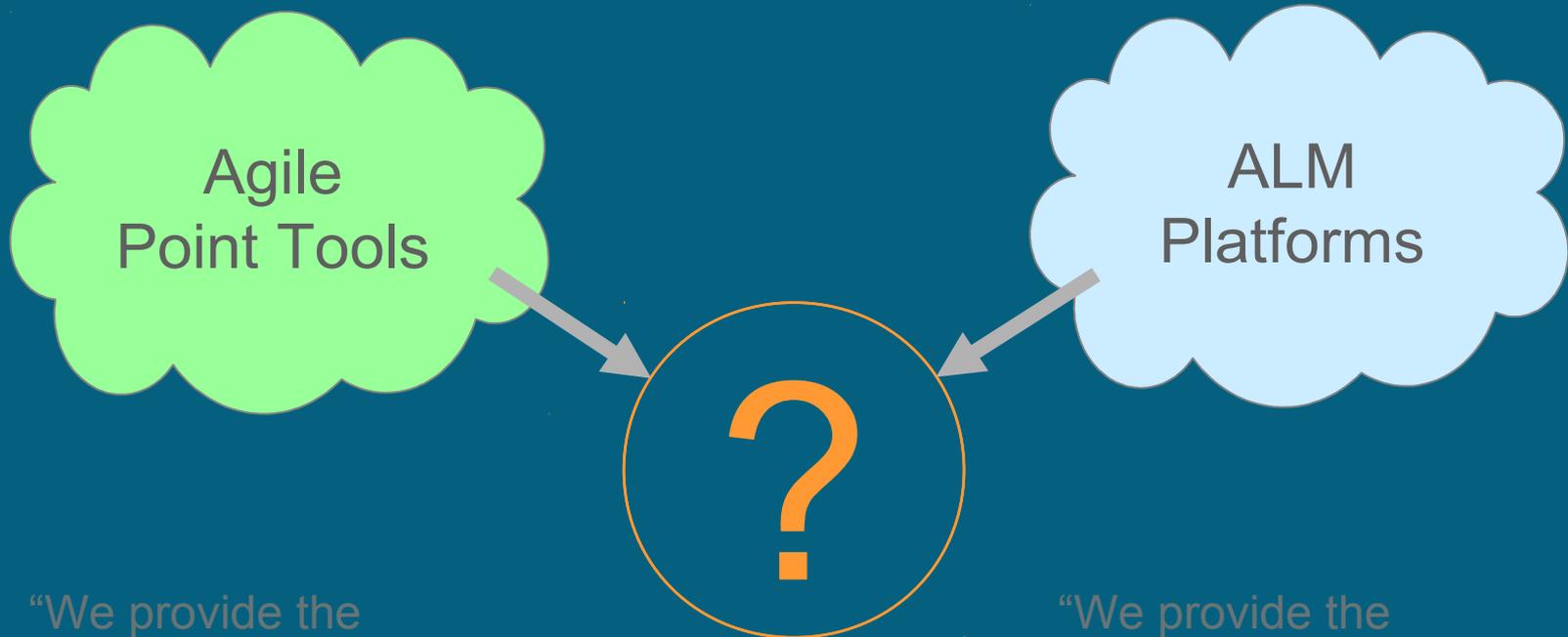
"We provide the features necessary to manage your new Agile projects really well"

 - Typical Agile Vendor

"We provide the features necessary to manage all of your software projects"

 - Typical ALM Vendor

Why start with Continuous Integration?

- There are lots of uses for private clouds in software development

- Why is *Continuous Integration* a compelling place to start?

  - Continuous Integration (CI) is a huge Agile trend. As development teams get leaner and meaner, they need to cast bigger safety nets.

  - To do CI right, you need servers… lots of them. If your software runs in multiple environments, you need even more, of different types.

  - Developers want to **orchestrate** their own environments, on demand, with minimal fuss. They need to develop, not rack & administer.

  - As **cloud-based** production environments increase, so then will the need for **cloud-based** continuous integration environments.

Full Lifecycle Continuous Integration in the Cloud

1. Requirements and tasks (i.e. development work items) are created and tracked using CollabNet's unique Dynamic Planning capability
2. When the work is ready to start, a developer picks up the next available work item and uses it to track his activities through the application lifecycle

Full Lifecycle Continuous Integration in the Cloud

3. As each work item progresses, it can be associated with any number of documents, discussions, Wikis, releases, etc. for complete traceability

4. When it finally comes time for the developer to commit changes on behalf of his work item, the ultimate association occurs between work item and code

Full Lifecycle Continuous Integration in the Cloud

5. After the developer commits his code, the Hudson server wakes up and prepares to run the test suite, known as a "job" in the Hudson console

6. Before running the tests, Hudson requests several build and test servers from the TerreMark cloud, one for each supported application platform

Full Lifecycle Continuous Integration in the Cloud

7. Based on the details of Hudson's request, TerreMark provisions several servers on the fly, cloned from "masters" for each supported platform

8. When the servers are ready for Hudson's use, TerreMark notifies Hudson so that it can begin executing tests in the continuous integration suite

Full Lifecycle Continuous Integration in the Cloud

The number of continuous integration servers over time
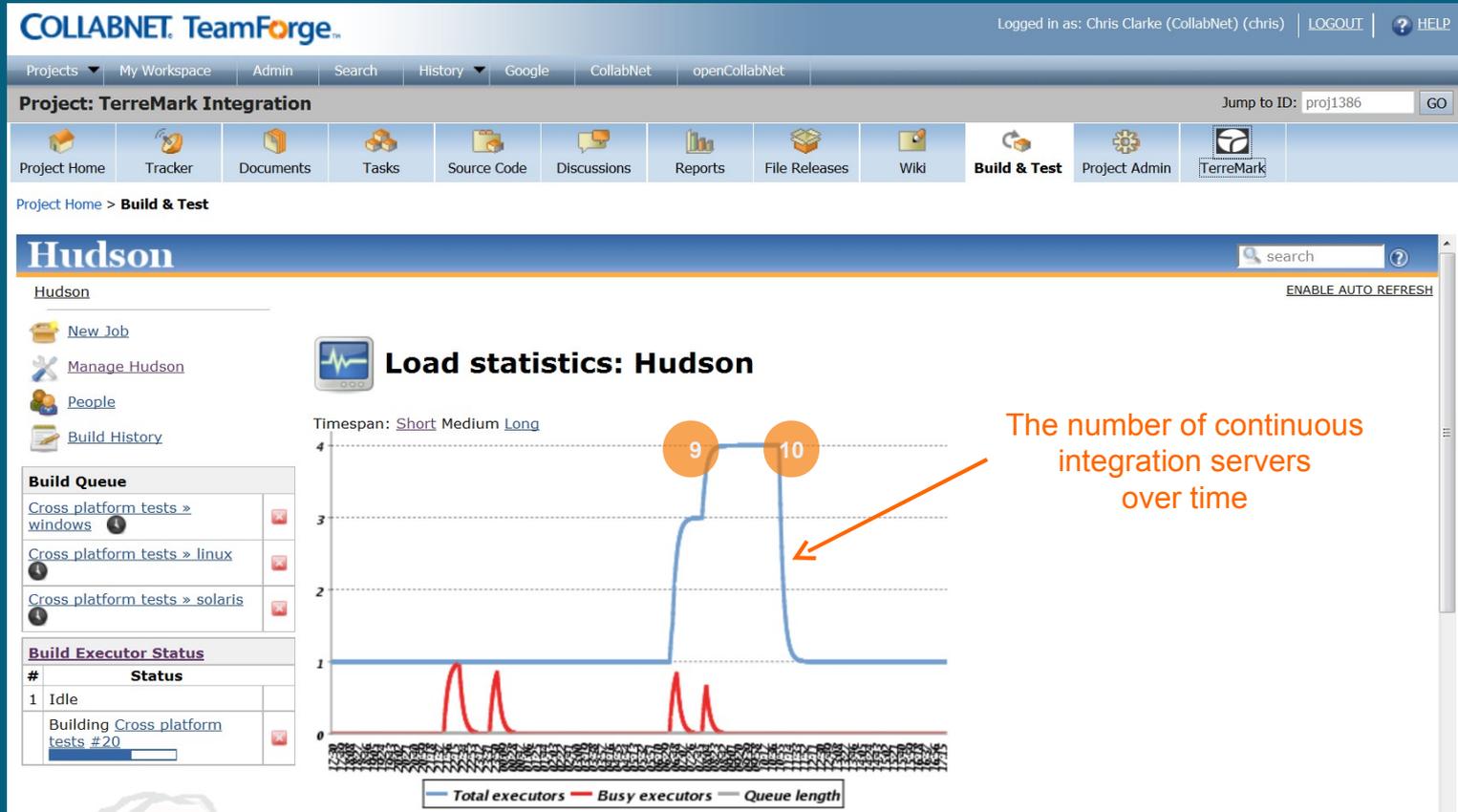
9. After being notified by TerreMark that the requested server resources are available, Hudson executes the tests and records the results

10. After the tests are finished, Hudson notifies TerreMark that the test resources can be relinquished (until the next continuous integration test run)

Full Lifecycle Continuous Integration in the Cloud

11. After being notified by Hudson that the provisioned resources are no longer required, TerreMark makes the resources available for other uses

Full Lifecycle Continuous Integration in the Cloud

12. With Hudson's successful execution of the continuous integration test suite, the software build is deposited into the TeamForge file release system

"Object" Traceability
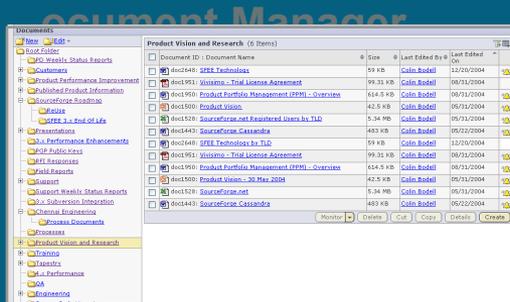
Document Manager

Upload, download, and share versioned documents with the distributed project team

Tracker
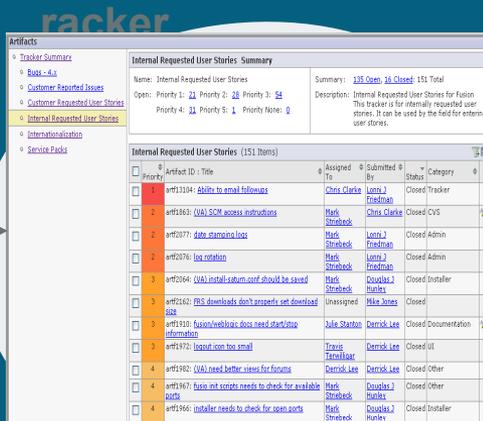
Manage "artifacts" (defects, requirements, user stories, story tasks, risks, tests, and more)

File Release System
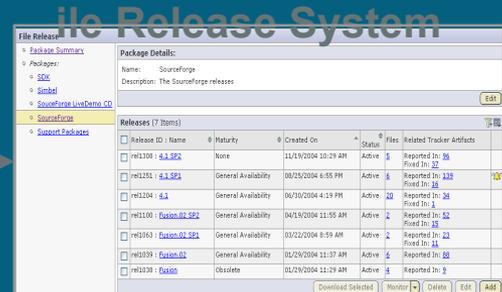
Upload, download, and share software release packages with your customer community

Task Manager

Upload a MS Project plan to the system and allow participants to update progress in real time

Source Code Manager

Connect and manage multiple version control repositories for storing project code, etc.

- Every asset is a uniquely identifiable "object"
- Any object can be associated with any other
- Associations are easy to find in the user interface
- Associations provide end-to-end asset traceability!

Stay Up-to-Date on Everything
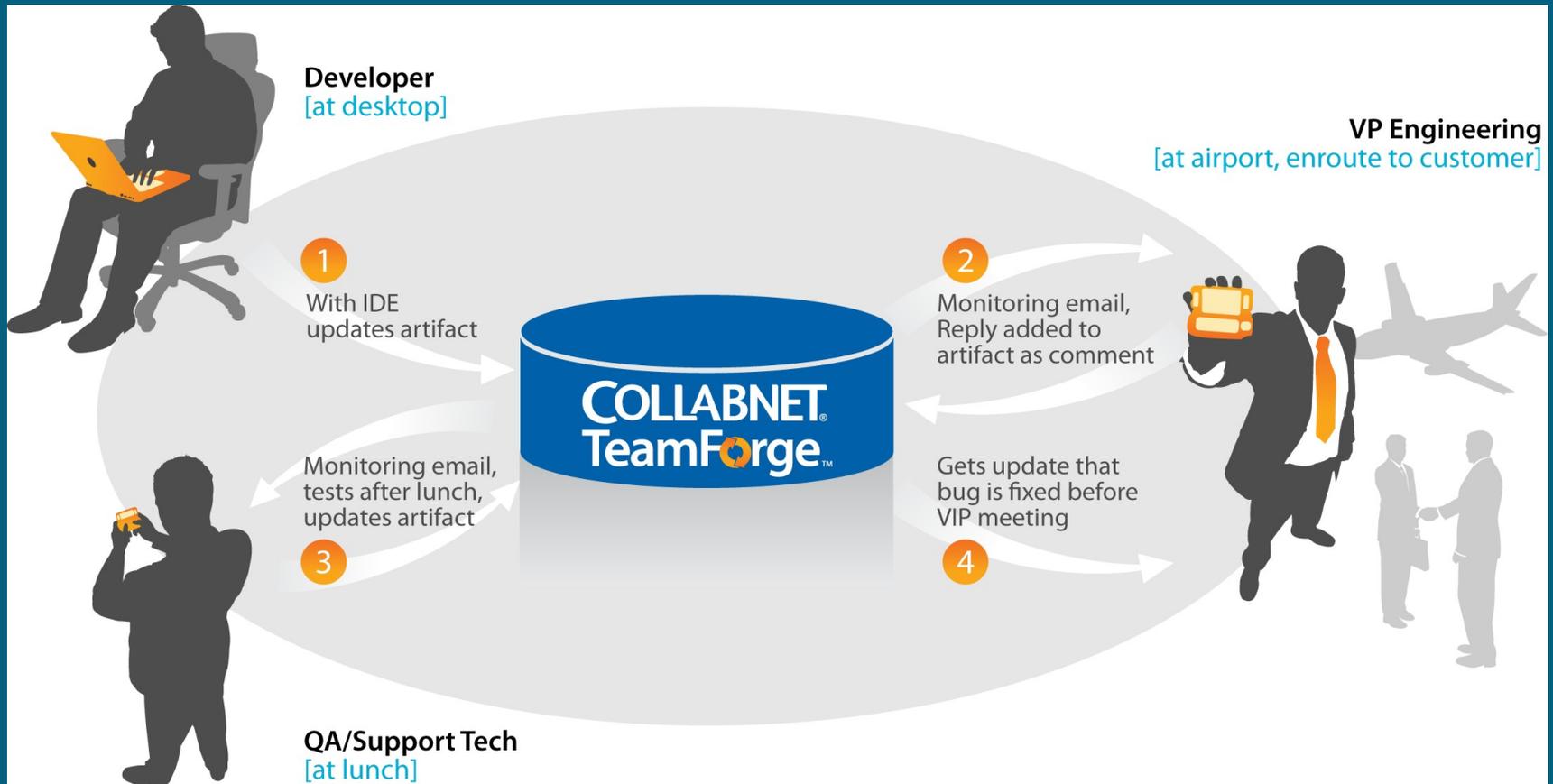
New to Agile? We can show you the ropes!

- Certified Scrum Courses
  - Public courses throughout N. America and Europe
  - Private courses on demand for organizations moving to Scrum

- Coaching
  - Our team possesses deep experience leading organizations — from small businesses to multi-national enterprises — through Scrum transformations

- Free Webinars
  - Led by our experts on a range of topics

- Find out more
  -

SCRUM CORE TRAINING

Questions and Answers

From our Live Audience