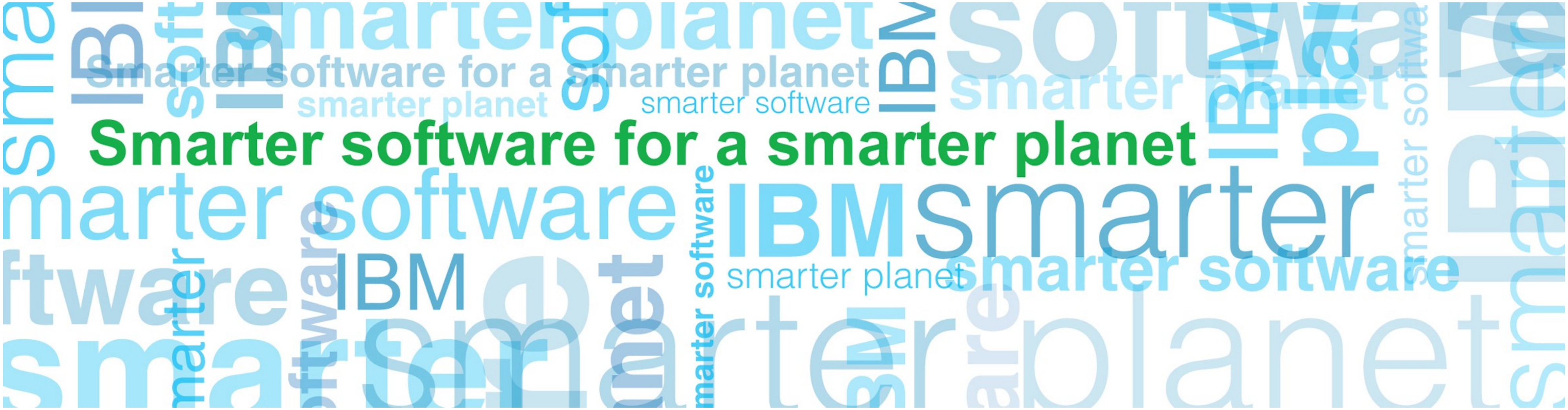
Alan Ogilvie – Java SDK Project Manager
21st March 2012

Delivering the IBM SDK, Java Technology Edition 7.0

Managing a large multi-site software engineering project using an Agile approach



Important Disclaimers

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILST EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

ALL PERFORMANCE DATA INCLUDED IN THIS PRESENTATION HAVE BEEN GATHERED IN A CONTROLLED ENVIRONMENT. YOUR OWN TEST RESULTS MAY VARY BASED ON HARDWARE, SOFTWARE OR INFRASTRUCTURE DIFFERENCES.

ALL DATA INCLUDED IN THIS PRESENTATION ARE MEANT TO BE USED ONLY AS A GUIDE.

IN ADDITION, THE INFORMATION CONTAINED IN THIS PRESENTATION IS BASED ON IBM’S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM, WITHOUT NOTICE.

IBM AND ITS AFFILIATED COMPANIES SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

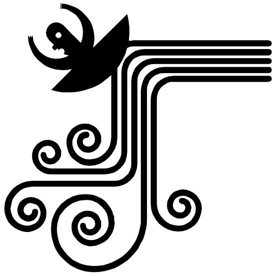
- CREATING ANY WARRANT OR REPRESENTATION FROM IBM, ITS AFFILIATED COMPANIES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS

Introduction to the speaker

- 16 years software engineering experience
  - Developer
  - Tester
  - Team Leader
  - Line Manager
  - Project Manager
- PMP® certified
- Based in IBM's Java Technology Centre, in Hursley, UK
- Recent work focus:
  - Project Manager responsible for IBM SDK, Java Technology Edition 7.0
  - Project Manager responsible for IBM SDK, Java Technology Edition 6.0 in WebSphere Application Server 8.0
- Email: alan_ogilvie@uk.ibm.com

Key Messages

- It is possible to apply Agile development techniques to large-scale software engineering projects
- Agile projects deliver products which are higher quality than those delivered using the Waterfall approach, whilst providing a more flexible approach to delivery, resulting in happier stakeholders
- IBM has demonstrated this with its recent Java 7 delivery



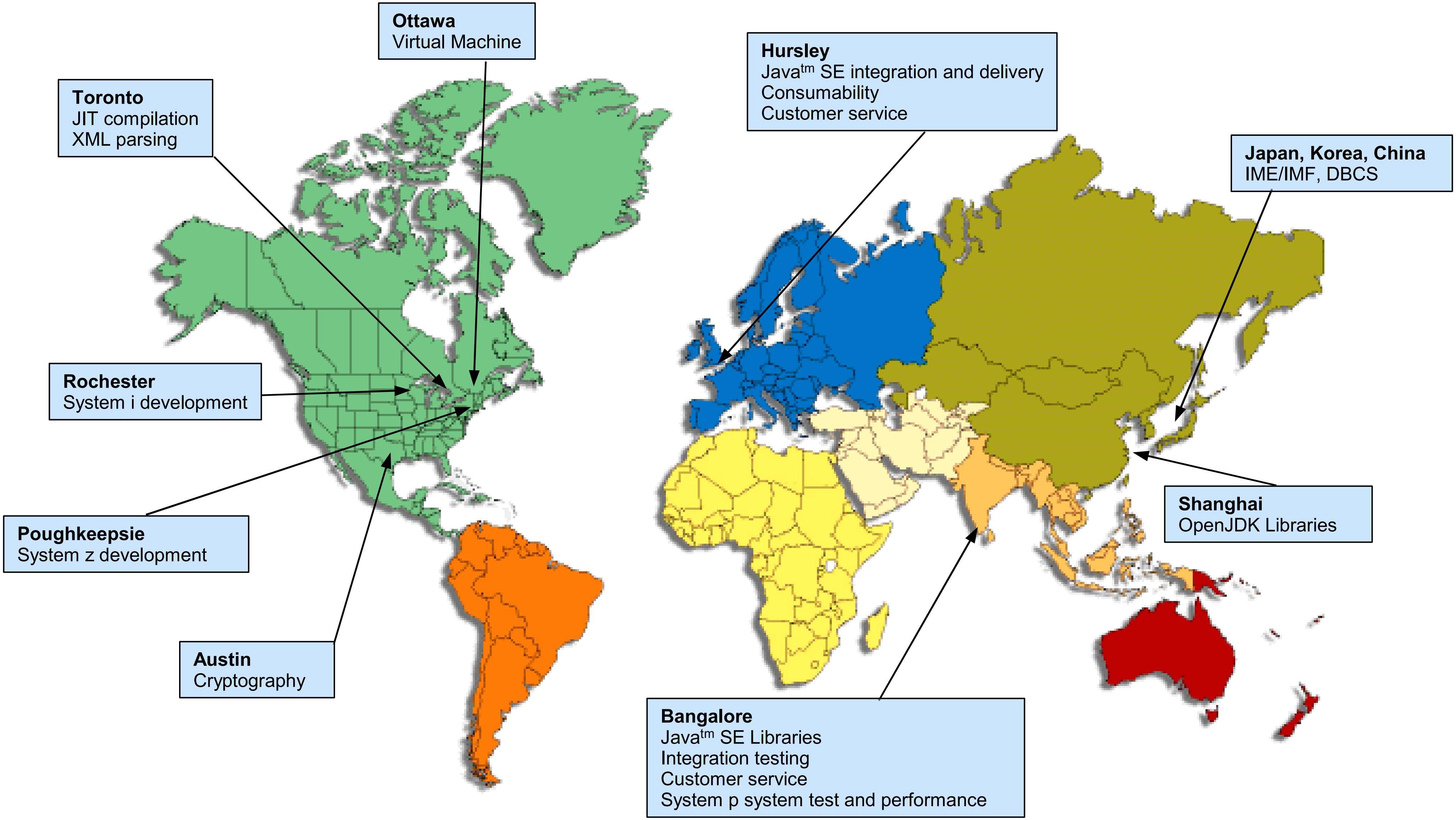
IBM Java Technology Edition Version 7.0

- General Availability 19th September 2011
  - Improved throughput
  - Faster startup
  - Smaller footprint
  - Introduces Balanced GC
  - Soft Real Time capabilities
  - Improved consumability
- Operating systems
  - AIX, Linux, z/OS, Windows, Solaris
- Platforms
  - Power, System Z, Intel, AMD, SPARC

Scale of the IBM Java 7 project

- Several hundred staff across 4 continents
- ~750,000 person-hours
- 20 platforms built and tested every day
- 1000’s of test machines
- >1 million hours of testing per month

IBM Java – A world wide team effort



Ottawa
Virtual Machine

Toronto
JIT compilation
XML parsing

Hursley
Java^tm^ SE integration and delivery
Consumability
Customer service

Japan, Korea, China
IME/IMF, DBCS

Rochester
System i development

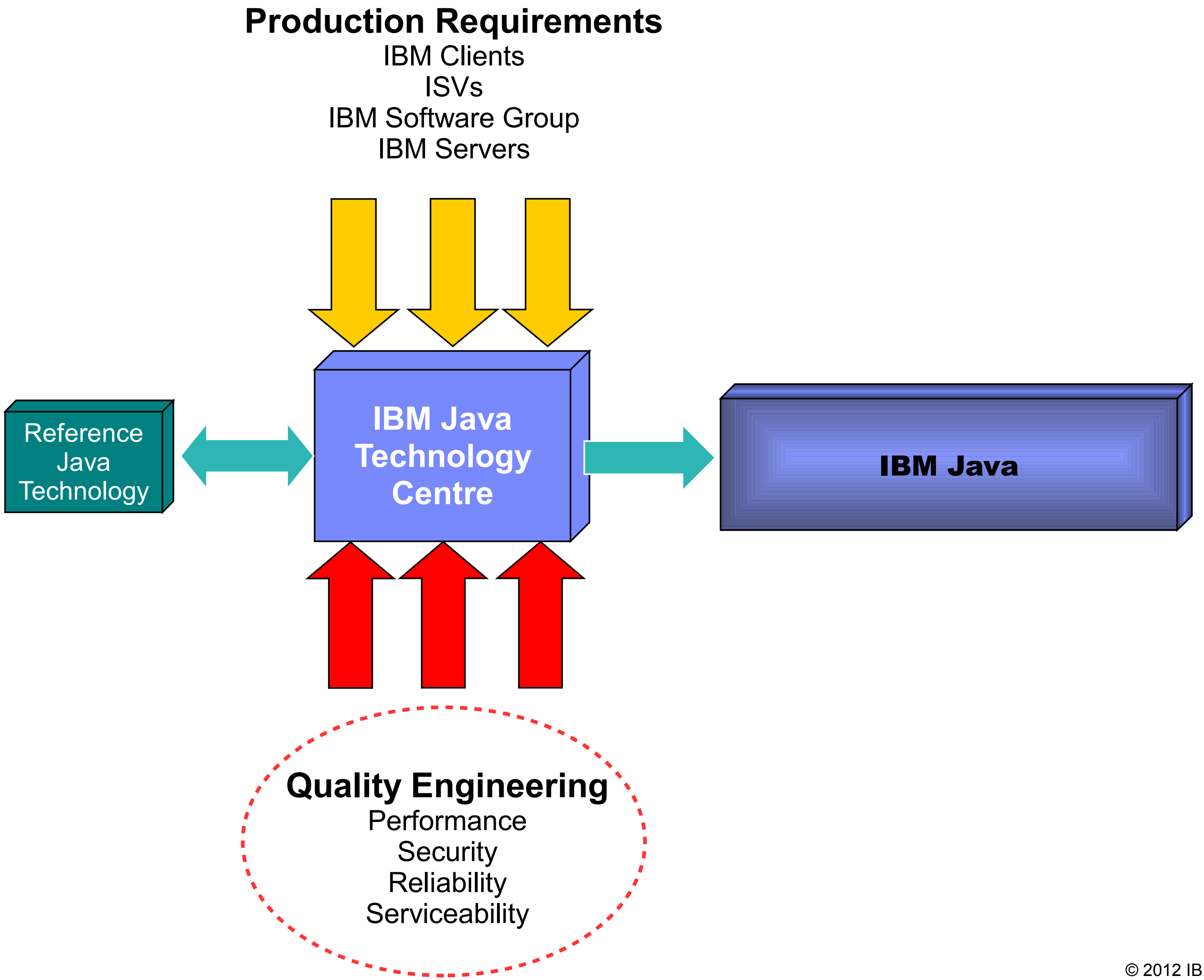
Shanghai
OpenJDK Libraries

Poughkeepsie
System z development

Austin
Cryptography

Bangalore
Java^tm^ SE Libraries
Integration testing
Customer service
System p system test and performance

IBM Java Technology Centre – what do we do?



Production Requirements
IBM Clients
ISVs
IBM Software Group
IBM Servers

Reference Java Technology

IBM Java Technology Centre

IBM Java

Quality Engineering
Performance
Security
Reliability
Serviceability

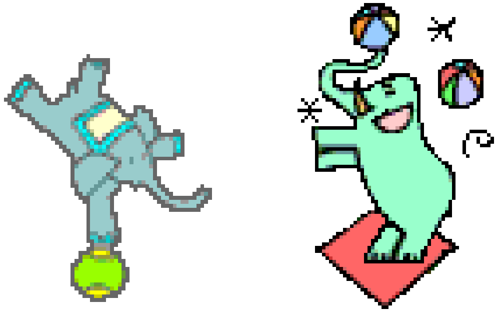
What do IBM (Software Group) Project Managers do?

- What type of person can bring this all together? What background do they need?

- Bridge between the technical community and the business
  - Communicate with, and lead, people who are far more technically talented!
  - Communicate with the business leaders; results matter
- Deal with traditional Project Management pressures
  - Content (Scope), Schedule, Resources (People & Machines), Quality
  - Balance the competing pressures of other projects (Portfolio Management)
  - Keeping stakeholders informed, and managing their expectations
  - Managing (avoiding if possible!) the inevitable end-of-project crunch
- Usually (but not always) have a software engineering background
  - Need to understand more than just the jargon & acronyms
  - Whilst metrics are key, project management is more than “managing by numbers”

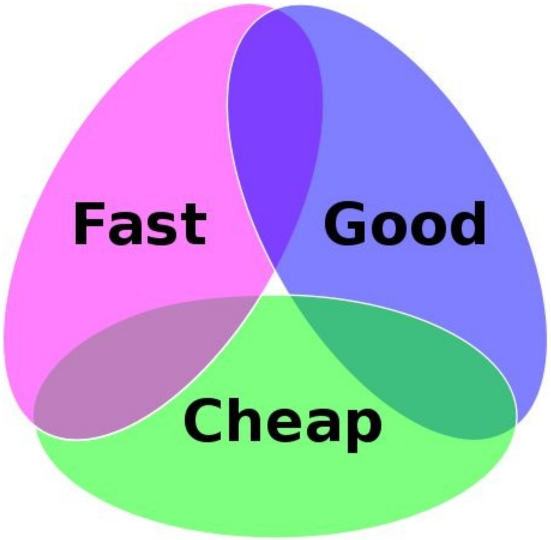
Software Engineering & Project Management in IBM

- Historically, IBM Software Engineering projects were based on the Waterfall model
- Agile started as a grass roots movement in IBM
- Agile has now become an accepted – in fact, the ***expected*** – model for Software Engineering projects
- IBM continues to tune and improve the application of Agile to large, complex projects; and in particular where the team is distributed geographically
- Innovation and evolution continues...

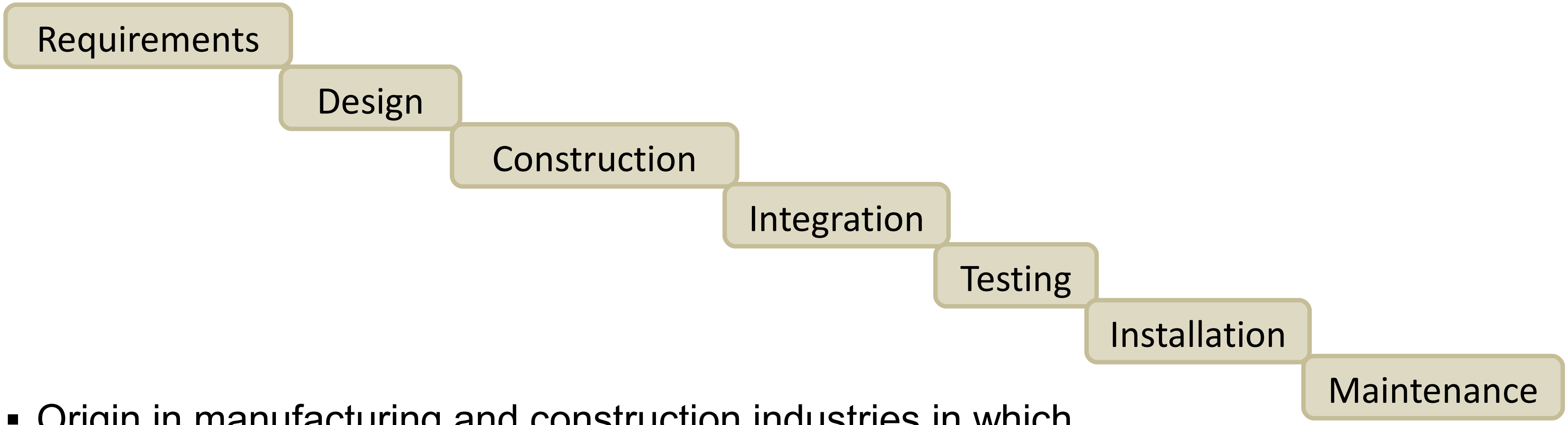


Project Management 101: The “Triple Constraint”

- Need to manage Schedule, Content & Resources (plus Quality!)
  - Schedule is usually the "immovable object"
  - Maintaining resources is always a challenge in a busy portfolio
  - Quality can rarely be compromised – nor should it be
- Various Project Management models have been designed to manage the triple constraint



Waterfall Model



- Origin in manufacturing and construction industries in which post-design changes are costly, if not impossible
  - Sequential
  - Up-front planning, sizing & content commitment
  - Little opportunity for accommodating late changes
- Model T Ford: “You can have any color as long as it's black.”

References:

•Royce, Winston (1970), "Managing the Development of Large Software Systems”

•Benington, Herbert D. (1 October 1983). "Production of Large Computer Programs". IEEE Annals of the History of Computing (IEEE Educational Activities Department) 5 (4): 350–361. doi:10.1109/MAHC.1983.10102.

Agile Principles

- From http://agilemanifesto.org/principles.html:
  - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - Business people and developers must work together daily throughout the project.
  - Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - Working software is the primary measure of progress.
  - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - Continuous attention to technical excellence and good design enhances agility.
  - Simplicity--the art of maximizing the amount of work not done--is essential.
  - The best architectures, requirements, and designs emerge from self-organizing teams.
  - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Principles

- From http://agilemanifesto.org/principles.html:
  - ***Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.***
  - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - Business people and developers must work together daily throughout the project.
  - Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - Working software is the primary measure of progress.
  - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - Continuous attention to technical excellence and good design enhances agility.
  - Simplicity--the art of maximizing the amount of work not done--is essential.
  - The best architectures, requirements, and designs emerge from self-organizing teams.
  - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Principles

- From http://agilemanifesto.org/principles.html:
  - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - ***Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.***
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - Business people and developers must work together daily throughout the project.
  - Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - Working software is the primary measure of progress.
  - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - Continuous attention to technical excellence and good design enhances agility.
  - Simplicity--the art of maximizing the amount of work not done--is essential.
  - The best architectures, requirements, and designs emerge from self-organizing teams.
  - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Principles

- From http://agilemanifesto.org/principles.html:
  - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - ***Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.***
  - Business people and developers must work together daily throughout the project.
  - Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - Working software is the primary measure of progress.
  - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - Continuous attention to technical excellence and good design enhances agility.
  - Simplicity--the art of maximizing the amount of work not done--is essential.
  - The best architectures, requirements, and designs emerge from self-organizing teams.
  - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Principles

- From http://agilemanifesto.org/principles.html:
  - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - ***Business people and developers must work together daily throughout the project.***
  - Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - Working software is the primary measure of progress.
  - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - Continuous attention to technical excellence and good design enhances agility.
  - Simplicity--the art of maximizing the amount of work not done--is essential.
  - The best architectures, requirements, and designs emerge from self-organizing teams.
  - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Principles

- From http://agilemanifesto.org/principles.html:
  - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - Business people and developers must work together daily throughout the project.
  - ***Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.***
  - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - Working software is the primary measure of progress.
  - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - Continuous attention to technical excellence and good design enhances agility.
  - Simplicity--the art of maximizing the amount of work not done--is essential.
  - The best architectures, requirements, and designs emerge from self-organizing teams.
  - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Principles

- From http://agilemanifesto.org/principles.html:
  - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - Business people and developers must work together daily throughout the project.
  - Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - ***The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.***
  - Working software is the primary measure of progress.
  - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - Continuous attention to technical excellence and good design enhances agility.
  - Simplicity--the art of maximizing the amount of work not done--is essential.
  - The best architectures, requirements, and designs emerge from self-organizing teams.
  - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Principles

- From http://agilemanifesto.org/principles.html:
  - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - Business people and developers must work together daily throughout the project.
  - Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - ***Working software is the primary measure of progress.***
  - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - Continuous attention to technical excellence and good design enhances agility.
  - Simplicity--the art of maximizing the amount of work not done--is essential.
  - The best architectures, requirements, and designs emerge from self-organizing teams.
  - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Principles

- From http://agilemanifesto.org/principles.html:
  - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - Business people and developers must work together daily throughout the project.
  - Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - Working software is the primary measure of progress.
  - ***Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.***
  - Continuous attention to technical excellence and good design enhances agility.
  - Simplicity--the art of maximizing the amount of work not done--is essential.
  - The best architectures, requirements, and designs emerge from self-organizing teams.
  - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Principles

- From http://agilemanifesto.org/principles.html:
  - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - Business people and developers must work together daily throughout the project.
  - Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - Working software is the primary measure of progress.
  - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - ***Continuous attention to technical excellence and good design enhances agility.***
  - Simplicity--the art of maximizing the amount of work not done--is essential.
  - The best architectures, requirements, and designs emerge from self-organizing teams.
  - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Principles

- From http://agilemanifesto.org/principles.html:
  - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - Business people and developers must work together daily throughout the project.
  - Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - Working software is the primary measure of progress.
  - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - Continuous attention to technical excellence and good design enhances agility.
  - ***Simplicity--the art of maximizing the amount of work not done--is essential.***
  - The best architectures, requirements, and designs emerge from self-organizing teams.
  - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Principles

- From http://agilemanifesto.org/principles.html:
  - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - Business people and developers must work together daily throughout the project.
  - Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - Working software is the primary measure of progress.
  - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - Continuous attention to technical excellence and good design enhances agility.
  - Simplicity--the art of maximizing the amount of work not done--is essential.
  - ***The best architectures, requirements, and designs emerge from self-organizing teams.***
  - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Principles

- From http://agilemanifesto.org/principles.html:
  - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - Business people and developers must work together daily throughout the project.
  - Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - Working software is the primary measure of progress.
  - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - Continuous attention to technical excellence and good design enhances agility.
  - Simplicity--the art of maximizing the amount of work not done--is essential.
  - The best architectures, requirements, and designs emerge from self-organizing teams.
  - ***At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.***

Planning vs. “The Plan” ...or... Agile vs. Waterfall

- “Plans are of little importance, but planning is essential” – Winston Churchill
- “Plans are nothing; planning is everything” – Dwight D. Eisenhower
- “A good plan, violently executed now, is better than a perfect plan next week” – General George S. Patton
- (...and my personal favourite...)
- “No battle plan survives contact with the enemy” – Field Marshal Helmuth von Moltke the Elder

Why did we choose Agile?

- Respond to change during the project
  - Lots can happen in a project which lasts more than a few weeks or months
- Better prediction of quality and content
  - Based on ***actual progress***, rather than “the perfect plan”
- The apparent "control" in the Waterfall model (sizings known, Gantt charts, etc.) is actually a mirage
- Engineers are motivated by challenging problems, and crave prompt and frequent feedback
  - Iterations provide these opportunities
- Better teamwork / team dynamics
- Agile projects deliver “small and often”
  - Management and customers see immediate results & better defined “progress”
  - Engineers get "quick wins" (and credit at annual appraisal time, especially if the project schedule spans the year boundary!)

Comparison of Project Characteristics

Characteristic	Agile	Waterfall	IBM Java 7
Team size	<10	50 → 100's	100's
Location	Single	Usually multiple	>10 locations across 4 continents
Duration	4 weeks →	1 – 3 years	Years
Planning focus	Short-term (iteration plans)	Up-front commitment	Medium-term planning: project split into 6-week milestones, with change control
Project sequencing	Based around time-boxed iterations (usually 1-4 weeks)	DCUT → FVT → SVT → FRT	3 x 2 week iterations with continuous regression testing, followed by QA period (bug fixing only) and Final Regression Test
Team communication style	Face-to-face	Usually remote, using email, phone, etc.	Also adopt collaboration tools: • Lotus Notes team rooms (shared workspaces) • Rational Team Concert (work item tracking) • Lotus Sametime (instant messaging)
Customer feedback	At the end of each iteration	At the end of the project	• Multiple beta program deliveries • Bi-weekly deliveries to IBM product teams
Testing	Test-driven development	Unit Test, FVT, SVT	• Early testing of new function • Continuous automated regression testing

Comparison of Project Characteristics

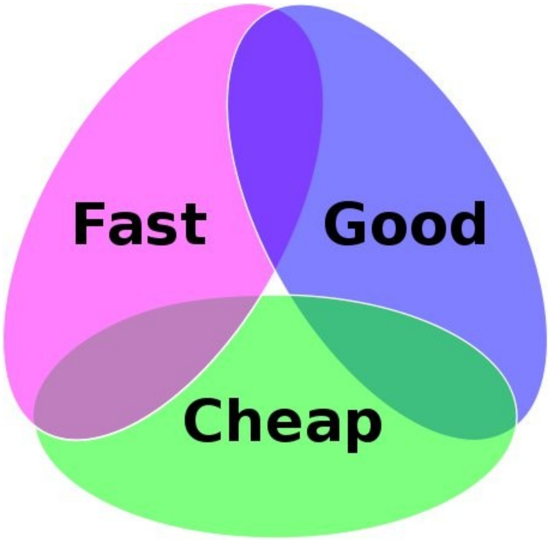
Characteristic	Agile	Waterfall	IBM Java 7
Team size	<10	50 → 100's	100's
Location	Single	Usually multiple	>10 locations across 4 continents
Duration	4 weeks →	1 – 3 years	Years
Planning focus	Short-term (iteration plans)	Up-front commitment	Medium-term planning: project split into 6-week milestones, with change control
Project sequencing	Based around time-boxed iterations (usually 1-4 weeks)	DCUT → FVT → SVT → FRT	3 x 2 week iterations with continuous regression testing, followed by QA period (bug fixing only) and Final Regression Test
Team communication style	Face-to-face	Usually remote, using email, phone, etc.	Also adopt collaboration tools: • Lotus Notes team rooms (shared workspaces) • Rational Team Concert (work item tracking) • Lotus Sametime (instant messaging)
Customer feedback	At the end of each iteration	At the end of the project	• Multiple beta program deliveries • Bi-weekly deliveries to IBM product teams
Testing	Test-driven development	Unit Test, FVT, SVT	• Early testing of new function • Continuous automated regression testing

Comparison of Project Characteristics

Characteristic	**Agile**	**Waterfall**	**IBM Java 7**
Team size	<10	50 → 100's	100's
Location	Single	Usually multiple	>10 locations across 4 continents
Duration	4 weeks →	1 – 3 years	Years
Planning focus	Short-term (iteration plans)	Up-front commitment	Medium-term planning: project split into 6-week milestones, with change control
Project sequencing	Based around time-boxed iterations (usually 1-4 weeks)	DCUT → FVT → SVT → FRT	3 x 2 week iterations with continuous regression testing, followed by QA period (bug fixing only) and Final Regression Test
Team communication style	Face-to-face	Usually remote, using email, phone, etc.	Also adopt collaboration tools: • Lotus Notes team rooms (shared workspaces) • Rational Team Concert (work item tracking) • Lotus Sametime (instant messaging)
Customer feedback	At the end of each iteration	At the end of the project	• Multiple beta program deliveries • Bi-weekly deliveries to IBM product teams
Testing	Test-driven development	Unit Test, FVT, SVT	• Early testing of new function • Continuous automated regression testing

Managing the “Triple Constraint” using Agile

- Tips:
  - Closely manage quality (pick useful metrics) – you will regret it if you don't
  - Test early – test often – fix the bugs – repeat
  - Emphasize to the business leaders the need to maintain ***flexibility of content*** if the date is immovable (as it usually is)
  - The Agile model allows you to ***trade-off content***, as required, to meet the schedule and quality commitments



How we did it...

- Selling Agile to the business
- Building a distributed project team
- Managing communications in a distributed team
- Managing quality
- Infrastructure

“Selling” Agile to your Business

- Stress “frequent planning” over “The Plan”
- Show regular progress
  - Preferable to “wait until the end”
- Be able to demonstrate “ready to ship”
  - Every iteration/milestone
- Invite regular feedback
- ***Welcome*** change – demonstrate flexibility
- If possible, bring on board key influencers – track record of successful Agile delivery
  - My rates are reasonable :-)

Building a distributed project team

- Disadvantages:
  - communication lags
  - “pass the parcel (bug)”
  - language barriers
  - "silo" mentality
- Advantage:
  - timezone coverage; "someone" is always awake to handle critical situations, or continue investigation when others are sleeping
- Consider the product structure and interfaces between components
  - Seek to build independent, accountable teams
  - Structure the product accordingly (Component #1 in Canada, Component #2 in India, ...)
- Focus on skills; look for groups of like-minded people
  - e.g. static compiler knowledge in IBM Toronto; we develop our JIT compiler there
- Still useful to have key general-purpose bug fixers in each location

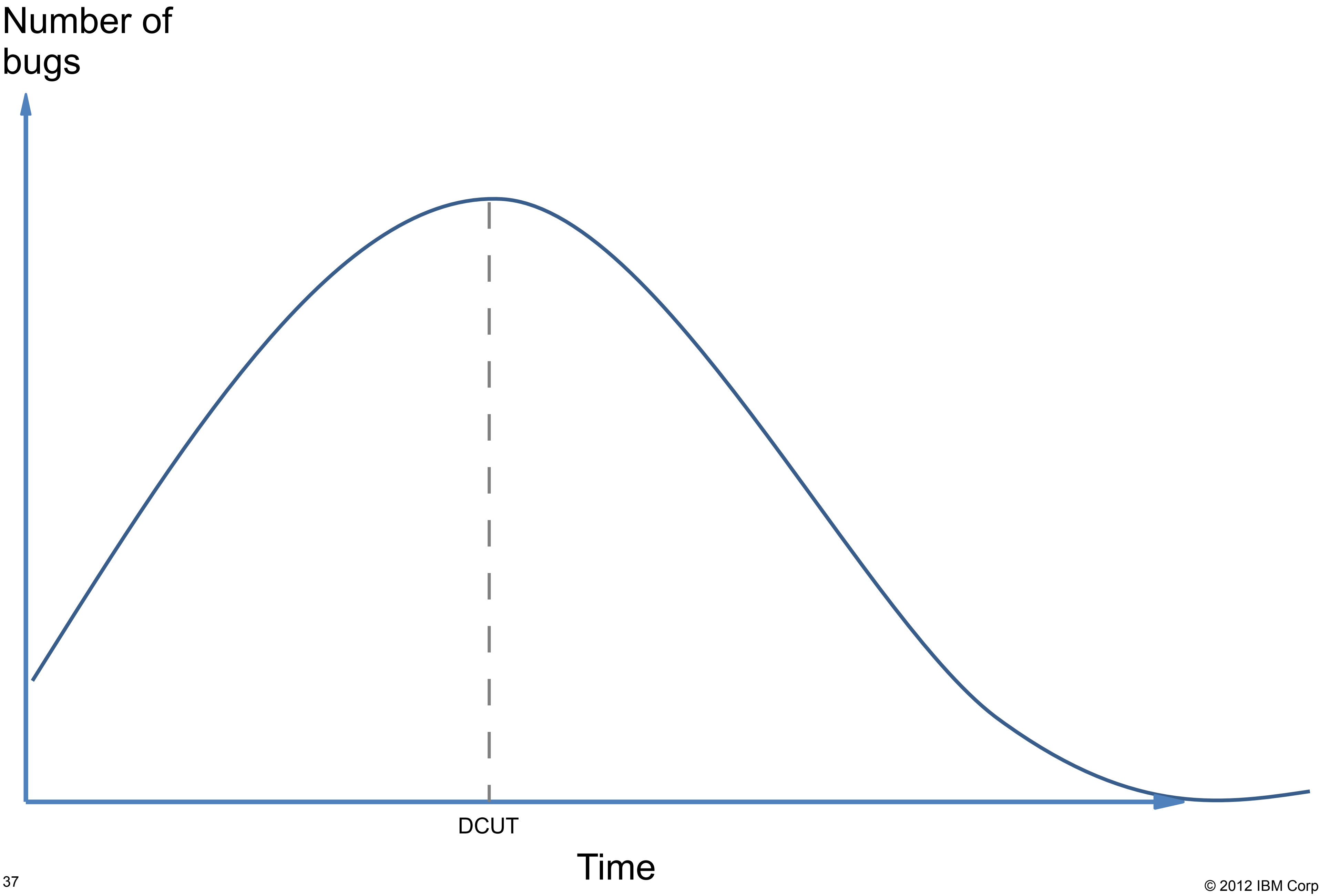
Managing communications in a distributed project team

- n * (n – 1) / 2
- Develop mechanisms for more efficient communication between geographically dispersed teams:
  - "Scrum of Scrums"
  - Collaboration tooling, e.g. Rational Team Concert, Lotus Notes workspaces, Lotus Sameime, webcams
- Keep stakeholders informed
  - Internal stakeholders; focus on their stakeholder key concerns and near-term commitments
  - Customers; demonstrate working software regularly (every iteration, or milestone in our case)
- Focus business leaders on high-level goals and achievements, not the low-level detail or “The Plan”

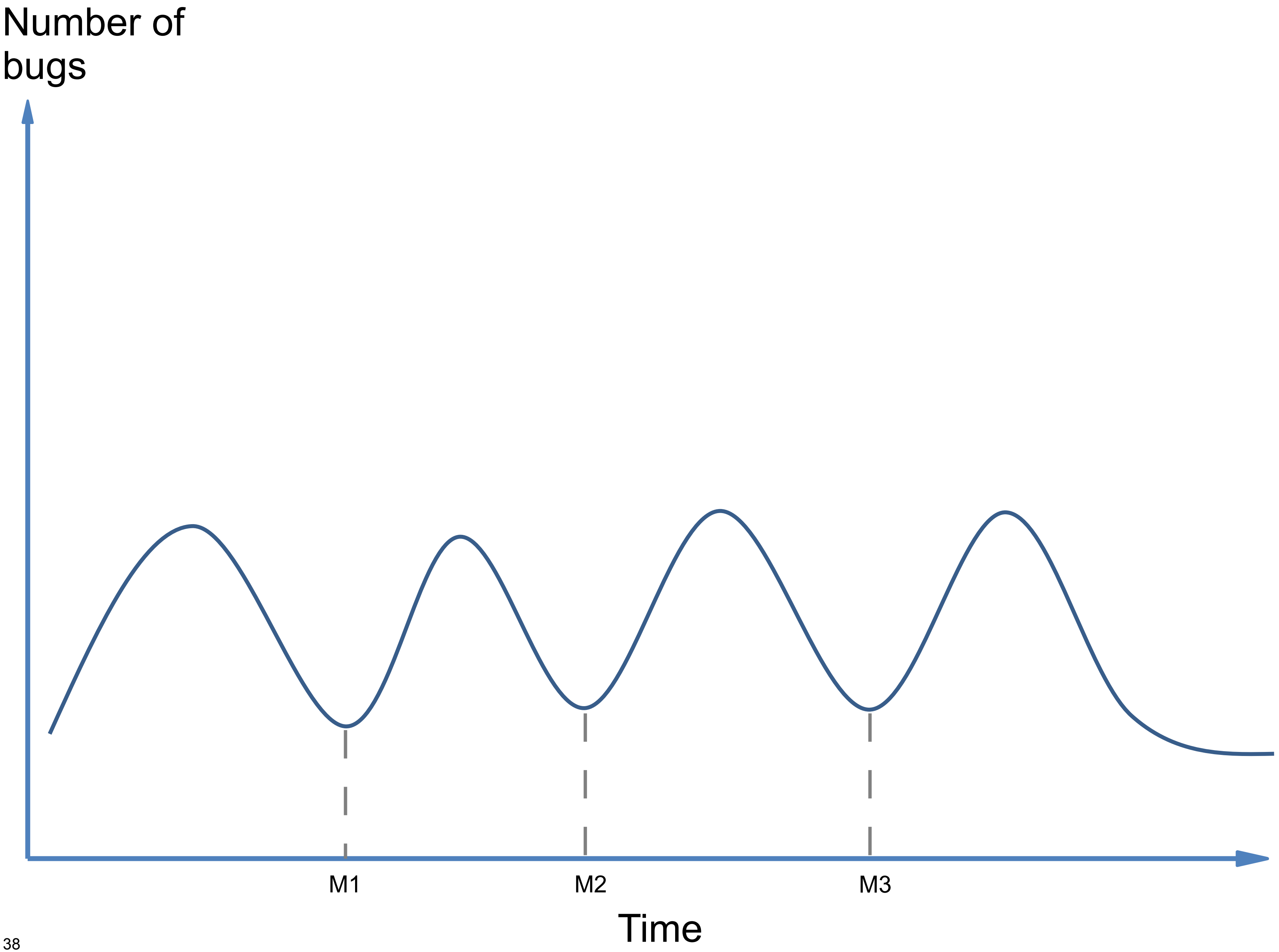
Managing quality

- Maintaining quality throughout the project:
  - Maintain a view of “technical debt” (stuff which didn't complete)
  - Don't let bug backlog get out of control; focus on key blocking bugs, try to remove roadblocks
- Meaningful quality metrics for individual teams, e.g.
  - code coverage
  - bug backlog
  - % of fixes with testcases committed
- End-of-iteration (or end-of-milestone) demonstrations and beta program focuses teams on near-term commitments and the importance of keeping quality high

Classic Waterfall project bug backlog profile

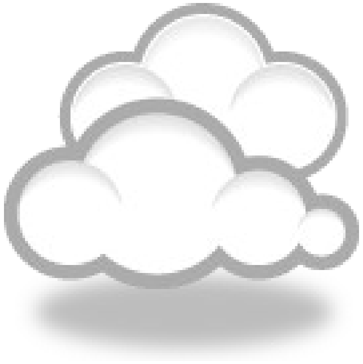


Agile project bug backlog profile

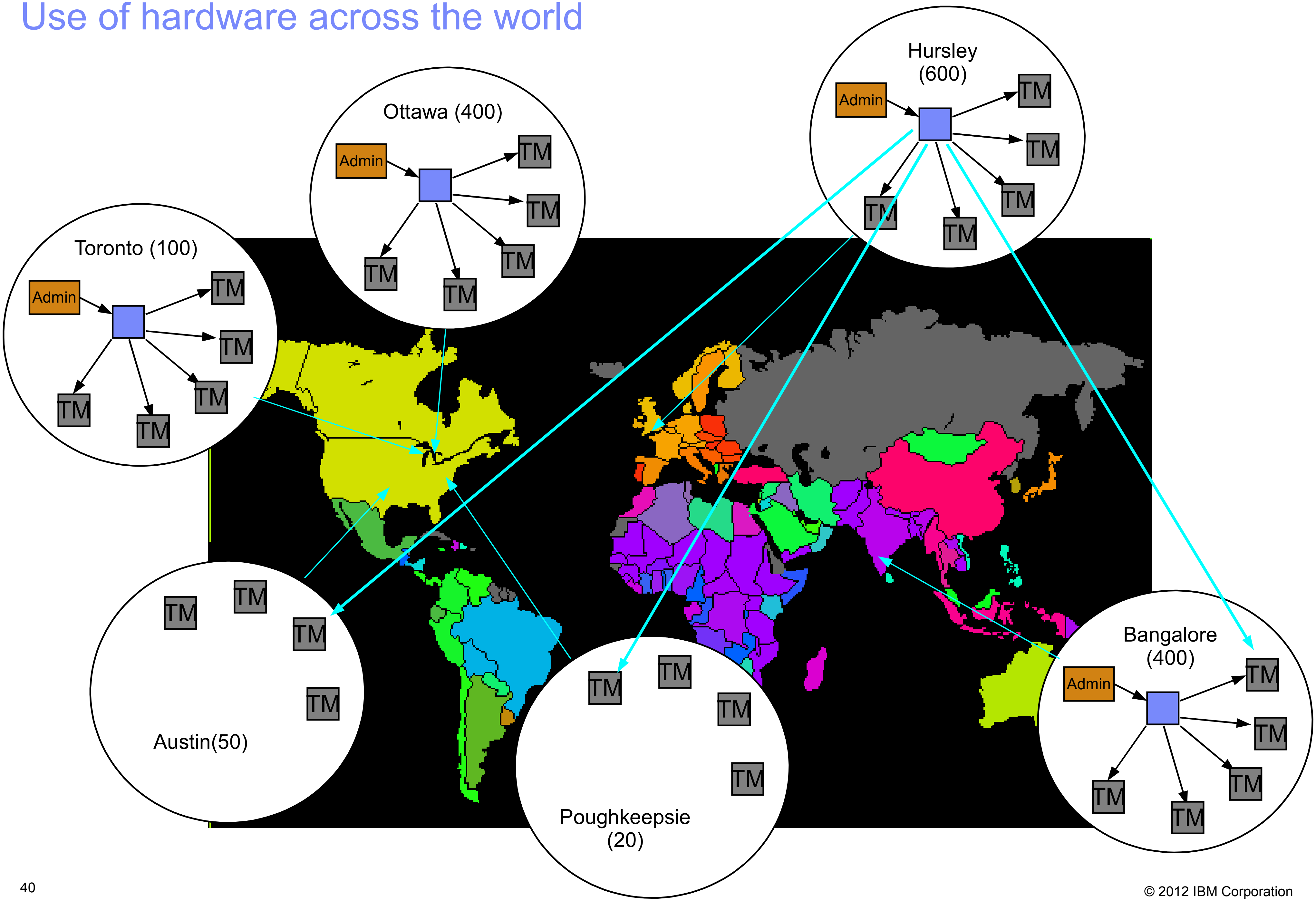


Infrastructure

- Common bug reporting & work item tracking system
  - Central view of quality, progress and work remaining
- Make best use of machines across the world
  - Take advantage of existing hardware to reduce costs
- We employ a cloud-like build & test environment
  - Local expertise & resources
  - Focus on delivery of components to central integration service
  - Without this, we would need people to FTP testcases, have people run tests manually and send back results, etc.
  - Use of cloud to standardize approach to testing
  - See the next presentation!



Use of hardware across the world



The Bottom Line

- Schedule
  - IBM Java 7 – delivered within 1 quarter of Oracle Java 7
  - IBM Java 6 – delivered within 1 year of Sun Java 6
- Quality
  - Highest quality of any IBM Java release to date
- Stakeholders
  - Good feedback from beta customers
  - Early adoption by IBM product groups

Summary

- Applying the Agile principles to large-scale software engineering projects is possible, but non-trivial!
- If possible, split up the product into pieces which can each be developed in an Agile fashion by smaller, co-located teams
- Gain trust from your business leaders by delivering small and often
- Welcome feedback from the business leaders – show them how flexible Agile projects can be
- Agile developers are motivated developers – because they can show progress more easily and quickly than in Waterfall projects
- Quote IBM Java 7 as an example
  - Hundreds of project members across the world
  - 750,000 person-hours effort
  - More timely delivery than previous release

References

- **Get Products and Technologies:**
  - IBM Java Runtimes and SDKs:
    - https://www.ibm.com/developerworks/java/jdk/
  - IBM Monitoring and Diagnostic Tools for Java:
    - https://www.ibm.com/developerworks/java/jdk/tools/

- **Learn:**
  - IBM Java InfoCenter:
    - http://publib.boulder.ibm.com/infocenter/javasdk/v6r0/index.jsp
    - http://publib.boulder.ibm.com/infocenter/java7sdk/v7r0/index.jsp

- **Discuss:**
  - IBM Java Runtimes and SDKs Forum:
    - http://www.ibm.com/developerworks/forums/forum.jspa?forumID=367&start=0

Copyright and Trademarks