

SCM in a Large-Scale Agile Development Project



James Spalding & Paul Dyson
e2x Limited

21st June 2005

Agenda

- Introductions:
 - James Spalding
 - Paul Dyson
- Project overview
- SCM headaches
- Conclusions
- Questions & Answers

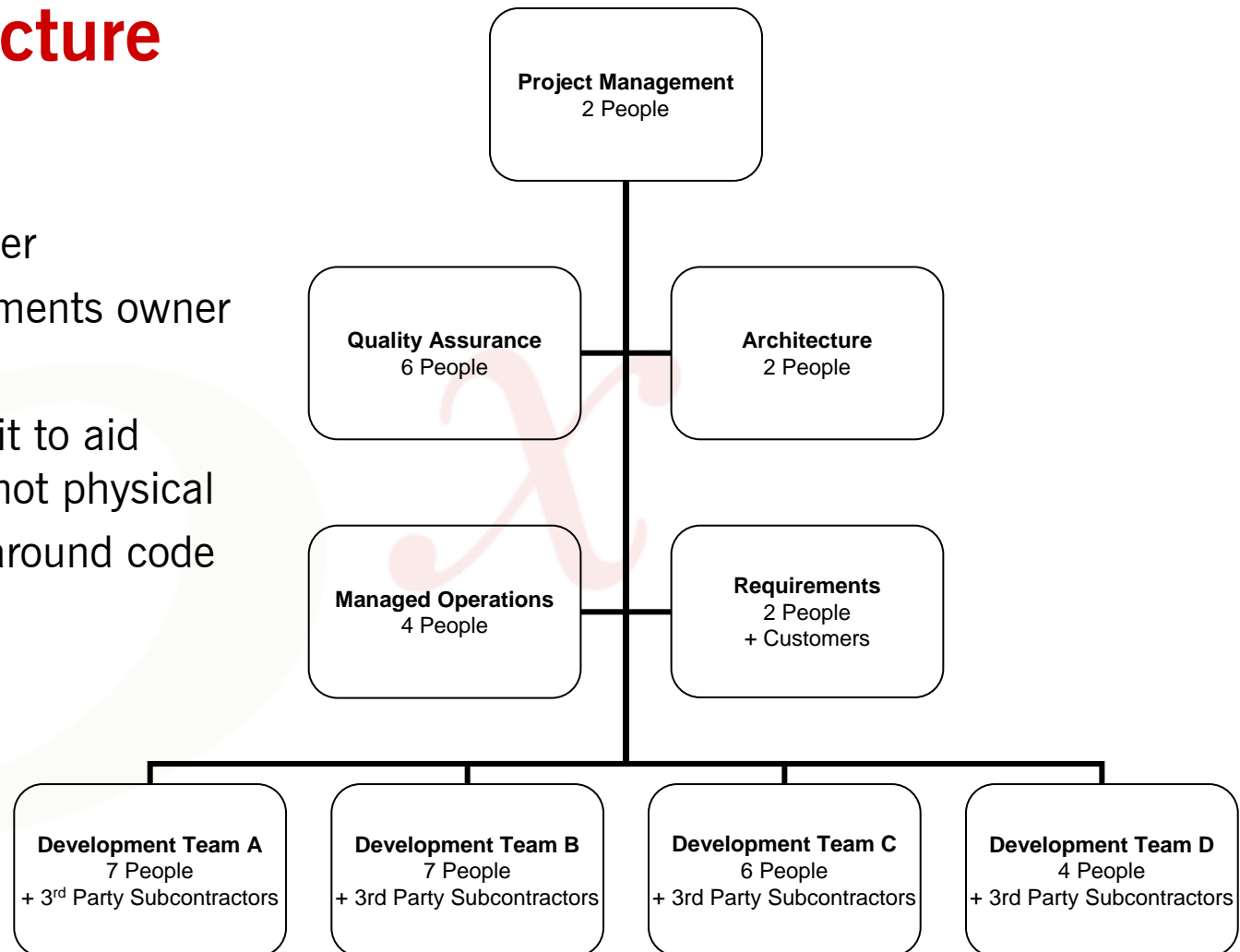


Project Background

- Global e-commerce platform for a multi-national company
 - 80+ Countries and 30+ Languages
 - Bespoke Catalogue per country
 - Requirements incomplete and changing
 - Functionality included B2C, B2B and Web services
 - Core, leading edge, technology previously selected
 - Application server
 - Content management
 - Search Engine
 - Time to market business pressure
 - Agile development process adopted
 - 50+ person team

Team Structure

- Co-Located
- Onsite customer
- Single requirements owner
- Integrated QA
- Functional split to aid management not physical
- Teams move around code base



Development Process

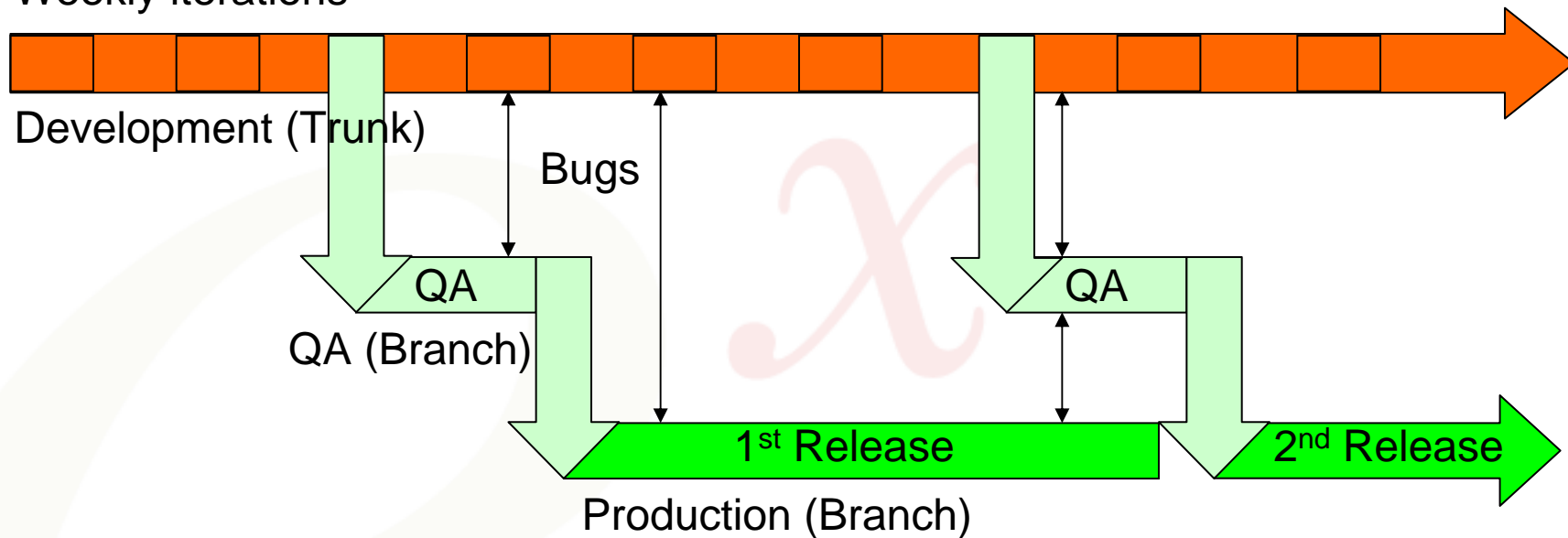
- Light weight communications
 - Stand up meetings
 - Daily customer reviews
 - 'Fit for purpose' documentation
- Heavy weight discipline
 - Single development environment
 - Integrated Quality Assurance
 - Continuous integration

Build & Release

- 5 Builds a day
- 'Tagged' release at the end of every day
 - Automatic QA builds
- Weekly iterations
- Release every 6 weeks
 - Business driven
 - Not all releases candidates went live!
 - Rework

Release Plan

Weekly iterations



Note: Ignoring the complexity of content development!

SCM Headache #1: Tool Selection

- Primary requirements:
 - Speed
 - Scalability
 - Simple administration and interaction
 - Integration with IDE
- Tool selected: CVS
 - Is this a joke?

SCM Headache #2: Maintenance

- At most, three branches:
 - Production
 - QA (release candidate)
 - Development
- Production fixes must be applied to all three branches
- QA fixes must be applied to QA and Development
- Many fixes implemented as a number of tasks carried out by a number of different people
 - How do we identify changes in the face of continuous integration?
 - How do we apply changes quickly and easily to other branches?

SCM Headache #3: 'Selective Assembly'

- Well-executed continuous integration and refactoring leads to a monolithic codebase
 - Not in terms of architecture but in terms of tracing code to requirements
- In a dynamic business environment, long-term requirements change
- An agile development process should be able to support these changes
- How do we ensure that only the required functionality is delivered in a release?

SCM Headache #4: Artefact Dependencies

- ‘Systems’ don’t consist of just code
 - Bespoke code in a variety of languages
 - 3rd-part libraries (application servers)
 - Configuration for various aspects of the bespoke and third-party code
 - Object/relational mappings
 - Schema definitions
 - Binaries
- These artefacts are managed using different processes and maintained by people with different skill sets
- In an agile development process, we want everyone to use the same tools
 - Or for the tools to seamlessly integrate

The SCM Migraine

- Given our choice of tool, all the other problems could only be resolved with lots of careful thought and very hard work
 - Supplemented with bespoke scripts developed over time
- Could another choice of tool have made this all much easier?
 - We don't think so

Conclusion

- SCM and Agile are not different types of project; agile development projects need to do SCM
- The Agile SCM problem is a tools problem
 - If you sacrifice agility, you are no longer agile
 - We need advanced SCM functionality from tools that are fast, scalable, simple to use and administer, and integrate with the tools used by the ‘developers’

Questions & Answers



Or contact us at

- paul.dyson@e2x.co.uk
- james.spalding@e2x.co.uk
- www.e2x.co.uk