

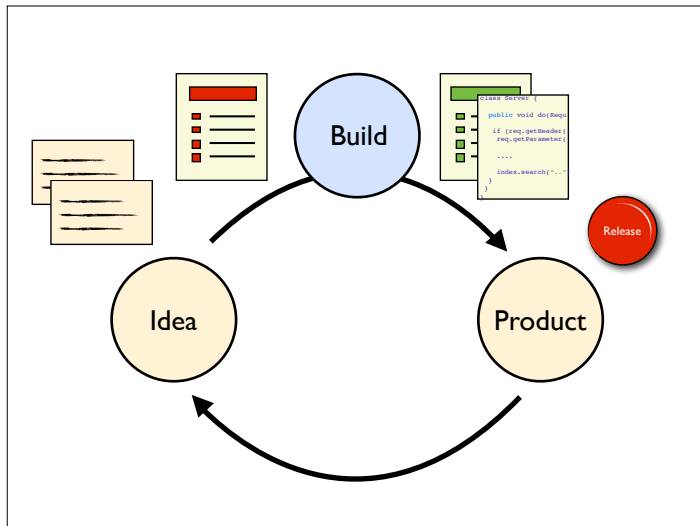
Paths to Production – Pipelines in Practice

Robert Chatley

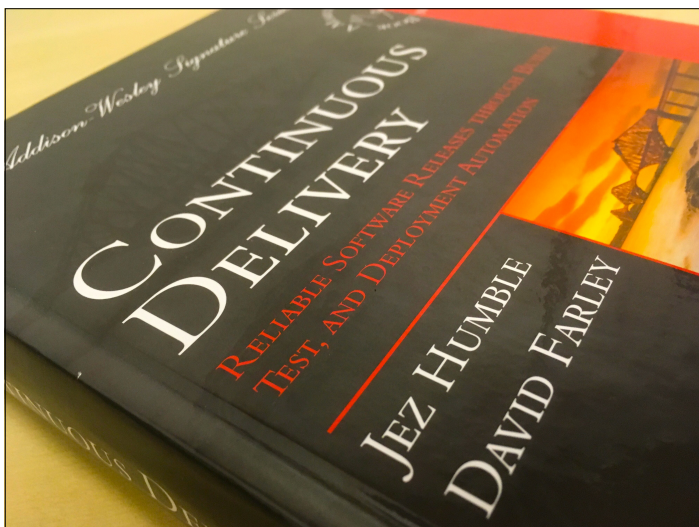
rbc@imperial.ac.uk

@rchatley

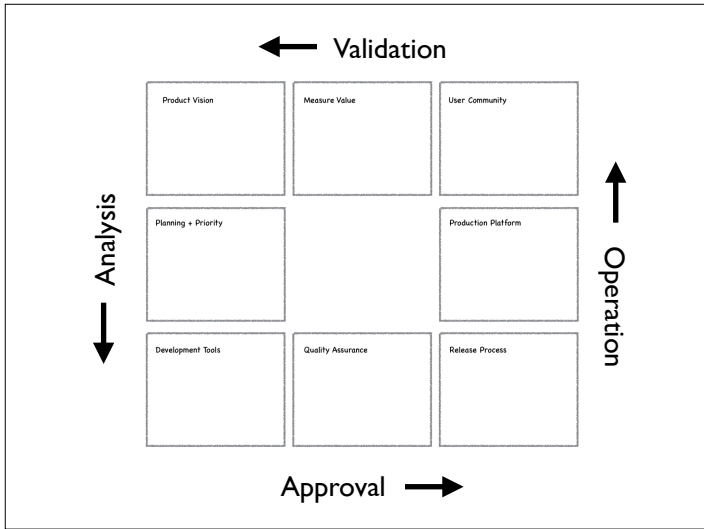
Robert Chatley is a Principal Teaching Fellow in Software Engineering at Imperial College London, but coming from an industrial background. This presentation focusses on observations from various consulting projects with different companies on how continuous delivery is done (or not).



Many development teams have adopted agile methods over the last 10-15 years, and within their teams many are doing well, employing user stories, TDD etc to build product increments. But sometimes releasing these increments to production is hard.



Sometimes the provisioning/infrastructure part causes trouble. The “last mile” is hard. Jez and Dave wrote a lot about this in their great book Continuous Delivery. They have a lot of practices that when joined together, enable us to have a pipeline leading all the way to production in a reliable repeatable way.

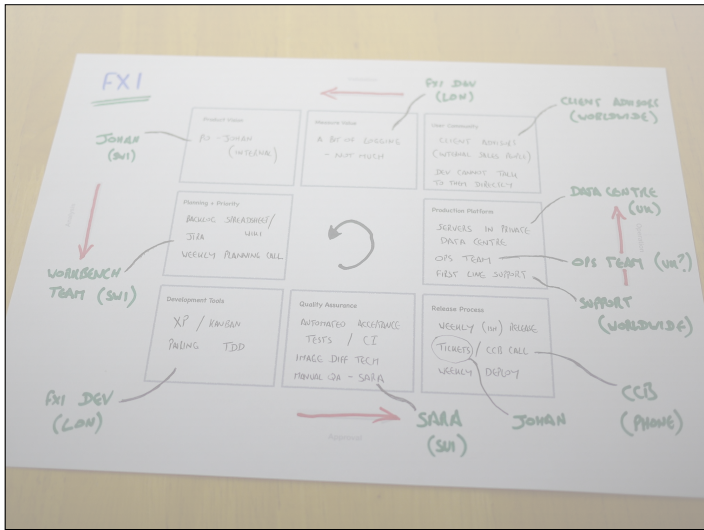


In working with teams I often use a chart that I call a Release Cycle Map to help the team to think about all the activities between someone having an idea for a feature, through that feature being prioritised, developed, tested, released, and ideally measured in terms of value delivered.

The stages are (anticlockwise from top left):
 Product Vision, Planning + Priority, Development Tools, Quality Assurance, Release Process, Production Platform, User Community and Measure Value.



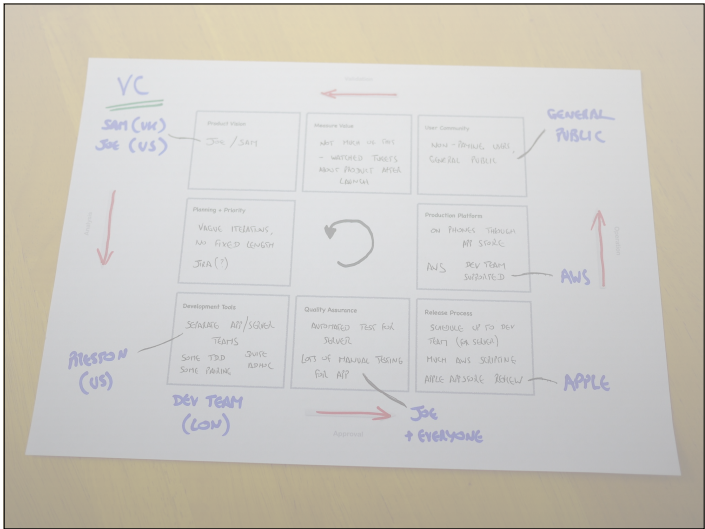
The next slide shows an example of the Release Cycle Map completed for a team in an investment bank building an internal application with business users.



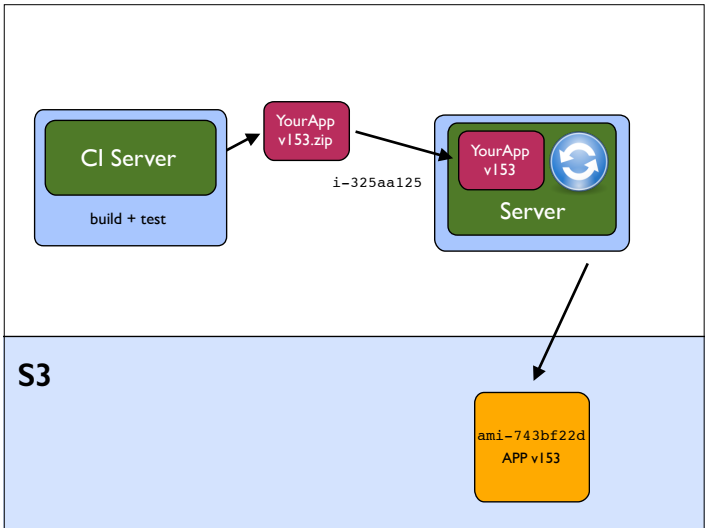
Blurred for anonymity. This team's release process was controlled by a separate ops team, who required a conversation on a weekly conference call in order to approve releases. Releasing more than once per week was practically impossible.



The next slide shows an example of the Release Cycle Map completed for a team in a startup building an iPhone application with a node.js backend deployed to AWS.



Blurred for anonymity. The small organisation had little bureaucracy, the backend team could release to AWS pretty much whenever they liked, but releasing a new version of the iPhone app required approval by Apple, and release through the app store, which could easily take a couple of weeks.

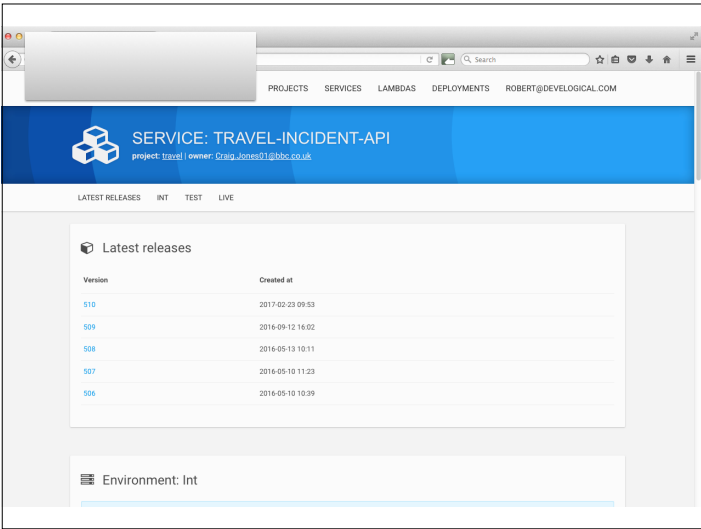


To deploy in AWS with immutable infrastructure, the team set up their CI server to “bake” AMIs as the output artefact by taking a snapshot of a running instance with the new software installed, store the AMIs in S3, and then deploy these into an auto-scaling group.

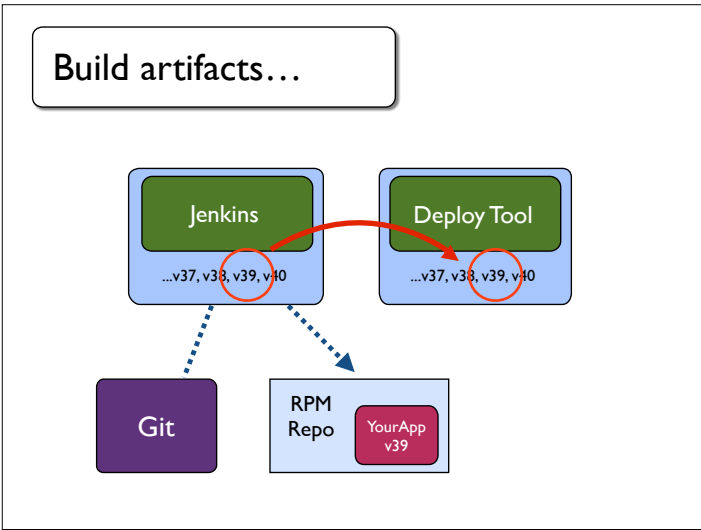
The next example is from a large media organisation.

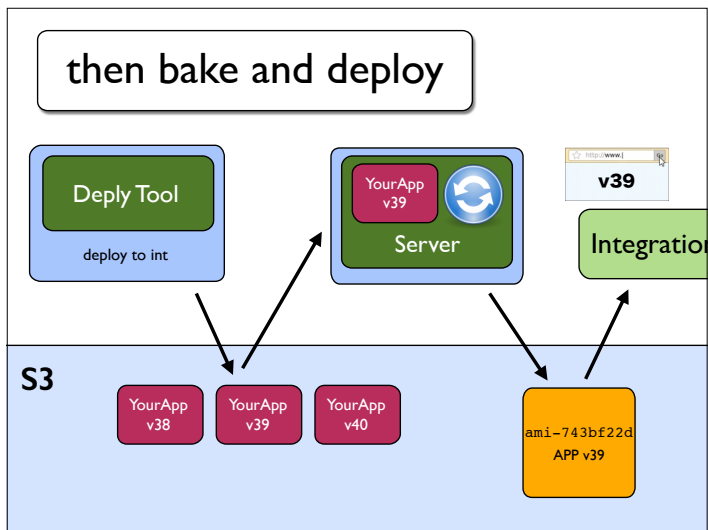


This organisation is moving from deployment through a central ops team, to each team deploying its own software to AWS, but through a tool provided by a central platform team. This gives a consistent deployment process across teams.



They set their CI builds to produce RPMS, and on a successful build of new RPMS, inform the Deploy Tool of the new version being available, ready for deployment.



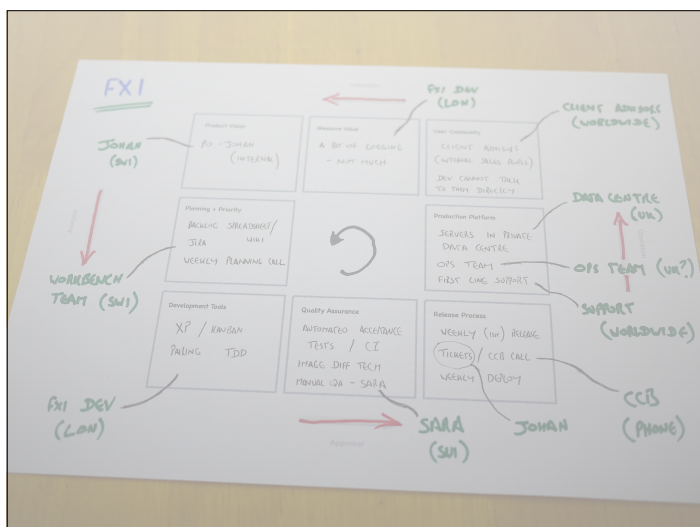


Once the RPMs are available, and a deploy is triggered, an AMI is baked and released into an autoscaling group, as with the startup example before.



These techniques all allow us to get product increments into production more smoothly, but in his book *The Lean Startup*, Eric Ries stresses the importance of producing “validated learning” from each iteration that you perform, not just releasing more code. What if you expend a lot of energy (and cash) to deliver a function that no-one cares about (or worse, hates) - but you have no way of measuring the perceived value.

We encourage thinking about the success criteria during the definition of each user story, and making the collection of appropriate data part of the delivery of each story.



In the banking environment, one feature was suggested by the product owner to change a free text input into a drop down box. But which values to put in the drop down? An experiment was run by adding one line of code to log each value typed into the box by users, so that these could be ranked to make a data-driven decision. But releasing that one line took a whole week. A faster (without a separate ops organisation) release process would have allowed for more continuous experimentation.



Any Questions?

Robert Chatley

rbc@imperial.ac.uk

 [@rchatley](https://twitter.com/rchatley)

